SESSION

NOVEL IMAGE PROCESSING ALGORITHMS

Chair(s)

TBA

Hybrid Implementation of Image Stitching on Computers with GPUs

Chenggang Lai and Miaoqing Huang
Department of Computer Science and Computer Engineering
University of Arkansas
{cl004,mqhuang}@uark.edu

Abstract— *Graphics processing units (GPUs) are capable of achieving remarkable performance improvements for a broad range of applications. However, they have not been widely adopted in embedded systems and mobile devices as accelerators mainly due to their relatively higher power consumption compared with embedded microprocessors. In this work, we conduct a comprehensive analysis regarding the feasibility and potential of accelerating image stitching application using GPUs in embedded systems in addition to desktop systems. High-resolution panoramas can be generated by stitching multiple images together. The image stitching process consists of four steps, i.e., feature extraction using Speeded Up Robust Feature (SURF) algorithm, image matching using Random Sample Consensus (RANSAC) algorithm, bundle adjustment, and image blending. An additional step can be taken to crop the dark surrounding areas in the stitched image. We carried out our experiments on both the Nvidia Jetson TK1 kit and a desktop computer. The Jetson TK1 ket consists of one Tegra K1 SOC, which features one quad-core ARM Cortex-A15 CPU and 192 Kepler CUDA cores. On both platforms, it is found that the pure GPU implementation can outperform the corresponding CPU by 2 times. Further we propose a hybrid approach that optimally distributes workload between the parallel GPU and the sequential CPU to achieve the best performance.*

Keywords: Panorama; SURF; RANSAC; image stitching; GPU.

1. Introduction

High-resolution panoramas can be made by stitching multiple images together. The method can improve the view field of a camera by combining several views of a scene into a single view. Image stitching algorithms are most widely used in computer vision. They create the high-resolution photo-mosaics, which are used to produce satellite photos and digital maps. In the real world, they can be used to create beautiful ultra wide-angle panoramas.

In general, image stitching consists of several steps including image matching and blending. For the image matching step, we first extract distinctive features from each image, and then match these features to establish a global correspondence. The last step is to estimate the geometric transformation between two images. There are some popular algorithms for extracting features, such as scale-invariant feature transform (SIFT) and SURF. SIFT, presented by Lowe in 2004 [1], is invariant to scale changes and rotation. It is used in many important applications, e.g., autonomous vehicle and computer-human interaction. However, it is typically computation intensive and requires a long processing time on traditional single-core processors. In this paper, we choose SURF algorithm to extract the features because SURF is a high speed algorithm with high quality [2]. And we use RANSAC algorithm [3] to identify the correspondences among sets of matches. For the blending process, first we adjust the images into the same coordinate by using bundle adjustment [4], and use multi-bending method for image blending because it has a good performance.

Hardware, like FPGA, GPU and Intel MIC (many-integrated-core) coprocessors, can be used as accelerators in image processing. GPU devices are wildly used in personal computers for image processing and have an ability to provide massively parallel computing capability. As the release of NVIDIA's Compute Unified Device Architecture (CUDA), researchers can use C-like programming language to design programs for both CPU and GPU without knowing fundamental knowledge on computer graphics. Nowdays, Nvidia's GPUs have powered many TOP500 supercomputers in the world [5]–[7]. The massive parallelism of GPU can be used to accelerate a broad range of applications.

In this work, we investigate the potential use of GPUs in desktop computers and embedded systems for image stitching. For embedded systems, we implement the image stitching application on the Nvidia Jetson TK1 kit, which contains both multicore CPU and manycore GPU. The results of this paper demonstrate that GPU implementations of image stitching are capable of achieving more than two times speedup than the CPU implementations. Based on the results in this work, we expect that future embedded systems and mobile devices will become hybrid systems in which the CPUs deal with most sequential tasks while GPUs are used to handle most parallel tasks.

The remainder of the paper is organized as follows. The related work is briefly discussed in Section 2. In Section 3 we introduce several main algorithms applied on image stitching application. In Section 4 we discuss the experiments on two

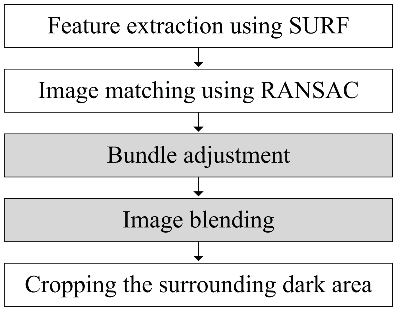

Feature extraction using SURF
Image matching using RANSAC
Bundle adjustment
Image blending
Cropping the surrounding dark area

Fig. 1: Hybrid implementation of image stitching algorithm (the steps with gray background are implemented on GPU).

different platforms, i.e., one workstation and one embedded system. We also compare the results between the CPU implementations and the GPU implementations. We provide the conclusions in Section 5.

2. Related Work

Hardware accelerators have been widely used in embedded system to achieve high performance. A large number of work, such as [8], [9], have been proposed to optimize code for a high performance. The work in [10] introduced a co-design method for embedded system with hardware acceleration. Some computer vision algorithms, such as SIFT and SURF, have been accelerated on many technologies. The work in [11] showed us several hardware architectures for improving performance of SIFT and SURF on ASIC or FPGA devices. But they only dealt with some small images. Given the mobile GPUs, such as Nvidia Tegra [12], PowerVR SGX [13] and Adreo [14], there were several work to implement image processing algorithms [15]–[17]. Compared with these GPUs integrated on the same system-on-chip with the CPUs, the standalone GPUs present a stronger computation ability and memory management ability.

Methods based on feature matching and direct estimation can be implemented for image stitching. The algorithms based on features [18], [19] establish correspondences between points or other geometrical entities. Direct estimations [20], [21] try to minimize the error based on a function by iteratively estimating camera parameters. Direct estimation is more accurate because it uses all of the available data, but it depends on a fragile brightness constancy assumption. Some work based on invariant features [22]–[24] have achieved a huge progress. Compared with the traditional work, these work provide more reliable results of detecting features. In this work, we use SURF algorithm to extract the features because SURF is a high speed algorithm with high quality. We use standalone GPUs to accelerator the whole process of image stitching and achieve a good performance. Further we distribute the workload between the CPU and the GPU for taking advantages of both types of processors, as shown in Figure 1.

Algorithm 1 RANSAC Algorithem.

```
while (iteration $\leqslant K$) do
    Randomly select a minimum set (say, $S1$) of $n$ samples to inliers;
    A model is fitted to these inliers;
    for (features in the data but not in inliers) do
        if (feature is fitted to model)
            Add it to inliers;
    end for
    if (number of inliers > $N_{max}$)
        find a better model;
        $N_{max}$ = number of inliers;
    Add one to iteration;
end while.
```

3. Image stiching mechanism

3.1 Feature extraction

The first step in the image stitching process is to extract SURF features from all images. There are some difference between SIFT and SURF in the process of detecting features. SIFT uses cascaded filters to detect scale-invariant characteristic points, filtering each layer based on Gaussians of increasing sigma values and taking the difference. For the SURF algorithm, it uses a square-shaped filter as an approximation of Gaussian smoothing. The square-shaped filter is chosen because it can be evaluated extremely efficiently using the so-called integral image, M, as shown in Equation (1), in which I is the input image. Due to the use of integral images, the process with a square filter is much faster than SIFT. SURF detects features by selecting an approximation to determine the Hessian for stability, repeatability and speed. An ideal filter would construct the Hessian by convolving the second-order derivatives of a Gaussian of a given scale σ with the input image. This is approximated by replacing the second order Gaussian filters with a box filter [25].

$$M(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \tag{1}$$

SURF uses a lot of different scales to evaluate the filters in order to achieve scale invariance. A minimum threshold is used to limit the total number of features. In order to achieve rotation invariance, SURF detects the dominant orientation of the image around each feature using the high-pass coefficients in the directions of x and y. When the scale, orientation and position of an element are determined, a feature descriptor can be produced for matching across images.

3.2 Image matching

Given the features extracted by SURF, RANSAC algorithm selects the matching features between two images or several images. Inliers and outliers are from a given dataset. Inliers represent the features that support our model hypothesis. Outliers represent the features that are false correspondence. First, RANSAC randomly selects a set of samples and the process will be repeated many times. For each process, a model is fitted to the set of samples and other features out of the set are used to test the model. The model with the largest support is used as a resulting model, as show in Algorithm 1. In order to verify the model, a confidence p (usually set to 0.99) gives the probability that the algorithm has a useful result. The maximum number of iterations K for finding a useful model can be computed as shown in Equation (2). n is the minimum number of samples in the set $S1$ to determine a model. ω represent the probability that any selected sample in the set $S1$ is an inlier.

$$K = \frac{\ln(1-p)}{\ln(1-\omega^n)} \tag{2}$$

3.3 Bundle Adjustment

Bundle adjustment can solve the problem for all of the camera parameters jointly. It is an essential step as concatenation of pairwise homographes would cause accumulated errors and disregard multiple constraints between images. Bundle adjustment can refine viewing parameters by minimizing some cost function that quantifies the model fitting error. It can deal with a very wide variety of features and camera types. Because of the accurate statistical error models, the reconstruction result of bundle adjustment has a well-developed quality.

The process of bundle adjustment consists of several steps. (1) We select one image as a reference surface and add each other image into this surface until all images are on the same surface. However, a problem is created in this process. Due to the possibly intrinsic calibration and radial distortion, all images cannot maintain a flat representation. Therefore, there is a choice to use a cylindrical or spherical projection for compositing a huge panorama. (2) For transformation process, we need to compute homography and optimize the matrix of parameters by minimizing the distance between neighbor images. In other words, we need to minimize the error of transformation. The nonlinear minimum square evaluation is used in this study, as shown in Equation (3). The d_e is the Euclidian distance. X_i amd X_i' represent the matching points. And estimated homography H will be adjusted for a better result.

$$d_e = \sum (d_e(X_i, H^{-1}X_i')^2 + d_e(X_i', HX_i)^2) \tag{3}$$

3.4 Image Blending

The goal of image blending is to generate a result image in which there is no obvious transition between original images. There are several popular methods for image blending. Linear method, such as Alpha blending, employs an alpha weighted sum of images in the overlapping result image as shown in Equation (4).

Fig. 2: Jetson TK1 kit.

$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right} \tag{4}$$

The idea of linear blending method is simple and the execution time is short. However, this method results in blurring problem in the high frequency detail. In other words, there is a tradeoff between the execution time and image quality. The linear blending method is still a good choice if you do not need a result with high quality. To generate high-quality result image, we use the multi-band blending algorithm, presented by Burt and Adelson in 1983 [26]. The multi-band blending algorithm can generate better results than the linear method. The idea of multi-band blending is to blend low frequencies over a large spatial range and high frequencies over a short range. It consists of 4 steps: (1) compute Laplacian pyramid; (2) compute Gaussian pyramid on weight image; (3) blend Laplacians using Gaussian blurred weights; (4) reconstruct the final image. The basic idea is to use Laplacians pyramid to decompose images into a collection of N band pass images. In this study, we choose a 2-band pass images. The Equation (5) forms a combined Laplacian pyramid. $I_{left,m}$ and $I_{right,m}$ represent two images that are the m_{th} level of Laplacian pyramid decomposition. G_m represents the m_{th} level of Gaussian pyramid decomposition of the image mask. L_m is the combined result based on the m_{th} Laplacian pyramid decomposition. The same process is applied to all the pixels.

$$L_m(i,j) = G_m(i,j) I_{left,m}(i,j) + (1 - G_m(i,j)) I_{right,m}(i,j) \tag{5}$$

4. Experiments and Results

We carried out the experiments on two platforms, a workstation with a high-end GPU and the Nvidia Jetson TK1 development kit. The workstation contains one Intel Core i7-3820 3.6-GHz CPU and one Nvidia Tesla K20 GPU, which contains 2,496 Kepler CUDA cores running at 706 MHz. The Jetson TK1 kit contains one Tegra K1 SOC as shwon in Figure 2, which features one quad-core ARM Cortex-A15 CPU running up to 2.3 GHz and 192 Kepler CUDA cores running up to 852 MHz.

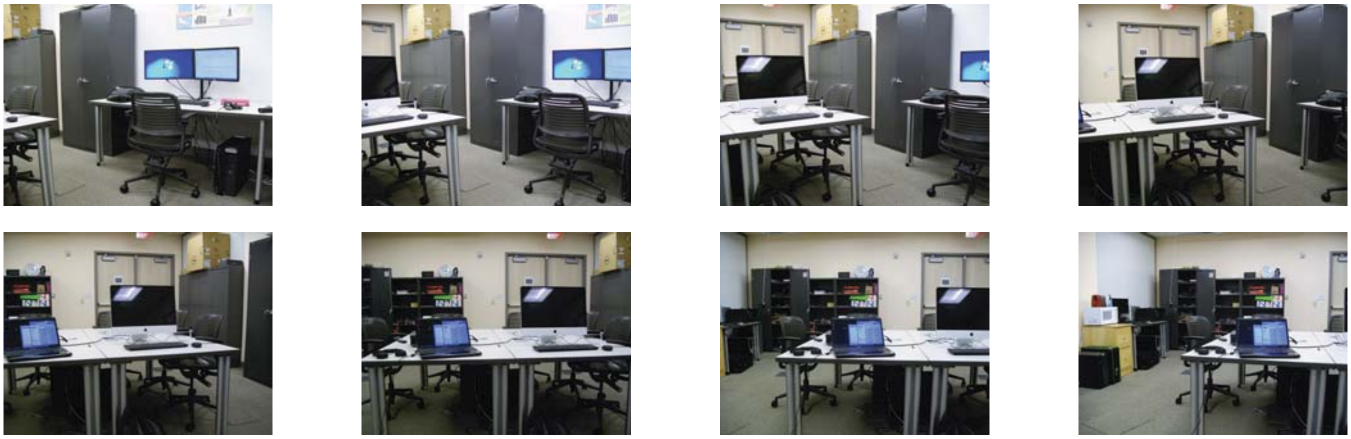
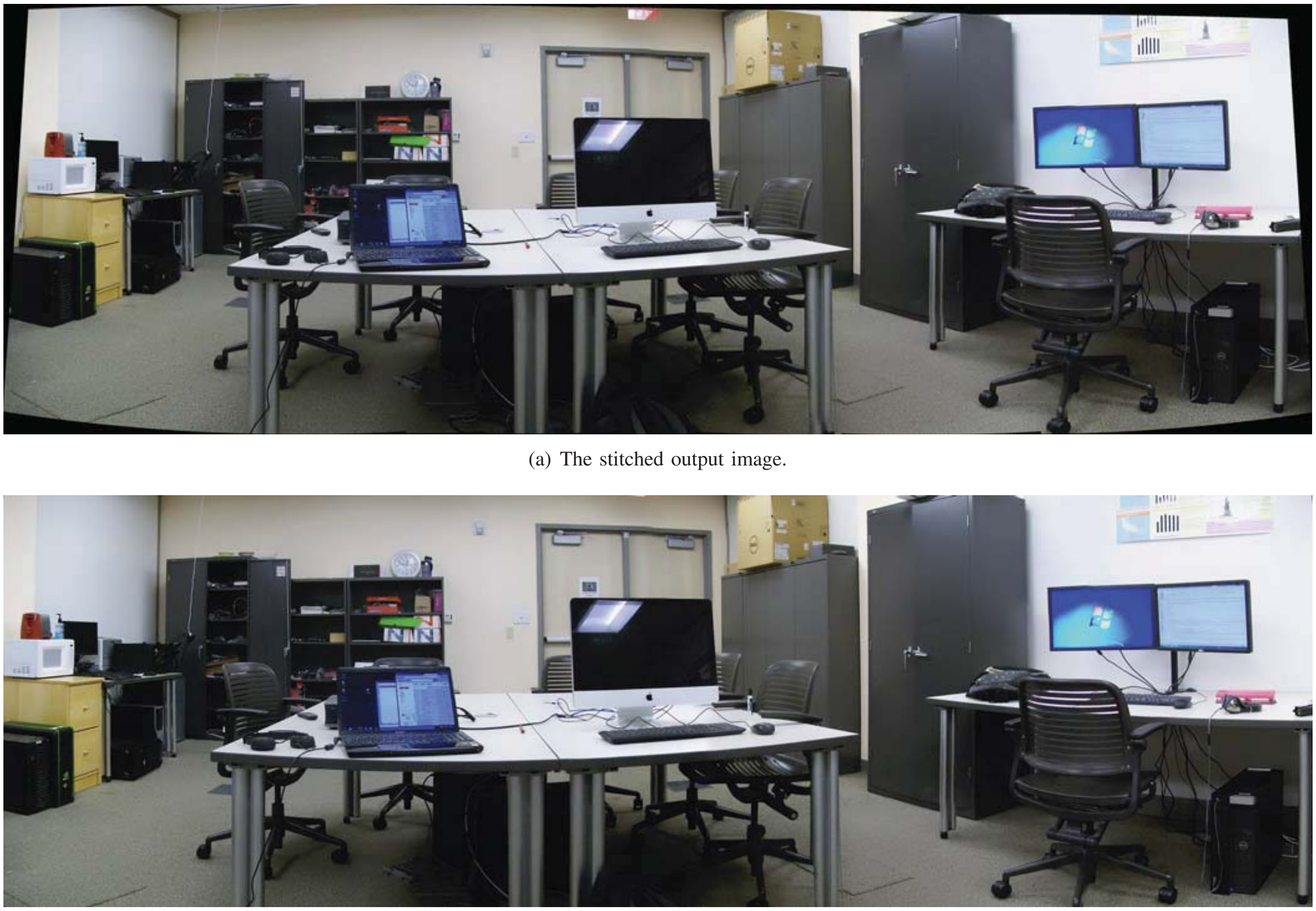


Fig. 3: The input images.



(a) The stitched output image.

(b) The cropped panorama.

Fig. 4: Typical results of image stitching.

In this study, the source data as shown in Figure 3 consists of 8 images. The size of each image is 3,664×2,748. The output of image stitching has a dimension of 8,278×2,721 as shown in Figure 4(a).

The whole image stitching process is divided into 5 stages as shown in Figure 1. For pure CPU model and pure GPU

Table 1: Performance comparison between CPUs and GPUs for image stitching on a single result image (unit: *second*).

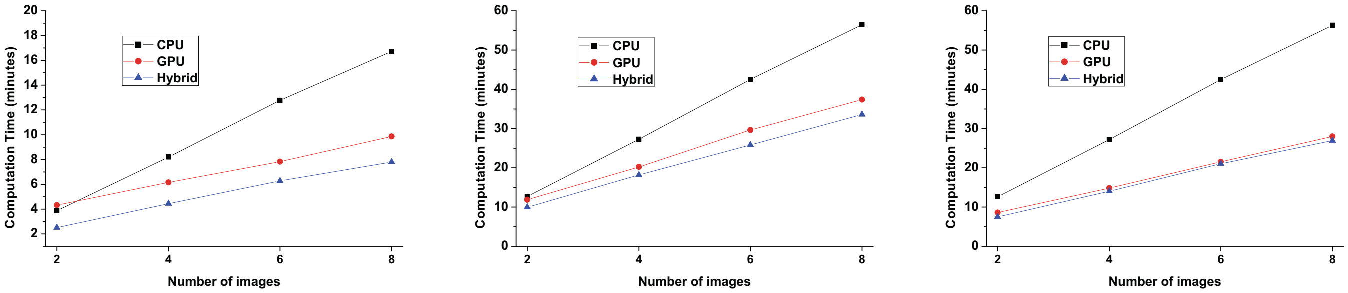
Stage of stitching	Input: 8 images of 3,664×2,748								
	Workstation			Jetson TK1 *			Jetson TK1†		
	CPU	GPU	Hybrid	CPU	GPU	Hybrid	CPU	GPU	Hybrid
Feature extraction	1.064	2.521	1.089	3.111	5.821	3.082	3.086	3.513	3.072
Image matching	0.459	0.458	0.469	1.243	1.663	1.251	1.245	0.999	1.242
Bundle adjustment	1.047	2.376	1.912	3.193	5.73	5.116	3.843	5.507	4.763
Image blending	13.976	4.332	4.324	48.204	23.29	23.468	48.123	17.302	17.192
Total‡	16.72	9.86	7.797	56.451	37.359	33.578	56.298	27.989	26.928

*TK1 GPU uses a default clock frequency of 72 MHz.
†TK1 GPU uses the maximum clock frequency of 852 MHz.
‡The total time includes the final cropping process, which takes about 1 second and runs on the CPU.

Table 2: Performance comparison between CPUs and GPUs for image stitching on multiple result images (unit: *second*).

Number of input images	Input image size: 3,664×2,748								
	Workstation			Jetson TK1*			Jetson TK1†		
	CPU	GPU	Hybrid	CPU	GPU	Hybrid	CPU	GPU	Hybrid
2	3.863	4.318	2.506	12.688	11.849	9.955	12.616	8.593	7.5037
4	8.197	6.156	4.438	27.255	20.235	18.172	27.18	14.834	14.045
6	12.773	7.832	6.276	42.495	29.622	25.825	42.472	21.516	21.011
8	16.72	9.86	7.797	56.451	37.359	33.579	56.298	27.989	26.928

*TK1 GPU uses a default clock frequency of 72 MHz.
†TK1 GPU uses a maximum clock frequency of 852 MHz.



(a) Workstation. (b) Jetson TK1 with default clock frequency. (c) Jetson TK1 with maximum clock frequency.

Fig. 5: Performance of image stitching on three different models.

model, all stages are executed on the CPU or the GPU platforms, respectively. For the hybrid model, the first two stages, feature extraction and image matching, are running on the CPU platform because SURF and RANSAC have a good performance on CPU. These two stages will take more time on GPU platform because of communication between the host and GPU. The next two stages are executed on the GPU platform. When we get the output of image stitching as shown in Figure 4(a), we will cut the dark areas that surround the image by a cropping process as shown in Figure 4(b).

Table 1 illustrates the performance of significant stages of image stitching among three different models. We exclude the time of cropping process due to the fact that execution time of the cropping process is relatively short (about 1 second). From Table 1, it can be found that the Tesla K20 GPU can consistently outperform the Intel Core i7 processor by about 1.5× because the GPU is very efficient at the image blending stage. In the hybrid model, it has a better performance than the pure GPU model because the CPU is faster than the GPU on the feature extraction stage. Therefore, a hybrid design involving both CPU and GPU may be more appropriate than the implementation on a single device. From Table 1, it seems that the CPU spends less time on the bundle adjustment than the GPU. However, in the real implementation, it is found that the two stages, bundle adjustment and image blending, are closely integrated together. If the bundle adjustment is moved to the CPU, it will introduce significant communication overhead between the CPU and the GPU, eventually taking longer time. Therefore, the bundle adjustment stays on the GPU side.

On the Jetson TK1 platform, we implement image stitching on different GPU clock frequencies. In one case the default clock frequency of the GPU is 72 MHz in order to save the power. The GPU frequency may rise due to the intensive computation. In the other case we set the default frequency at 852 MHz so that the GPU always runs at the maximum rate for image processing. From the result, it can be found that the performance with high frequency is better than that with low frequency. Besides, the hybrid model with high GPU clock frequency has a about two times speedup than CPU model. Compared with the Jetson TK1 platform, the workstation has a better performance because the number of CUDA cores drops from 2,496 (on Telsa K20) to 192 (on Tegra K1 SOC). In addition the Tesla K20 has a higher memory bandwidth than the Tegra K1 SOC.

We also try to run image stitching application on different numbers of input images. The result is shown in Table 2 and Figure 5. From the Figure 5(a), GPU model takes more time than CPU model when dealing with 2 input images. The reason is that the communicating overhead between CPU and GPU on the first two stages has a large weight when using a small number of images. As the number of input images grows, the GPU can increasingly outperform the CPU. Overall, the hybird mode can achieve the best performance. On the Jetson TK1 platform, GPU model running at the maximum frequency has a similar performance as the hybrid mode. This is due to the fact that the GPU has similar performance as the CPU when the GPU is running at 852 MHz.

Through the above three models, it has been clearly illustrated that the parallel implementation of image stitching on the hybrid model has an evident advantage. Given the platform of Jetson TK1, the GPU with maximum clock frequency can outperform the CPU by almost 2 times.

5. Conclusions

The GPU has been widely adopted as a hardware accelerator in high-performance computing domain for performance improvement. In this work, we parallelize the image stitching on two platforms, i.e., the workstation with high-end GPUs and embedded systems with energy-efficient GPUs. The experiment results clearly demonstrate the advantages for using GPU accelerators to improve the performance of computer vision algorithms. Among the three models, i.e., the pure CPU implementation, the pure GPU implementation, and the hybrid implementation, the hybrid model working on both the CPU and the GPU can achieve the best performance.

In the future, we plan to develop optimal algorithms of image stitching and their efficient implementations on GPUs and other accelerators.

6. Acknowledgments

This work is supported in part by NASA (Grant No.: 12-2012EPSCoR-0026). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NASA. The authors would also thank Nvidia for GPU donations.

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[3] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communication of the ACM*, vol. 24, no. 6, p. 381Ű395, June 1981.

[4] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustmentâĂŤa modern synthesis,” in *Vision algorithms: theory and practice*. Springer, 1999, pp. 298–372.

[5] P. Jetley, L. Wesolowski, F. Gioachin, L. V. Kalé, and T. R. Quinn, “Scaling hierarchical n-body simulations on gpu clusters,” in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society, 2010, pp. 1–11.

[6] S. S. Hampton, S. R. Alam, P. S. Crozier, and P. K. Agarwal, “Optimal utilization of heterogeneous resources for biomolecular simulations,” in *proceedings of the 2010 ACM/IEEE international conference for high performance computing, networking, storage and analysis*. IEEE Computer Society, 2010, pp. 1–11.

[7] C. Yang, F. Wang, Y. Du, J. Chen, J. Liu, H. Yi, and K. Lu, “Adaptive optimization for petascale heterogeneous cpu/gpu computing,” in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*. IEEE, 2010, pp. 19–28.

[8] G. Chen, M. Kandemir, M. J. Irwin, and J. Ramanujam, “Reducing code size through address register assignment,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 5, no. 1, pp. 225–258, 2006.

[9] W. Li and Y. Zhang, “An efficient code update scheme for dsp applications in mobile embedded systems,” *ACM Sigplan Notices*, vol. 45, no. 4, pp. 105–114, 2010.

[10] S. Pedre, T. Krajnìk, E. Todorovich, and P. Borensztejn, “A co-design methodology for processor-centric embedded systems with hardware acceleration using fpga,” in *Programmable Logic (SPL), 2012 VIII Southern Conference on*. IEEE, 2012, pp. 1–8.

[11] J. Šváb, T. Krajník, J. Faigl, and L. Přeučil, “Fpga based speeded up robust features,” in *Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on*. IEEE, 2009, pp. 35–41.

[12] “Nvidia corporation, "variable smp (4-plus-1^{TM})," white paper v1.3, 2011, available online on,” http://www.nvidia.com.

[13] "Qualcomm," http://developer.qualcomm.com/discover/chipsets-andmodems/adreno-gpu.

[14] "Imagination technologies," http://www.imgtec.com/powervr/powervr-graphics.asp.

[15] Y.-C. Wang, B. Donyanavard, and K.-T. T. Cheng, "Energy-aware real-time face recognition system on mobile cpu-gpu platform," in *Trends and Topics in Computer Vision*. Springer, 2010, pp. 411–422.

[16] K.-T. Cheng and Y.-C. Wang, "Using mobile gpu for general-purpose computing–a case study of face recognition on smartphones," in *VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on*. IEEE, 2011, pp. 1–4.

[17] B. Rister, G. Wang, M. Wu, and J. R. Cavallaro, "A fast and efficient sift detector using the mobile gpu," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2674–2678.

[18] P. H. Torr and A. Zisserman, "Feature based methods for structure and motion estimation," in *Vision Algorithms: Theory and Practice*. Springer, 1999, pp. 278–294.

[19] D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 885–891.

[20] R. Szeliski and S. B. Kang, "Direct methods for visual scene reconstruction," in *Representation of Visual Scenes, 1995.(In Conjuction with ICCV'95), Proceedings IEEE Workshop on*. IEEE, 1995, pp. 26–33.

[21] M. Irani and P. Anandan, "About direct methods," in *Vision Algorithms: Theory and Practice*. Springer, 1999, pp. 267–277.

[22] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[23] M. Brown and D. G. Lowe, "Invariant features from interest point groups." in *BMVC*, no. s 1, 2002.

[24] A. Baumberg, "Reliable feature matching across widely separated views," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1. IEEE, 2000, pp. 774–781.

[25] T. B. Terriberry, L. M. French, and J. Helmsen, "Gpu accelerating speeded-up robust features," in *Proc. Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT)*. Citeseer, 2008, pp. 355–362.

[26] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics (TOG)*, vol. 2, no. 4, pp. 217–236, 1983.

Background Estimation Using Image Processing Technique

Maha Thafar [1], **Salwa Aljehane** [2]
[1,2] Department of Computer Science, Kent State University, Kent, OH, USA

Abstract **-** *Image Inpainting is the art of reconstruct missing areas. It is used to fix any damage in the images by removing the damage and filling this region based on estimating background. Background estimation is used to interpolate an estimate color of the background based on surrounding pixels' color as an image processing technique. The purpose of this paper is to implement an existing digital image inpainting algorithm. This code is proposed to be used as a plugin in image processing framework in order to be easily accessed and used to fill a missing region with smooth background color in reasonable way.*

Keywords: *Image Processing, Background Estimation, ImageJ, and Inpainting.*

1 Introduction

Nowadays, digital images are widely used in everyday life for different purposes. However, having a desirable image could be a challenge. Maintaining any damage in the image, or removing a selected object from the image, can be done using several methods to fill in the missing area, or to modify the damaged region. The most popular techniques are image inpainting and texture synthesis [1]. Image inpainting is a technique, which is used to fill or restore the area of removed object using the surrounding pixels to generate an appropriate color [2]. Texture synthesis is another technique to fill the missing image regions using texture information [3]. Background estimation technique is used for both static image and video [4].

In this paper, inpainting technique is explained and used in purpose of editing biological images such as estimating the background color after removing specific cell manually. In contrast, *Olivera* [1] applied mask on the whole image to remove the damage, and then fill the removing area using inpainting technique. In general, inpainting methods are done by the following steps. First, the desired region is selected automatically using different techniques such as color detection, or manually by the user. Then, specify the boundary of this region and clear its color. After that, the missing value will be computed to color this area. The last step is different from algorithm to algorithm. Nevertheless, this project used an existing method that presented in [1] and add some more steps to improve the results using some existing enhancement and filtering techniques.

The remaining sections of this paper are organized as follow: literature review is presented in Section 2. The project concept is discussed in section 3. Design and development stages are explained in section 4. Section 5 includes the implementation details. Then results and discussions are given in section 6. Finally the conclusion with future work are drawn in the last section.

2 Review of literature

There has been a significant amount of research that has been done in the digital image-inpainting in computer vision, computer graphics, and other computer science fields. They manipulate the inpainting approach in several processes. Many algorithms have been presented in this area such as Fast Marching Method Algorithm [5] , Exemplar-Based Image Inpainting [2], etc. There are different factors that affect the inpainting process such as missing region size, surrounding pixels, image type, and the speed of the algorithm.

Oliveira introduced a Fast Digital Image Inpainting algorithm, which used a convolution process [1]. In this method, the region to be inpainted is convolved with the kernel to compute the weighted average value based on the neighbor pixels' value. Two diffusion kernels were used in this method in order to compute the missing value. It works by convolving the boundary with the filter and propagating the new color toward the target region. However, one hundred iterations were used in the inpainting process to generate an accurate color [1].

Alexandra [5] used inpainting technique based on the Fast Marching Method (FMM). This algorithm starts from the boundary of selected region and goes inside gradually by filling every pixel in the boundary first. The pixel to be inpainted is replaced by normalized weighted sum of all the known pixels in the neighborhood. Selection of the weights is an important matter. Once a pixel is inpainted, it moves to next nearest pixel using Fast Marching Method. This method gave a good result with short run time.

Another study proposed a novel algorithm for digital image inpainting [6]. It is built based on mathematical theory of the Navier-Stoked equations [7] for fluids dynamic to apply image inpainting. This algorithm automatically transfers information into selected region. The main concept is: first travels along the edges from known regions to unknown regions. It continues isophotes, which means propagates the image Laplacian in the level-lines, while matching gradient vectors at the boundary of the inpainting region. Thus, some methods from fluid dynamics are used and the color is filled to reduce minimum variance in that area. The advantage of this approach is providing theoretical and numerical results [6].

3 Project concept and design

The project idea is based on outlining manually a specific area in a biological image such as cell, then removing everything inside the boundary by clearing the color. After that, filling that area with an estimation background color using neighbor pixels around the selected area as shown in figure 1. The last step is to make the colors blend seamlessly. In addition, many variables that affect the background should be taken into the account such as color, illumination, etc.

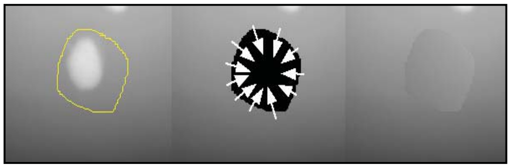


Figure 1: Background Estimation Concept

4 Design and development

4.1 Data

A biological image from biology department at Kent State University in TIFF extension (Tagged Image File Format) was used. This format is chosen among different image formats since it is the default format of ImageJ [8], which is used later in implementation stage.

4.2 Software

1) ImageJ: It is a public domain Java Image Processing program. It can display, edit, analyze, process, and save images. In addition, it can print 8-bit, 16-bit and 32-bit images. It can read many image formats. We will use this tool, which provides a lot of built-in or user- define macros and plugins that can be applied on the image [9].

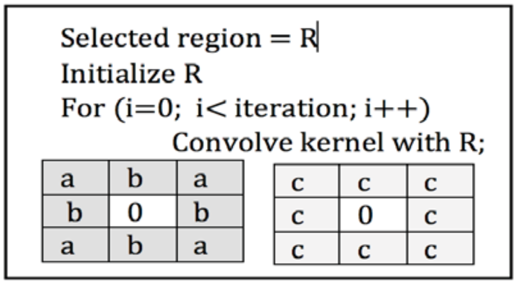
2) Eclipse: It is an Integrated Development Environment *(IDE). Eclipse* can be used to develop application, which is written by Java programming language and then export plugins to be used in different applications. All plugins are executed through *Eclipse* software.

Process:

1. Make a segmentation of target area manually
2. Remove this area or object
3. Clear the background
4. Fill the region by applying specific method
5. Improve the image by applying different filters.

5 Implementation

Our code is developed based on Fast Digital Image Inpainting Algorithm using Convolution Based Method [1]. As most inpainting algorithms, four steps were used to perform this task. The first step is allowing the user to select the desired region by using any selection tools. The second step is detecting this region which called Region of Interest (*ROI*). The third step is initializing the ROI by clearing its color using clear function. The final step is generating the color based on the surrounding pixels to inpaint the target region. The main idea to generate the color depends on convolution operation. A kernel with specific coefficients' values that were given in Oliveira's paper [1] is used. These coefficients are a = 0.073235, b = 0.176765, c = 0.125. The most important instructions in the pseudocode and two diffusion kernels that used in the convolution operation are shown in figure 2.



```
Selected region = R
Initialize R
For (i=0;  i< iteration; i++)
        Convolve kernel with R;
```

a	b	a
b	0	b
a	b	a

c	c	c
c	0	c
c	c	c

Figure 2: Diffusion kernels and pseudocode

The kernel is convolved with the ROI and generates the color starting from the border inward to the region as shown in figure 3.

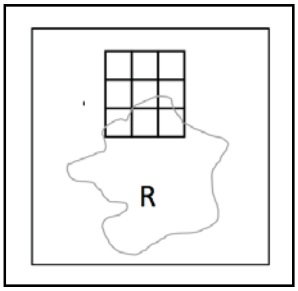


Figure 3: Convolution with kernel

The developed code is divided into three major parts as the following:

Part 1: Accesing the *ImageJ* framework for image processing and its plugins through *Eclipse*. The image is considered as variable to be read.

Part 2: Detecting the ROI and getting its information, such as the start point coordinates, the width, and the height of the ROI. These parameters are sent to the kernel convolution method.

Part 3: *Kernel-Convolution* method is the main part of the code. It receives the information including: start point of the ROI, and its width and height then starts to do inpainting. The process of this method is iterative. It applies the convolution kernel on each pixel in a spiral- like fashion, and then shrinks the border by going inside ROI. The last step of the project is applying ImageJ filters such as Mean and Gaussian blur filters on the ROI to improve and smooth the result. All these steps are shown in figure 4.

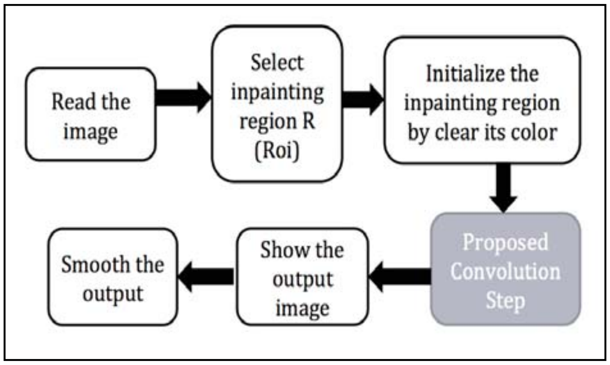

Read the image
Select inpainting region R (Roi)
Initialize the inpainting region by clear its color
Proposed Convolution Step
Show the output image
Smooth the output

Figure 4: Implementation Flowchart

The code is exported to be an inpainting plugin, which is used through ImageJ framework.

6 Results and discussion

We applied the plugin on biological image using ImageJ framework. Different region of interests were tested and the results are shown in figures 5 and figure 6. The result is better if the user selects a small area to be inpainted rather than larger area.

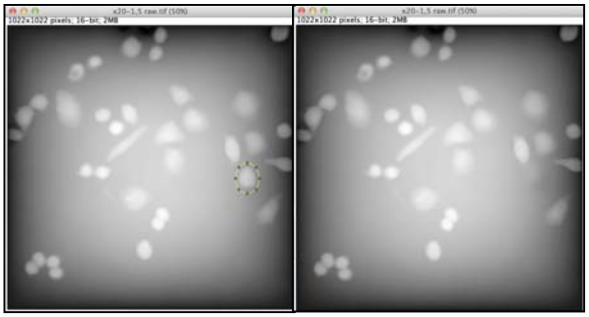


Figure 5a: ROI selection on original image

Figure 5b: result of inpainting

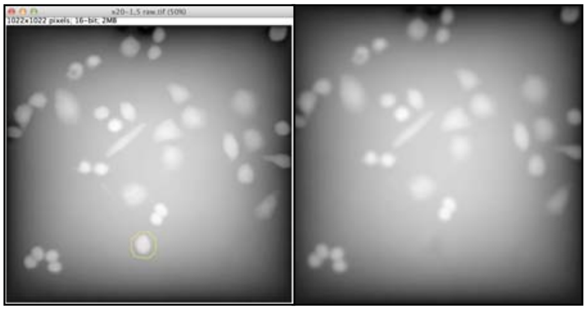


Figure 6a: ROI selection on original image

Figure 6b: result of inpainting

7 Conclusion and future work

In this paper, an image inpainting implementation based on Oliveira algorithm has presented. However, generating the background color of removing area is done based on surrounding pixels colors. Some filters are applied to improve the result such as removing the blur. ImageJ and Eclipse java were used to perform this task.

For future works, we are looking to improve the result to be more accurate. In addition, we want to include different filters while designing the plugin. This is useful to increase the performance and minimize user interaction to do everything automatically.

8 References

[1] Richard, Manuel M Oliveira Brian Bowen and M. Y. Chang, "Fast digital image inpainting," in *Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain,* 2001, pp. 106-107.

[2] T. K. Shih, N. C. Tang and J. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *Circuits and Systems for Video Technology, IEEE Transactions On,* vol. 19, pp. 347-360, 2009.

[3] P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *Multimedia, IEEE Transactions On,* vol. 13, pp. 453-465, 2011.

[4] M. Granados, K. I. Kim, J. Tompkin, J. Kautz and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *Computer Vision–ECCV 2012*Anonymous Springer, 2012, pp. 682-695.

[5] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools,* vol. 9, pp. 23-34, 2004.

[6] M. Bertalmio, A. L. Bertozzi and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference On,* 2001, pp. I-355-I-362 vol. 1.

[7] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques,* 2000, pp. 417-424.

[8] W. Bailer, "Writing ImageJ plugins—a tutorial," *Upper Austria University of Applied Sciences, Austria,* 2006.

[9] T. Ferreira and W. Rasb, "ImageJ user guide," 2012.

Automatic Damaged Region Detection and Inpainting Method for Digital Images

C. Martin[1*] **and M. Allali**[1]

[1]School of Computational and Data Sciences, Chapman University, 1 University Drive, Orange, CA 92866, USA

Abstract - *In this paper we examine three worthy inpainting techniques and present the beginning stages of our proposed hybrid technique that minimizes required user input.*

Keywords: Image processing, inpainting, diffusion barrier, SVD

1 Initial Exploration and Research

To gather an intuition of what inpainting means within the constraints of a discretized programing domain, we explored the generic heat transfer equations represented as difference equations. Before difference equations are applied to the target region, a two-pixel thick boundary immediately surrounding the region is utilized to fill in the initial one-pixel thick interior of the region. In order to force the information from the outside of the region into the inside of the region, given edge location in relation to the "origin" of the image, the discrete double derivative equations must be altered counter intuitively. As shown in Figure 1, for an image $I = I(x, y)$, the top and left edges may be approximated using equations (i) and (ii). However, for the bottom and right edges, the application is reversed, equations (iii) and (iv).

$$\textit{Top edge: } I(x,y) = 2I(x-1,y) - I(x-2,y), \quad \textit{where } x = i \textit{ is constant and } y = j{:}\, j+n \qquad \text{(i)}$$

$$\textit{Left edge: } I(x,y) = 2I(x,y-1) - I(x,y-2), \quad \textit{where } y = j \textit{ is constant and } x = i{:}\, i+m \qquad \text{(ii)}$$

$$\textit{Bottom edge: } I(x,y) = 2I(x+1,y) - I(x+2,y), \quad \textit{where } x = i+m \textit{ is constant and } y = j{:}\, j+n \qquad \text{(iii)}$$

$$\textit{Right edge: } I(x,y) = 2I(x,y+1) - I(x,y+2), \quad \textit{where } y = j+n \textit{ is constant and } x = i{:}\, i+m \qquad \text{(iv)}$$

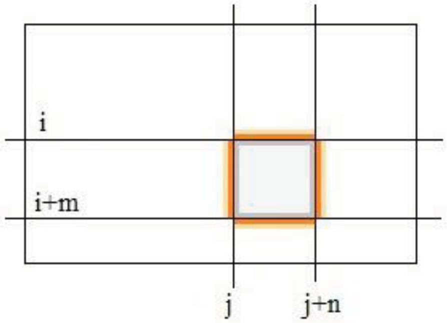


Figure 1

One may expect the results from applying the heat transfer algorithm to the damaged region based on this newly-set pixel-wide boundary would offer a faded appearance to the inpainted region. However, although the new image is not perfect, particularly texture-wise, it is striking nonetheless considering the simplicity of the algorithm. The color matching is superb. Aside from the obvious texture contrast, the flow of shadow and highlight throughout the inpainted region is satisfactory per the human visual system.

A paper with good results and reasonable processing times called, "Fast Digital Image Inpainting" [1], from 2001, uses an algorithm analogous to the discrete version of convolution. The kernels suggested and used include a Gaussian kernel (isotropic diffusion/linear heat equation) and a weighted average kernel. Both kernels consider values from the eight nearest pixels, although an iterative approach, slightly different from the heat transfer algorithm, is used. We found the results from utilizing the heat transfer code comparable to the results from convolving the region to be inpainted with the rotation invariant kernel in Figure 2. Figure 3 shows part of an image with a horizontal five pixel wide black bar added to it as a stand-in for damage. The results from using one thousand iterations of finite difference approximations and one thousand iterations of convolution are shown in Figure 4 and Figure 5, respectively. The stone pattern comes from source [2]. One noticeable difference between the two algorithms is in the appearance of the noise each process leaves behind. Magnifying the inpainted regions shows a grainy appearance that favors a

$$\boxed{\frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}}$$

Figure 2

[*] Corresponding Author: Chloe Martin, marti192@mail.chapman.edu

vertical drift for the heat transfer program and blurring for the diffusion kernel in Figure 2. Although the two methods offer visually comparable results, the heat code runs remarkably faster than the convolution code.

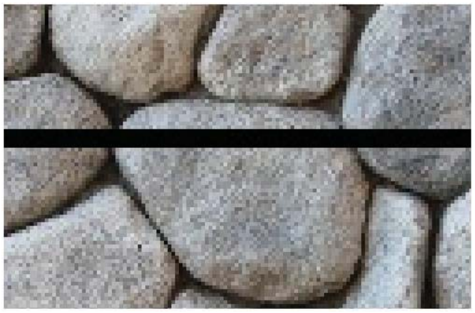


Figure 3

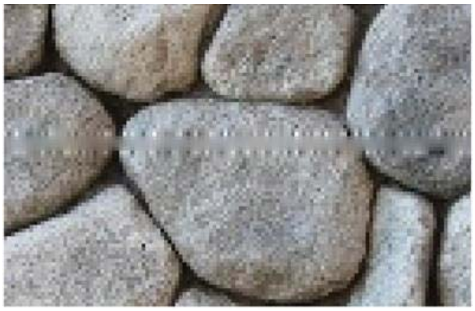


Figure 4

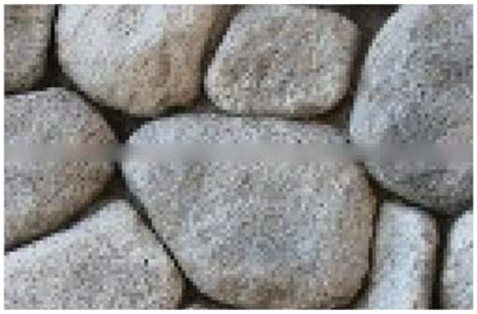


Figure 5

2 Diffusion Barriers

The most pleasing results in [1] occurred with the implementation of one of the above filters combined with diffusion barriers. Diffusion barriers are effective at preventing blurred regions in resulting images due to intersections between the region to be inpainted and high-contrast edges. The results from implementing the paper's pseudo-code algorithm in Matlab are given. Figure 6 shows a simple image with two perpendicular lines simulating damage, a generic mask, and a mask including two user-defined diffusion barriers. Figure 7 shows the results from 100 iterations of convolving the masked portion of the image with the kernel in Figure 2. The circled regions indicate where the inpainting mask crosses high-contrast edges. Figure 8 shows the results from 100 iterations convolving the mask and diffusion barriers team with the kernel in Figure 2. Unfortunately, we were not satisfied with the diffusion barriers technique when implementing a dark, two-pixel wide barrier as was utilized in [1]. The results from implementing diffusion barriers did not appear to offer an improvement when compared to using the inpainting mask sans diffusion barriers. Using dark, thin diffusion barriers worked well in [1]. It is possible that we misunderstood the algorithm description or were not as precise in the creation of our mask and diffusion barriers team.

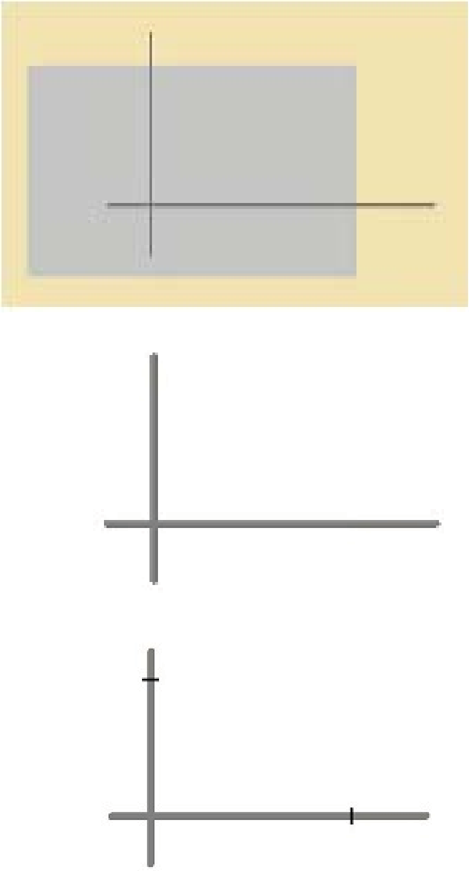


Figure 6

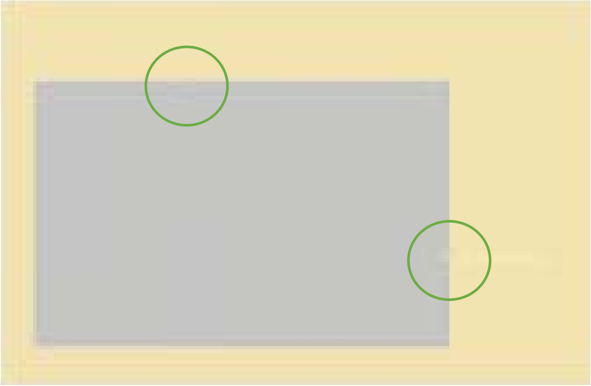


Figure 7

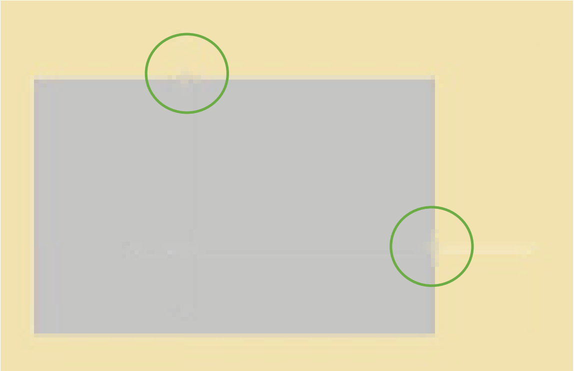


Figure 8

Our program worked best when we used white, linear, diffusion barriers five pixels wide. This is reasonable because the algorithm takes into consideration a pixel on either side of the center pixel when performing convolution with a 3 × 3 kernel. A two-pixel wide barrier, such as the one described in [1], is not sufficient for our program since there would still be information spilling over edges and thus causing blurring. Having a white diffusion barrier is ideal because the program associates a pixel value of one, or white, with a part of the image to be preserved. This means that although the pixels surrounding the barrier will undergo the inpainting process, the parts of the image beneath the barrier will remain relatively intact and therefore the edge will appear more prevalent than without such devices.

The credibility of our analysis is based on results from applying the diffusion barrier technique to a damaged photograph of Abraham Lincoln [3]. Figure 9 shows the damaged photograph. Figure 10 shows the isolated region to be inpainted (left) and the isolated region to be inpainted with two white diffusion barriers straddling the contrast between Abe's dark hair and the light background (right). Figure 11 and Figure 12 show the difference between the two results. As can be seen in Figure 12, the inpainted edges in the image on the right are more well-preserved than the inpainted edges in the image on the left.

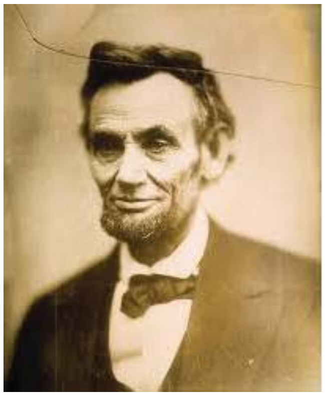


Figure 9

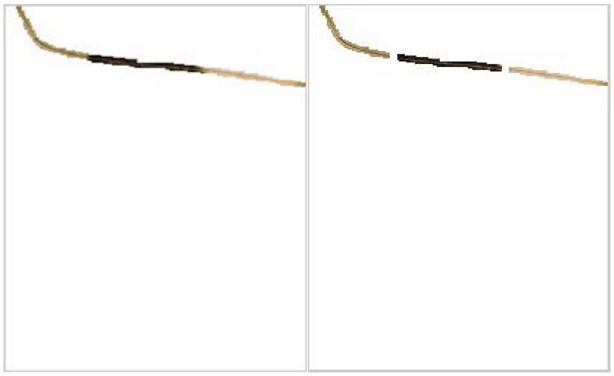


Figure 10

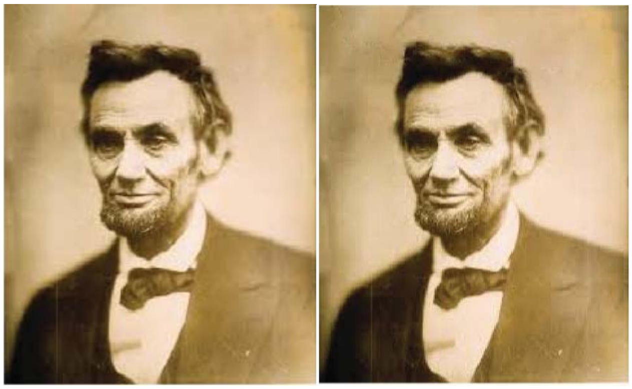


Figure 11

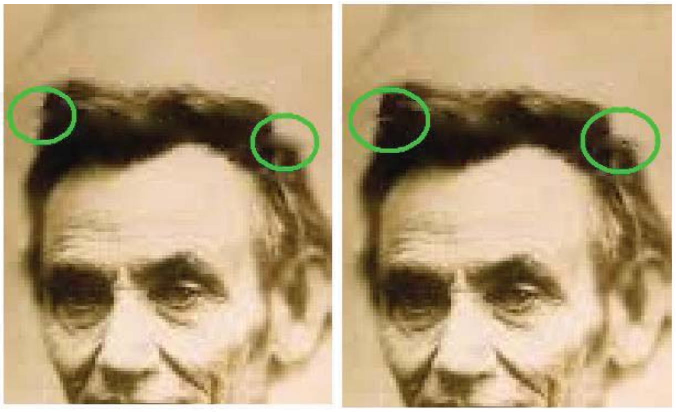


Figure 12

These barriers sound like a promising solution to information from one side of an edge spilling over into the other side thus causing a blurring effect. The frustration occurs in the amount of user-necessary interaction with the inpainting algorithm. It seems that the more desirable the results are, the more human interference is needed to account for such complexities. Every image is unique, so not only does the user need to define and initialize the mask, they also need to include, either within the mask itself or otherwise, diffusion barriers. It would be ideal if the user only needed to define the mask then rely on the algorithm to conclude where edge reconstruction is necessary based on high-contrast intersection points.

3 Total Variation Minimization

A fairly technical and mathematical paper that we looked at called "Total Variation Wavelet Inpainting" regards a digital image as a vector valued function [4]. The authors utilize a technique known as total variation (TV) minimization to resolve issues surrounding inpainting. It appears that TV minimization is analogous to arc length minimization for an oriented curve. What makes this paper different from most of the others that we read, is that the inpainting process is guided by the wavelet domain rather than the pixel domain. In the pixel domain, it is necessary to assume a decoupled relationship between pixels that are a threshold distance away from each other for the sake of de-noising. However, wavelets inpainting requires a forced relationship between wavelet regularities to correlate the missing and existing components [4]. In a noiseless image, the proposed model simply fills in the missing wavelet coefficients based on TV minimization. Alternatively, if the image is noisy, a second model for wavelet inpainting is used.

4 Singular Value Decomposition

In an effort to discover a realistic solution to human intervention in inpainting projects, we regarded a paper called, "SVD Based Automatic Detection of Target Regions for Image Inpainting" [5]. As the title indicates, the authors use singular value decomposition to select inpainting target regions thus diminishing the need for manual creation and introduction of a mask. The authors use a sliding window technique that compares two adjacent pixels based on m by n patches of neighboring pixels. The following conceptualizations are based on our understanding of the written description provided by the authors. For adjacent pixels, $p = I(i,j)$ and $q = I(i,j+1)$, where $m = n = 3$, Equations (v) and (vi) represent the 3 by 3 patches associated with each pixel. Notice that there is a 3 by 2 patch of overlapping pixels. The column vectors v_p and v_q are comprised of the reshaped patches as can be seen in Equations (vii) and (viii). Once the 9 by 2 matrix A is constructed for patches ϕ_p and ϕ_q, singular-value decomposition is applied to A. Fortunately, matrices U, V, and $\sum$ can be generated using the built-in Matlab function, $[U,S,V] = svd(A)$. There are only two singular values in $\sum$ on the diagonal and therefore, $\sum$ can be reduced from a 9 by 2 matrix to a 2 by 2 matrix making it possible to find $V\sum$. Now, the similarity between the patches ϕ_p and ϕ_q is given by the cosine of the angle between w_1 and w_2, where w_1 and w_2 are the rows of $V\sum$. Equation (x) shows this similarity measure.

$$\phi_p = \begin{bmatrix} I_{i,j} & I_{i,j+1} & I_{i,j+2} \\ I_{i+1,j} & I_{i+1,j+1} & I_{i+1,j+2} \\ I_{i+2,j} & I_{i+2,j+1} & I_{i+2,j+2} \end{bmatrix} \quad \text{(v)}$$

$$\phi_q = \begin{bmatrix} I_{i,j+1} & I_{i,j+2} & I_{i,j+3} \\ I_{i+1,j+1} & I_{i+1,j+2} & I_{i+1,j+3} \\ I_{i+2,j+1} & I_{i+2,j+2} & I_{i+2,j+3} \end{bmatrix} \quad \text{(vi)}$$

$$v_p = [I_{i,j} \quad I_{i,j+1} \quad I_{i,j+2} \quad \cdots \quad I_{i+2,j+2}]^T \quad \text{(vii)}$$

$$v_q = [I_{i,j+1} \quad I_{i,j+2} \quad I_{i,j+3} \quad \cdots \quad I_{i+2,j+3}]^T \quad \text{(viii)}$$

$$A = [v_p \quad v_q] \quad \text{(ix)}$$

$$cos(\theta_{rs}) = \frac{w_1 \cdot w_2}{\|w_1\|\|w_2\|} \quad \text{(x)}$$

The authors describe comparing the overlapping portion of the patches for similarity by creating a set E_p such that E_p contains the neighborhood of $p = I_{i,j}$; pixels $I_{i+1,j}$ and $I_{i,j+1}$.

5 Early Stage Hybrid Method Proposal

Our algorithm is initially based on our interpretation of the algorithm in [5]. We chose to use SVD to compare adjacent pixels but found that this primarily detected vertically damaged regions. We altered our program to detect horizontal and diagonal regions as well by including a third vector, w_3, and using a modified similarity metric.

$$\phi_t = \begin{bmatrix} I_{i+1,j} & I_{i+1,j+1} & I_{i+1,j+2} \\ I_{i+2,j} & I_{i+2,j+1} & I_{i+2,j+2} \\ I_{i+3,j} & I_{i+3,j+1} & I_{i+3,j+2} \end{bmatrix} \quad \text{(xi)}$$

$$v_t = [I_{i+1,j} \quad I_{i+1,j+1} \quad I_{i+1,j+2} \quad \cdots \quad I_{i+3,j+2}]^T \quad \text{(xii)}$$

$$A = [v_r \quad v_s \quad v_t] \quad \text{(xiii)}$$

$$cos(\theta^*_{rst}) = cos(\theta_{rs})\, cos(\theta_{st})\, cos(\theta_{rt}) \quad \text{(xiv)}$$

In our method, $\sum$ is resized to be a 3 by 3 diagonal matrix with the singular values of A on the diagonal. Since $V\sum$ now has three rows, a third vector, w_3, is introduced. The similarity measure used in our program is an evolved version of (x) given by (xiv). Next, we create a 3 by 3 matrix, occasionally a 4 by 4 matrix for experimental comparison, for each primary pixel $I_{i,j}$ consisting of a subset of associated values calculated with $cos(\theta^*_{rst})$. Finally, a similarity matrix of size I is produced where each element is calculated as a sum of all values contained within each subset of $cos(\theta^*_{rst})$. The values of this similarity matrix are then compared with a threshold value, δ which is proportionally based on the size of ϕ_r. Experimentally, we found that a universal $\delta \approx \frac{1}{mn} * 0.97$ to be effective for most cracked or discolored damaged photographs. We did however experiment with varying this ratio both manually and within the program automatically.

6 Experimental Results

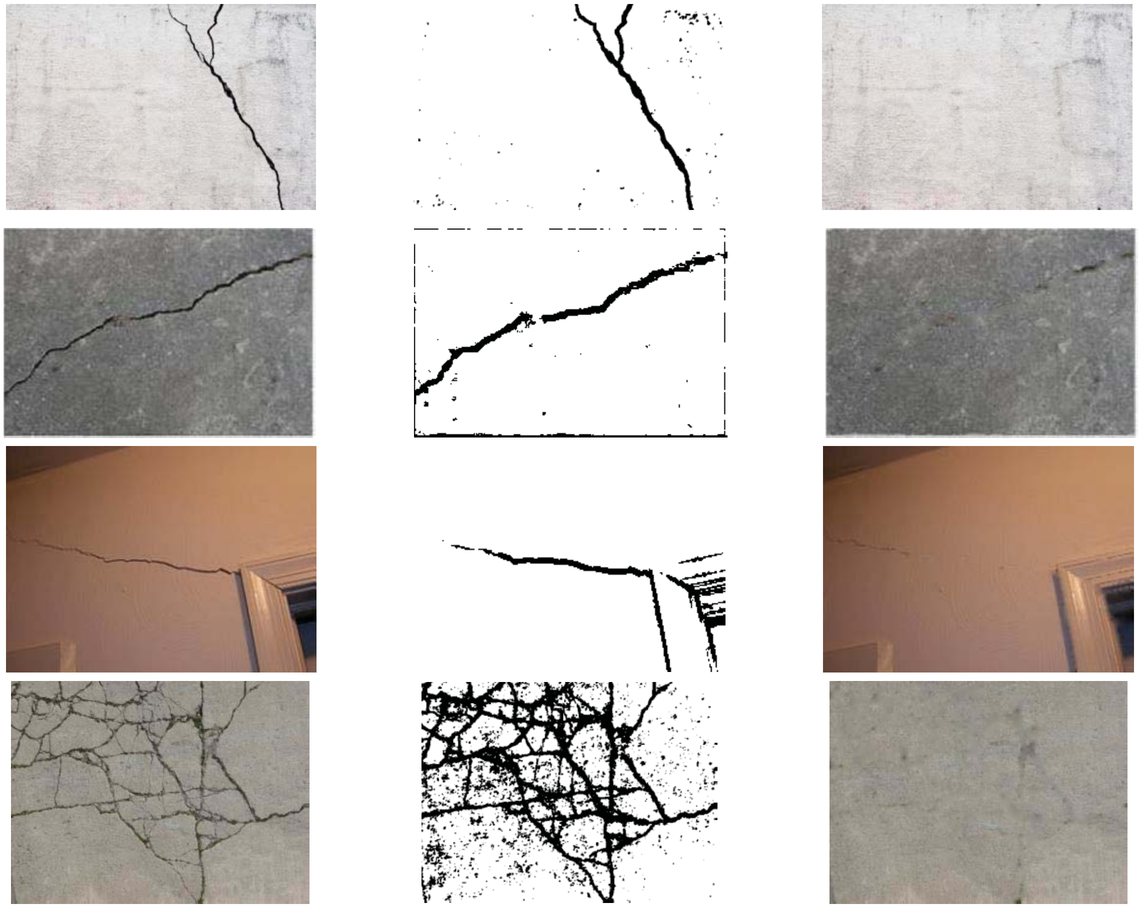


Figure 13

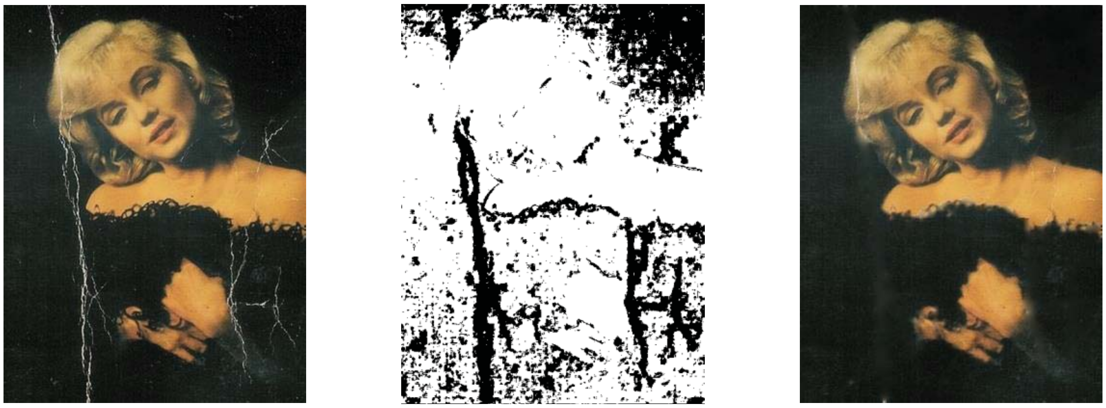


Figure 14

We sought to examine the usefulness of our proposed method with regard to both scanned damaged photographs and digital images of cracks in walls and pavement. Some of our preliminary results are catalogued in this paper. We believe that with more experimental exploration and mathematical thoughtfulness a successful hybrid method that does not rely on manual boundary detection is possible for many different types of images containing noise or damage. The images utilized experimentally and featured in this paper were downloaded from the internet [6]. Figure 13 shows images of damaged walls and pavement (left column), damaged region detection output (center column), and the results from using a simple inpainting algorithm based on heat diffusion (right column). Figure 14 shows a scanned image of a damaged photograph (left) and the results from processing the image in the same manner as was performed in Figure 13. In order to create a mask that is large enough to encompass the damaged regions, the program tends to over-detect. This is particularly troublesome when it comes to images of people. Figure 15 does seem to represent a small visual improvement from the original scratched photograph, however, the blurred appearance around the eyes and mouth is unsuitable for portrait preservation. Although not ideal, we experimented with manually adjusting the mask in order to prevent such detrimental results and fill in portions of the mask that merely outlined damage. The results are shown in Figure 16. Although this type of intervention is not ideal for a method that seeks to limit manual mask creation, it does inspire potentially automatic algorithms. Also, we found that in cases where manual mask creation is necessary, it makes preprocessing faster and more precise. Figure 17 and Figure 18 show results from applying this same partially user dependent method. Figure 17 represents the automatically detected regions and inpainting results. Figure 18 represents the tweaked mask and inpainting results. The original damaged image can be found in [7].

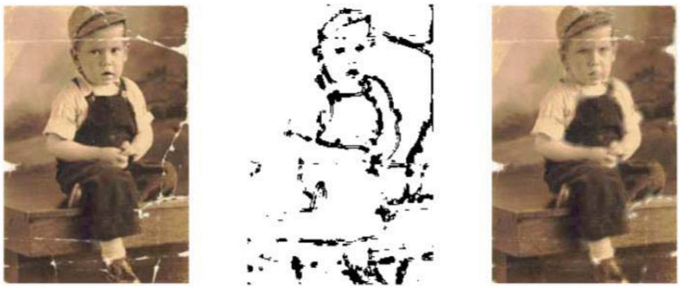


Figure 15

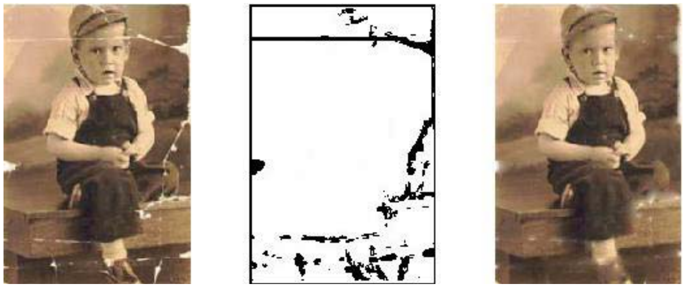


Figure 16

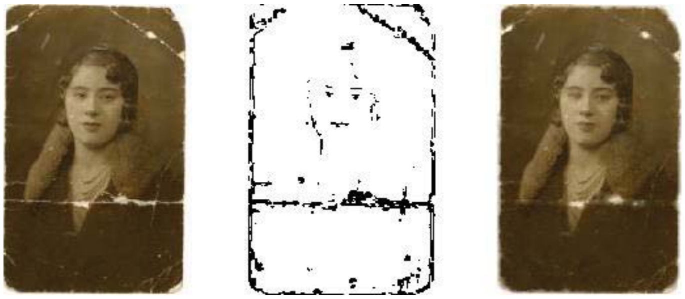


Figure 17

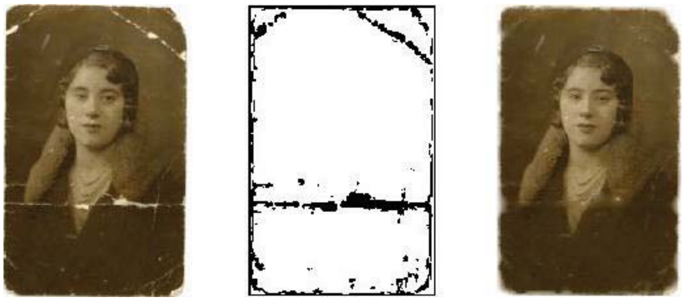


Figure 18

7 Future and Improvements

There is certainly a compromise concerning image quality. For example, in order to successfully detect the damaged regions without user intervention, the threshold for comparison may be such that the algorithm will “over-detect”. This means that although the program capably locates the regions in need of inpainting, it may also include non-damaged areas. Since we use the automatically detected regions as our boundary masks when implementing our inpainting algorithm, the repaired image may appear foggy or grainy in places that were previously clear. This issue occurred commonly around the eye area in images of human subjects. In the future, we would like to create an algorithm, or group of algorithms, that successfully identifies damaged regions, produces a mask and barrier team, and applies inpainting techniques with minimal human interaction.

8 Conclusion

We opted to research and experiment with digital color image inpainting techniques. We focused mainly on the three papers; [1], [4], and [5]. We implemented the beginnings of an automatic detection and inpainting algorithm gaining inspiration from the above papers. Desiring to find a technique that performs well independent of user intervention, we utilized singular value decomposition with the reduction suggested in [5]. Once the damaged region is detected, the damaged image and the detected region are passed into a function that treats the detected region as a mask and applies convolution with a pre-specified kernel. We opted to experiment with various kernels, patch sizes (ϕ in [5]), and similarity thresholds. Our goal is to build a program that is powerful enough to detect horizontal, vertical, and diagonal damage, pinpoint where edge reconstruction is necessary based on high-contrast intersection points, and apply inpainting techniques accordingly. Although there is plenty of work already being done in this field, we would like to continue exploring image reconstruction techniques, particularly those that do not involve or require user intervention.

9 References

[1] M. M. Oliveira, B. Bowen, R. McKenna and Y.-S. Chang, "Fast Digital Image Inpainting," in *Proceedings of the*

International Conference on Visualization, Imaging and Image Processing, Marbella, 2001.

[2] "TextureX.com," 2013. [Online]. Available: http://www.texturex.com/Stone-Textures/Stone+Texture+wall+large+rock+grey+image.jpg.php. [Accessed December 2015].

[3] "kottke.org," 3 December 2013. [Online]. Available: http://kottke.org/tag/Abraham%20Lincoln. [Accessed 17 December 2015].

[4] T. F. Chan, J. Shen and H.-M. Zhou, "Total Variation Wavelet Inpainting," *Journal of Mathematical Imaging and Vision,* vol. 25, no. 1, pp. 107-125, 2006.

[5] M. G. Padalkar, M. A. Zaveri and M. V. Joshi, "SVD Based Automatic Detection of Target Regions for Image Inpainting," in *Asian Conference on Computer Vision; Computer vision - ACCV 2012 Workshops*, Daejeon, 2012.

[6] "Google Images," [Online]. Available: http://www.images.google.com. [Accessed 2016].

[7] "Photo Valet: Photo Retoration," [Online]. Available: http://www.photovalet.co.uk/Torn_or_scratched_photos.html. [Accessed March 2016].

An Educational Tool for Understanding Discrete Fourier Transforms

Leonidas Deligiannidis

Wentworth Institute of Technology
Department of Computer Science and Networking
550 Huntington Avenue
Boston, MA 02115 USA
deligiannidisl@wit.edu

Abstract
In this paper we present a light-weight Image Analysis tool targeted at undergraduate students. More specifically, our tool enables undergraduate students to interact with the application in real time to understand how Fourier Transforms work (FT). This is achieved by displaying intermediate results in different windows in the application's canvas, while the user interacts with the tool's controls. The user is also able to add functionality to the tool or modify existing capabilities in order to understand better the way FTs work. This way, the students can introduce and experiment with new algorithms which can be implemented programmatically and incorporated in the tool's library.

Keywords: Discrete Fourier Transforms, Image Processing, Cross Correlation.

1. Introduction

A Fourier Transform (FT) of an N-dimensional signal is its transformation from its original time or space domain into a representation in the frequency domain and vice versa [1]. FTs take as input a signal and convert it into the frequencies of the waves that the signal is composed of. These transforms can be applied to 1D signals such as signals form sensors, audio signals etc., 2D signals such as images or video, or in general N-dimension signals. FTs are reversible, meaning that a signal from the time domain can be transformed to the frequency domain, and then back to its original state. Signal operations/filtering can be performed in either the time or the frequency domain. However, sometimes it is easier and faster to perform the same operation in the frequency domain. The relationship between the Cartesian and the Polar form is given by Euler's Formula:

$$e^{\omega i} = \cos(\omega) + i\sin(\omega)$$

In essence, we can represent any signal as a sum of sinusoidal waves. Based on this, the Discrete Fourier Transform (DFT) of a signal is defined as:

$$F(k) = \sum_{n=0}^{N-1}\left(\mathrm{f(n)} * e^{-i2\pi kn/N}\right)$$

And its inverse (moving from the frequency domain, back to time domain) as:

$$f(n) = \frac{1}{N}\sum_{k=0}^{N-1}\left(F(k) * e^{i2\pi kn/N}\right)$$

Where "N" is the number samples and "n" and "k" are the current sample and frequency respectively.

We should note that the Fast Fourier Transform (FFT) is an improved method for calculating the DFT but has a requirement that the size of the signal (number of samples) should be of power of two [2]. There are many tools that incorporate DFTs for signal and image analysis. ImageJ [3] is a great free tool and provides a large number of operations that can be performed to an input signal as well as it provides a large number of filter. ImageJ also provides a mechanism to add functionality through plugins. Our tool however, is designed with the focus of enabling students to implement the filters. We provide the framework where additional filters and image analysis techniques can be implemented and tested in an interactive way before installing them and making them part of a custom library. Python, for example, provides most of this functionality but the implementation is hidden from the user. Complex operations in Python can be performed very easily within a few lines of code. This however, hides the details of the implementation. Our tool exposes these details of the implementation and thus provides a mechanism where students dive-in in order to comprehend the implementation specifics. Our framework is built around FFTW which is a freely available [4] C library for computing the discrete Fourier Transform (DFT) for one or more dimensions [5]. Our tool converts images from the time domain to the frequency domain, and back, using FFTW. The user can get handles to the frequencies which the images are composed of, and implement his or her filtering. Filtering or any other operation in the frequency domain can be performed programmatically.

This can be done by implementing Java or C++ methods. Hooks for interactivity are provided by the framework to dynamically adjust parameters of the user's implemented algorithms.

2. Implementation

The framework of our tool is built around the FFTW [4] library and we chose to use it because of its great speed performance. FFTW is a C library. Our application is built in Java with bindings to FFTW. The user, however, can create filters in either language. One of the sample applications is shown in Figure 1. Two images are loaded, one being the template. Both images are transformed into the frequency domain and their cross correlation map is displayed in another window. The user can apply a low-pass or a high-pass Gaussian filter and interactively change its parameters while observing the results in real time. The user can apply a low-pass or a high pass Gaussian filter on either the template or the original image in either the time domain or the frequency domain. In the figure 1 we apply the filter in the frequency domain:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

We multiply, in the frequency domain, the Fourier transform of one with the complex conjugate of the other [6]. The output of this operation is a number which indicates the confidence of the match and the x, y coordinates or offset of where the template image should be moved to align it with the original image. The phase correlation is calculated as:

$$R = \frac{F\bar{G}}{|F\bar{G}|}$$

F and G are the FFT of the image and the template respectively. $\bar{G}$ is the complex conjugate of the template image (it could be of the original image as well).

References

[1] Steven W. Smith. "The Scientist and Engineer's Guide to Digital Signal Processing". San Diego, California: California Technical Publishing, 1999.

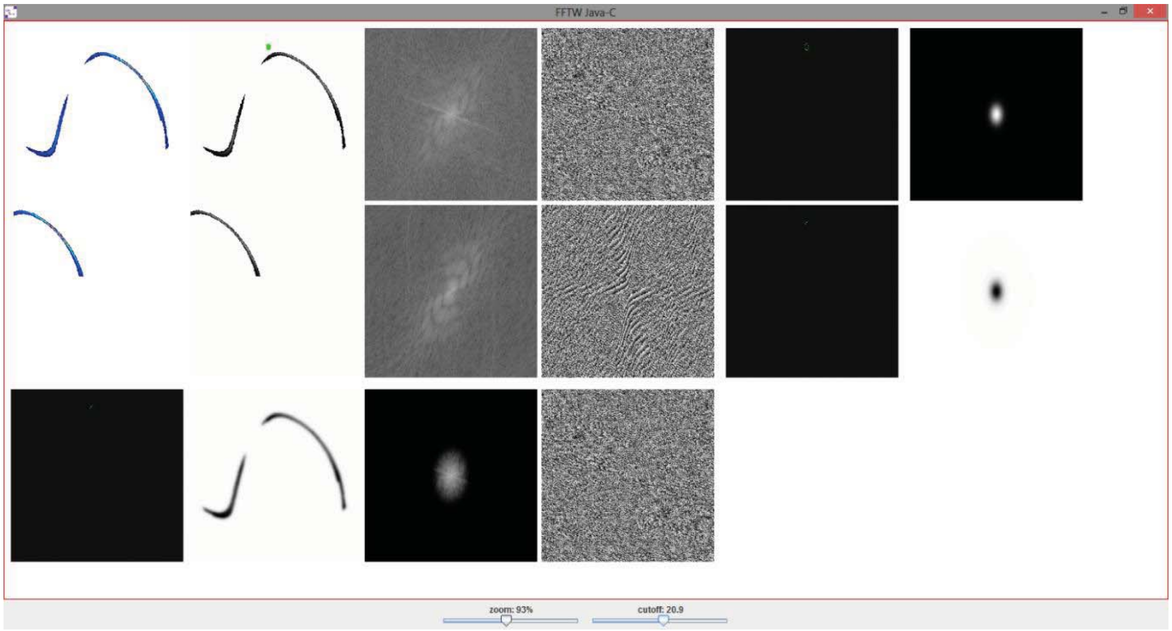
[2] P. Duhamel, and M. Vetterli, "Fast Fourier Transforms: A Tutorial Review and a State of the Art" Signal Processing, vol. 19, pp. 259-299, Elsevier Apr. 1990.

[3] Home page of the ImageJ (An Image Processing and Analysis tool in Java). http://imagej.nih.gov/ij/. Retrieved Feb. 2016.

[4] Fastest Fourier Transform in the West (FFTW) download page: http://www.fftw.org/download.html. Retrieved Feb. 23 2016.

[5] Matteo Frigo and Steven G. Johnson, "The Design and Implementation of FFTW3," Proceedings of the IEEE Vol. 93 (2), 216–231 (2005). Invited paper, Special Issue on Program Generation, Optimization, and Platform Adaptation.

[6] Rafael C. Gonzalez, Richard E. Woods. "Digital Image Processing" Third edition. Pearson Prentice Hall, 2008. ISBN: 978-0-13-168728-8.



zoom: 93%

cutoff: 20.9

Figure 1. Main (customizable) window. It shows the image loaded (top left), and right below it the template image. In the middle, we see the frequency spectrum and the phase. To the right we see the low-pass and high-pass Gaussian filters. The bottom row shows the cross correlation map, the image after it was filtered, along with the filtered frequency spectrum and phase. In the second sub-window at the top-left, we show with a green circle where the template image should move to align with the original image.

Accelerating DCT-based color image watermarking on GPUs

Hong Fan[1], **Miaoqing Huang**[2], **Chenggang Lai**[2], **Jinming Yu**[1], **and Wujun Xu**[1]
[1]School of Information Science & Technology, Donghua University, Shanghai, China
[2]Department of Computer Science and Computer Engineering, University of Arkansas, Arkansas, USA

Abstract— *Watermarking is widely used to protect information authenticity and integrity. In this work we propose to add invisible watermark into one channel of color RGB images in the frequency domain. Using Discrete Cosine Transform (DCT) in watermarking can result in better watermark quality and robustness due to the use of uncorrelated coefficients that operate in the frequency domain. The watermarking process consists of three steps, i.e., DCT, adding watermark, inverse DCT (IDCT). Applying DCT/IDCT on large images using CPUs typically takes quite long time, which prevents the online integration of invisible watermarks. GPUs can significantly improve the performance of the watermarking process. In our experiments, it is found that Nvidia Tesla K20 GPU is able to add watermark to one 512×512 image in less than 0.1 ms, which is 170 times faster than the Intel Core i7-3820 CPU. The energy-efficient GPU on Tegra TK1 SOC can achieve more than 20× speedup than the ARM CPU.*

Keywords: Color image invisible watermarking; GPU; DCT

1. Introduction

In images, a watermark is a pattern inserted in the image to copyright it. The development of networked multimedia systems is conditioned by the development of efficient methods to protect data owners against unauthorized copying and redistribution of the material put on the network. The watermark is intended to be permanently embedded into the digital data so that authorized users can easily read it. To be really effective, a watermark should be unobtrusive, readily extractable, robust, unambiguous and innumerable [1]. Image watermarking techniques proposed so far can be divided into two main groups: those embedding the watermark directly in the spatial domain and those operating in a transformed domain, e.g., the frequency domain. There are two ways to apply the watermarking in images. The first one is called visible watermark. The characteristic of this type is that you can see the watermark over the image like a logo or a sign. The second one is the invisible watermarking. This type of watermarks cannot be perceived by human eyes. It is necessary to use electronic devices for inserting or extracting the invisible watermarks [2]. Many algorithms proposed in literatures are complicated in computation, which restrains their adopting in some real-time occasions. Improving the efficiency of image authentication process has become one of the challenges in this field. The two-dimensional variation of the transform that operates on 8×8 blocks (DCT8×8) is widely used in image and video watermarking [3] because it exhibits high signal decorrelation rates and can be easily implemented on the majority of contemporary computing architectures.

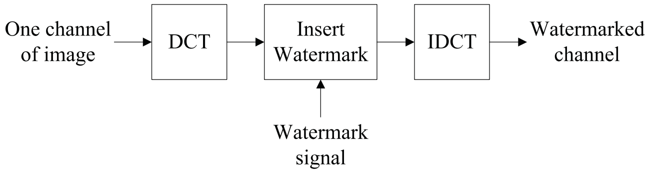


Fig. 1: Add watermark into one channel of the digital images.

Some researchers have used hardware such as FPGA and custom ASIC as co-processors in image processing. On the other hand, nowadays, GPU devices are wildly equipped in personal computers and have a great potential in providing massively parallel computing capability. As the release of NVIDIA's Compute Unified Device Architecture (CUDA), researchers can design programs for both CPU and GPU conveniently with a C-like programming language, without knowing fundamental knowledge on computer graphics. The massive parallelism on GPU for general-purpose problems has arrived as a cheap and feasible solution to accelerate the process. There have been some works in which GPUs have been used in watermarking for image and video purposes [4]–[6]. In this work, we propose to insert the watermark to only one channel of a color digital image and to process it faster using one channel data (red or green or blue). Further, we take the advantage of the GPU technology for a short processing time.

The remainder of the paper is organized as the follows. In Section 2 we introduce the watermarking approach applied on one channel of the color images. In Section 3 we discuss the experiments on two different platforms, i.e., one workstation and one embedded system. We also compare the results between the CPU implementations and the GPU implementations. We provide the conclusions in Section 4.

2. Watermarking mechanism

Figure 1 shows the 3-step approach for inserting the watermark signal to one channel of the color images. DCT is first applied to the data of one channel. Then the watermark

(a) Original. (b) Red channel watermarked. (c) Green channel watermarked. (d) Blue channel watermarked.

Fig. 2: The original image of Lena (512×512) and the three variants, each of which has one watermarked channel.

(a) Original. (b) Red channel watermarked. (c) Green channel watermarked. (d) Blue channel watermarked.

Fig. 3: The original image of a landscape painting (640×640) and the three variants, each of which has one watermarked channel.

is added into the image in the frequency domain. The last step is to convert the channel back to the spatial domain.

How to perform DCT/IDCT on CPUs and GPUs has been studied in details previously [3], [7], [8]. There are several types of DCT. The most popular one is the two-dimensional symmetric variation of the transform that operates on 8×8 blocks (DCT8×8) and its inverse. The DCT8×8 is utilized in JPEG compression routines and has become a standard in image and video coding algorithms and other DSP-related areas. The two-dimensional input signal is divided into the set of nonoverlapping 8×8 blocks and each block is processed independently. This makes it possible to perform the block-wise transform in parallel, which is the key feature of the DCT8×8. For the implementation of the DCT and IDCT in this work, we used the approaches described in [3]. The CUDA SDK provides the optimal implementations of DCT8×8 and IDCT8×8 on both CPUs and GPUs.

In order to improve the speed of the watermarking process of color images, we only choose one channel to apply watermark. Once the data of the selected channel has been converted to the frequency domain, the watermark can be added. What watermark signals are used and how to insert the watermarks into the image are not the foci of this work. In this work, we do not use an additional watermark signal. Instead, a part of the coefficients in the frequency domain are altered. The original channel is divided into nonoverlapping 8×8 blocks. Each 8×8 block is separately converted to its frequency domain. We choose the 256 blocks (i.e., 16×16) on the top left corner in the frequency domain. In each 8×8 converted block, we multiple the top left coefficient with 1.01, i.e., increasing its value by 1%. Once we alter the 256 coefficients across 256 blocks, IDCT is applied to convert the channel back to the spatial domain. Then the watermarked channel can be combined with other two unaltered channels to produce the watermarked color image.

The quality of this DCT-based watermarking approach can be evaluated by the fidelity, which represents the similarity between the watermarked image and the original image. The consistency checking is performed using the objective similarity metric PSNR, which stands for Peak Signal to Noise Ratio. The PSNR between two images I and K of size M×N is defined in Equation 1.

$$PSNR(I,K) = 20\log_{10}\frac{MAX_I}{\sqrt{MSE(I,K)}} \tag{1}$$

In Equation 1, I is the original image, K is the watermarked image. MAX_I is the maximum pixel value in image I. MSE is the mean square error between I and K, as shown in

Table 1: Performance comparison between CPUs and GPUs for inserting watermark.

512×512 Lena image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	12.491 ms	12.293 ms	12.115 ms
	GPU	0.072 ms	0.072 ms	0.072 ms
	Speedup	172.9	170.0	167.4
Jetson TK1	CPU	47.171 ms	46.545 ms	45.261 ms
	GPU	2.230 ms	2.235 ms	2.189 ms
	Speedup	20.5	20.8	20.7
640×640 landscape image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	19.317 ms	19.103 ms	19.300 ms
	GPU	0.104 ms	0.107 ms	0.104 ms
	Speedup	186.3	177.9	186.3
Jetson TK1	CPU	65.854 ms	67.103 ms	66.201 ms
	GPU	2.564 ms	2.843 ms	2.886 ms
	Speedup	25.7	23.6	22.9

Equation 2.

$$MSE(I,K) = \frac{1}{M}\frac{1}{N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\|I(i,j)-K(i,j)\|^2 \quad (2)$$

Because we only watermark one channel in this work, only the data in the watermarked channel is used in the computation of PSNR.

Which channel to be used for watermarking does not matter based on our experiments. In Figures 2 and 3 we show two original images, i.e., the Lena image and one landscape image, and the respective three watermarked images. In each watermarked image, the watermarking is only applied to one channel. Naked eyes will not be able to tell the differences among the three watermarked images. Later we will show that due to the watermarking implementation used in our experiments, the PSNRs of the three watermarked images compared with the original image are almost the same.

3. Experiments and Results

We carried out the experiments on two platforms, a workstation with a high-end GPU and the Nvidia Jetson TK1 development kit. The workstation contains one Intel Core i7-3820 3.6-GHz CPU and one Nvidia Tesla K20 GPU, which contains 2,496 Kepler CUDA cores running at 706 MHz. The Jetson TK1 kit contains one Tegra K1 SOC, which features one quad-core ARM Cortex-A15 CPU running up to 2.3 GHz and 192 Kepler CUDA cores running up to 852 MHz. We ran the watermarking process outlined in Figure 1 on both the CPU and the GPU on the same platform. The optimal CPU and GPU implementations were adopted from CUDA SDK. The implementation details are described in [3].

Table 2: PSNRs of implementations on CPUs and GPUs.

512×512 Lena image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	34.189	33.146	33.037
	GPU	34.189	33.146	33.037
Jetson TK1	CPU	34.189	33.146	33.037
	GPU	34.189	33.146	33.037
640×640 landscape image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	33.513	32.827	32.884
	GPU	33.513	32.827	32.884
Jetson TK1	CPU	33.513	32.827	32.884
	GPU	33.513	32.827	32.884

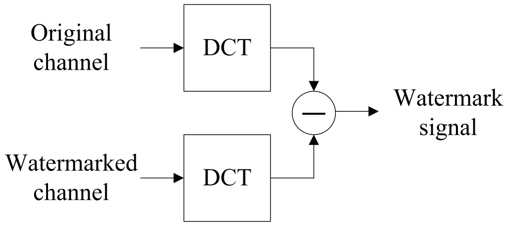


Fig. 4: Extract the watermark.

Following the watermarking approach in Section 2, it only needs to apply watermarking on one channel. Nevertheless, we implemented the watermarking on all three channels separately in order to show that there is no difference in terms of processing time and PSNR among the three channels. The performance comparison between the CPUs and the GPUs on these two systems are listed in Table 1. It can be found that the Tesla K20 GPU can consistently outperform the Intel Core i7 processor by more than 170×. On the Jetson TK1 platform, the performance speedup due to GPU implementation is above 20×. The drop of performance speedup is mainly due to the large performance gap between the Telsa K20 and the GPU part of the Tegra K1 SOC. Firstly, the number of CUDA cores drops from 2,496 (on Telsa K20) to 192 (on Tegra K1 SOC). Secondly, the Tesla K20 has higher memory bandwidth than the Tegra K1 SOC. Tesla K20 is equipped with GDDR5 memory. On the other hand, Tegra K1 SOC is equipped with DDR3 memory. Memory bandwidth is very important for DCT/IDCT application, which is a both computation intensive and memory intensive application.

The PSNRs between the original image and the watermarked image on both platform using CPUs and GPUs are listed in Table 2. It can be found that the PSNR of the DCT-based approach is consistently greater than 30. Further, there is no significant difference in terms of the PSNR when using different channels for watermarking.

We followed the approach shown in Figure 4 to implement

Table 3: Performance comparison between CPUs and GPUs for extracting watermark.

512×512 Lena image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	4.358 ms	4.395 ms	4.358 ms
	GPU	0.040 ms	0.040 ms	0.040 ms
	Speedup	108.0	108.7	107.6
Jetson TK1	CPU	18.192 ms	17.576 ms	17.512 ms
	GPU	2.107 ms	1.976 ms	1.810 ms
	Speedup	8.6	8.9	9.7
640×640 landscape image				
		Image channel		
		Red	Green	Blue
Workstation	CPU	6.915 ms	6.943 ms	6.893 ms
	GPU	0.059 ms	0.059 ms	0.059 ms
	Speedup	116.5	117.8	117.3
Jetson TK1	CPU	22.453 ms	23.454 ms	23.456 ms
	GPU	2.342 ms	2.643 ms	2.543 ms
	Speedup	9.6	8.9	9.2

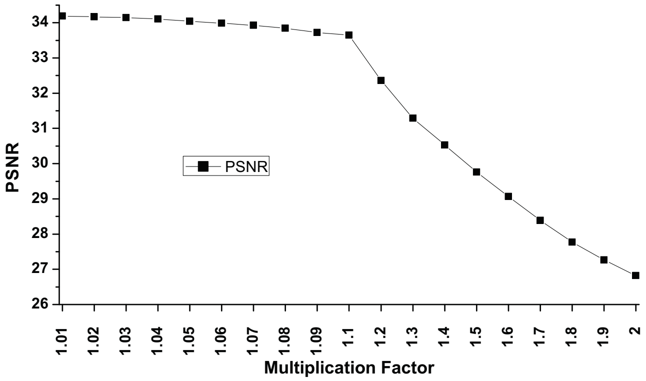


Fig. 5: The trend of PSNR as the multiplication factor grows.

the extraction of watermark for authentication. It is worth mentioning that we can take a different method to extract the watermark following our implementation of inserting watermark. Because we multiple 256 DCT coefficients by 1.01 in the insertion step, one approach to extracting the watermark is to divide the corresponding 256 DCT coefficients by 1.01. Nevertheless, the flow in Figure 4 is more general. The performance comparison between the CPUs and the GPUs is illustrated in Table 3. CPU is more efficient on performing DCT than IDCT. Therefore, the speedup from GPU implementation on the extraction of watermark is less than the speedup achieved in the insertion process.

We also checked the relationship between the PSNR and the multiplication factor, which determines how much the 256 coefficients are increased in the watermarking process. We increased the multiplication factor from 1.01 to 1.2 and checked the PSNR between the watermarked image and the original image when the watermark is applied on the red channel of the Lena image. The trend is shown in Figure 5. The trend meets our expectation that the PSNR worsens when the multiplication factor grows. It is also found that the PSNR drops slightly when the multiplication factor grows to 1.1. After that, the descending rate starts accelerating. Although we choose 256 DCT coefficients to be increased by 1% in the watermarking implementation in this work, we can not claim that these two parameters are suitable for other images of difference sizes and difference scenes. We even do not want to claim that this watermarking method is superior than other watermarking mechanisms. What we try to claim is that GPUs is capable of outperforming the CPUs for one to two orders of magnitude of performance speedup and the watermark only needs to be added to one channel of color images.

4. Conclusions

In this work, we propose to insert watermark to only one channel of color images. The watermark is added to the channel after it is converted into frequency domain using DCT. Once the watermark is added, the channel is converted back to the spatial domain and combined with the other two unaltered channels to produce the watermarked color image. In our implementation, we choose 256 DCT coefficients across 256 8×8 blocks and increase their values by 1%. The implementation results show that the powerful Tesla K20 GPU is able to outperform Intel Core i7 by more than 170 folds. On the energy-efficient Nvidia Tegra K1 SOC, the on-chip GPU can achieve more than 20× speedup than the ARM CPU on the same die.

In the future, we plan to develop more robust and difficult-to-detect watermarking mechanisms and their efficient implementations on GPUs and other accelerators.

References

[1] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 357–372, May 1998.

[2] AlpVision. Digital watermarking. Last accessed on May 25, 2016. [Online]. Available: http://http://www.alpvision.com/watermarking.html

[3] A. Obukhov and A. Kharlamov, "Discrete cosine transform for 8x8 blocks with CUDA," White paper V1.0, 2012.

[4] S. P. Mohanty, N. Pati, and E. Kougianos, "A watermarking co-processor for new generation graphics processing units," in *Proc. International Conference on Consumer Electronics (ICCE)*, Jan. 2007, pp. 1–2.

[5] C. Lin, L. Zhao, and J. Yang, "A CUDA based implementation of an image authentication algorithm," in *Proc. 2nd International Conference on Information Engineering and Computer Science (ICIECS)*, Dec. 2010, pp. 1–5.

[6] P. L. V. Vihari and M. Mishra, "Image authentication algorithm on GPU," in *Proc. 2012 International Conference on Communication Systems and Network Technologies (CSNT)*, May 2012, pp. 874–878.

[7] R. Kresch and N. Merhav, "Fast DCT domain filtering using the DCT and the DST," *IEEE Trans. Image Process.*, vol. 8, no. 6, pp. 821–833, June 1999.

[8] T. Sung, Y. Shieh, C. Yu, and H. Hsin, "High-efficiency and low-power architectures for 2-D DCT and IDCT based on CORDIC rotation," in *Proc. Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, Dec. 2006, pp. 191–196.

Multi-Level l_2- l_1-Structured Regularization Technique For Recovery of Material Abundances Maps From Hyperspectral Remote Sensing Imagery

Yuriy Shkvarko[1] **Pedro Perez**[1], **Josué López**[1], **Guillermo García**[2], **and Stewart Santos**[2]
[1]CINVESTAV – IPN, Guadalajara, Jalisco, Mexico
[2]Electronics Department, University of Guadalajara, Guadalajara, Jalisco, Mexico

Abstract - *This paper addresses a novel approach to the problem of feature enhanced recovery of material abundances maps from hyperspectral remote sensing imagery. In contrast to the competing methods that exploit separately vertex component analysis and minimum volume constrained nonnegative matrix factorization requiring knowledge of endmembers libraries, the proposed approach suggests a multilevel l_2-l_1-structured robust spatial regularization (MLl$_2$l$_1$SR) based technique for abundances maps recovery that avoids the use of endmembers libraries. Traditionally, a pure pixel search is performed to initialize the endmembers matrix estimation followed by nonnegative matrix factorization. In contrast, in the proposed approach, the spectral signatures are recovered from the original data in an adaptive fashion via a nonlinear estimation procedure with l_2-l_1-structured robust spatial regularization performed in a multi-step iterative fashion that does not require knowledge of endmember libraries, nor pure pixels. Computer simulations on synthetic images corroborate the efficiency of the new multistage MLl$_2$l$_1$SR method that is robust against model uncertainties and outperforms the competing techniques in the accuracy of recovery of abundances maps.*

Keywords: Hyperspectral unmixing, endmember extraction, abundances maps recovery, nonnegative matrix factorization, nonlinear estimation.

1 Introduction

In the last decades, hyperspectral remote sensing (RS) has made significant progress. Among the most important applications are Earth exploration, monitoring of natural resources and material identification via unmixing the composition of endmembers and abundances [1, 2]. In order to determine the elements or materials present in the RS image, the abundances maps are to be reconstructed from the hyperspectral measurements. The hyperspectral unmixing problem consists in recovering the abundances maps of each endmember from the observed data. The recent approaches for feature enhanced endmember extraction and estimation of the abundances maps employ the vertex component analysis [3], the N-Finder method [4], the minimum volume enclosing simplex technique [5], the minimum volume constrained nonnegative matrix factorization (MVC-NMF) [6], the most prominent among others [3–7]. All those techniques have several disadvantages [5–7]. The most crucial one relates to the initialization of iterative endmembers estimation procedures, namely, the assumption of presence of pure pixels in the image and poor estimations of the abundance maps in absence of precise endmembers libraries [6]. In contrast, our new proposition that we refer to as Multi-Level l_2-l_1-structured spatial regularization (MLl_2l_1SR) technique does not require knowledge of endmember libraries, nor pure pixels and thus manifests robustness against such class of model uncertainties in hyperspectral RS data unmixing.

2 Problem Statement

The majority of the endmembers mapping approaches consider a linear mixing model (LMM) of the hyperspectral images [1–9]. Let $y_{n,l}$ be a measurement of a hyperspectral camera at spectral band l and at pixel n. Following the LMM, vector $\mathbf{y}_n = [y_{n,1}, \ldots, y_{n,L}]^{\mathrm{T}} \in \mathbb{R}^L$ composed of L measurements from each spectral band is modeled as [8]

$$\{\mathbf{y}_n = \sum_{r=1}^{R} \boldsymbol{\alpha}_{r,n}\mathbf{m}_r + \boldsymbol{\upsilon}_n\}_{n=1}^{N} \tag{1}$$

for $n = 1, \ldots, N$, where $\mathbf{m}_r \in \mathbb{R}^R$ is the endmember vector that represents the spectral signature of a specific material; R is the number of endmembers or materials contained in the image; $\alpha_{r,n}$ is the contribution of the material r at the pixel n; $\boldsymbol{\alpha}_n = [\alpha_{1,n}, \ldots, \alpha_{R,n}]^{\mathrm{T}}$ is the abundance vector at pixel n; and $\boldsymbol{\upsilon}_n$ represents noise vector with statistics usually unknown to the observer. By the construction of the LMM (1), the abundance vector must always satisfy two constraints [6]: the non-negativity, $\alpha_{r,n} \geq 0$ for all r = 1, … , R and the sum-to-one restriction, $\sum_{r=1}^{R} \alpha_{r,n} = 1$ for every $n = 1, \ldots, N$.

The entire RS image can be modeled by a set of all R endmembers arranged into the endmembers matrix $\mathbf{M}$. Each column of this matrix is composed of the particular endmember signature vector $\mathbf{m}_r$; r = 1, …, R. Next, each measurement vector at all pixels can be arranged into the L×N matrix $\mathbf{Y}$. The resulting LMM is given by [6]

$$\mathbf{Y} = \mathbf{MA} + \mathbf{N} \tag{2}$$

where the columns of matrix **A** represent the corresponding abundance vectors, and **N** corresponds to the noise that encompasses also all LMM model uncertainties.

The blind hyperspectral unmixing problem consists in factorization of the image **Y** and obtaining the estimates of the endmember matrix **M** and the corresponding abundances map **A** and thus the estimates of all N abundances vectors $\boldsymbol{\alpha}_n = [\alpha_{1,n}, \ldots, \alpha_{R,n}]^{\mathrm{T}}$; $n = 1, \ldots, N$ from the noise corrupted observations (1), (2) [8].

In the next Section, for the purpose of generality, we first feature two most prominent competing techniques for the blind hyperspectral unmixing. The first one is based on the so-called pure pixels search and the related approximations of the endmembers and abundances. The second one performs the so-called MVC-NMF to factorize the endmembers matrix **M** and the recovered abundances map **A**. Here, we compare both approaches and feature their disadvantages. And finally, we construct our MLl_1l_2SR technique that next is employed to perform the feature enhanced recovery of the abundances map from the synthesized hyperspectral RS images.

3 Advanced Techniques for Recovery of Material Abundances

3.1 Pure pixels search-based recovery

An endmember r has a pure pixel if for some index ℓ_r the equality $\boldsymbol{\alpha}_{\ell_r} = \mathbf{e}_r$ holds. The vector $\mathbf{e}_r \in \mathbb{R}^{\mathrm{N}}$ is a unit vector with one at the r-th entry and all other zero entries. Assuming that the image has pure pixels and no noise, the observed vector at pixel ℓ_r is expressed as $\mathbf{y}_{\ell_r} = \mathbf{m}_r$; r = 1, …, R. In order to find pure pixels of all endmembers in a hyperspectral image the successive projections search method was developed in [8]. To identify the first endmember we have that $\hat{\mathbf{m}}_1 = \mathbf{y}_{\hat{\ell}_1}$, where $\hat{\ell}_1 = \arg\max_{n=1,\ldots,N} \|\mathbf{y}_n\|_2^2$. When $k-1$ previous endmembers have been identified, the orthogonal complement projector $P^{\perp}_{\mathbf{M}_{1:k-1}}$ of $\mathbf{M}_{1:k-1} = [\mathbf{m}_1, \ldots, \mathbf{m}_{k-1}]$ is used to perform nulling [8] and to identify the next endmember with $\hat{\mathbf{m}} = \mathbf{y}_{\hat{\ell}_k}$ where $\hat{\ell}_k = \arg\max_{n=1,\ldots,N} \|P^{\perp}_{\hat{\mathbf{M}}_{1:k-1}} \mathbf{y}_n\|_2^2$. Clear that in the case of the noise degraded data (1), (2), this approach fails to accurately recover all pure pixels.

3.2 Minimum volume constrained nonnegative matrix factorization

Next, to extract the endmember signatures one may employ the prominent MVC-NMF technique proposed in [6]. It suggests to perform the nonnegative matrix factorization for extracting the endmembers and abundances matrices via restricting the volume of a simplex formed by the observed data. Thus, the MVC-NMF method assumes solution of the problem

$$\text{minimize}\, f(\mathbf{M},\mathbf{A}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{MA}\|_F^2 + \lambda J(\mathbf{M}) \;\; \text{w.r.t.} \;\; \mathbf{M},\mathbf{A} \quad (3)$$
$$\text{subject to } \mathbf{M} \geq \mathbf{0} \text{ and } \mathbf{A} \geq \mathbf{0};\;\; \mathbf{1}_R^T\mathbf{A} = \mathbf{1}_N^T$$

where $\nabla_{\mathbf{M}}$ and $\nabla_{\mathbf{A}}$ represent the corresponding numerical gradient operators.

The crucial problem relates to satisfying the sum-to-one constraint on the recovered abundances. In [11], this problem is treated via replacing in (3) **Y** and $\hat{\mathbf{M}}$ by the corresponding augmented matrices $\bar{\mathbf{Y}} = \left[\mathbf{Y}^T \;\; \delta\mathbf{1}_N\right]^T$ and $\hat{\bar{\mathbf{M}}} = \left[\hat{\mathbf{M}}^T \;\; \delta\mathbf{1}_N\right]^T$ where δ is a positive augmentation parameter, usually from the interval (10, 20) [11], i.e., δ = 10…20. The fix point iterations for resolving (3) are to be terminated after the gradient of the Euclidean norm of the objective function in (3) becomes less than some user specified threshold ε. In our simulations we adopted the proposition from [11] and adjusted ε = 0.05. As in the pure pixels-based search technique, in the case of the noise degraded data (1), (2), the factorization procedure (3) is extremely sensitive to initializations of **M** and **A**. At improper initializations, the procedure usually produces incorrect results [9, 10], hence, the method fails to operate as well.

3.3 Nonlinear robust recovery using MLl_1-l_2SR method

Our alternative approach for obtaining the robust enhanced estimates of the abundances map consists in incorporation into the modified nonlinear estimation algorithm the sparsity promoting l_1-norm-type spatial regularizer proposed originally in [12]. The modified unmixing problem is cast as minimization of the generalized cost function

$$\min_{\mathbf{A}} \rightarrow J(\mathbf{A}) = J_{\mathrm{err}}(\mathbf{A}) + \eta J_{\mathrm{sp}}(\mathbf{A}) \quad (4)$$
$$\text{subject to } \mathbf{A} \geq \mathbf{0};\;\; \mathbf{A}^T\mathbf{1}_R = \mathbf{1}_N$$

Here, J_{err} represents the error of the model specified next by (6), J_{sp} is the regularization term aimed at promoting the abundances similarity in the neighboring pixels specified next by (7), and the regularization parameter η controls the tradeoff between two error measures in (4) [12]. The general hyperspectral image model (1) is now replaced by its nonlinear counterpart, $y_{n,\ell} = \psi_{\boldsymbol{\alpha}_n}(\mathbf{m}_{\lambda_\ell}) + \upsilon_{n,\ell}$, in which an unknown nonlinear function $\psi_{\boldsymbol{\alpha}_n}$ specifies now the interaction between the endmember spectra. Let $\psi_{\boldsymbol{\alpha}_n}(\mathbf{m}_{\lambda_\ell}) = \boldsymbol{\alpha}_n^T\mathbf{m}_{\lambda_\ell} + \psi_n(\mathbf{m}_{\lambda_\ell})$, where ψ_n is a properly specified Gaussian kernel function [12]. Then, the unmixing problem becomes

$$\hat{\boldsymbol{\alpha}}_n, \hat{\boldsymbol{\psi}}_n = \arg\min_{\boldsymbol{\alpha}_n, \boldsymbol{\psi}_n} \frac{1}{2}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\boldsymbol{\psi}_n\|^2 + \frac{1}{\mu}\|\boldsymbol{\upsilon}_n\|^2\right) \quad (5)$$
$$\text{subject to } \boldsymbol{\alpha}_n \geq \mathbf{0},\;\; \mathbf{1}^T\boldsymbol{\alpha}_n = 1$$

where $\boldsymbol{\upsilon}_{n,\ell}$ represents an (L × 1) disadjustment error vector at pixel n [12] with the l-th element $\upsilon_{n,\ell} = y_{n,\ell} - (\boldsymbol{\alpha}_n^T \mathbf{m}_{\lambda_\ell} + \psi_n(\mathbf{m}_{\lambda_\ell}))$ .With these specifications, the modeling error term in (4) is expressed as

$$J_{\text{err}}(\mathbf{A},\boldsymbol{\psi}) = \frac{1}{2}\sum_{n=1}^{N}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\boldsymbol{\psi}_n\|^2 + \frac{1}{\mu}\|\boldsymbol{\upsilon}_n\|^2\right) \qquad (6)$$

$$\text{subject to } \mathbf{A} \geq \mathbf{0};\ \mathbf{A}^T\mathbf{1}_R = \mathbf{1}_N$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_N]$ and $\boldsymbol{\psi} = \{\psi_n; n = 1, ..., N\}$ represents a set of Gaussian modeling kernels [12, 13]. Next, we define the second term in the objective function in (4) as an l_1-type regularizer

$$J_{\text{sp}}(\mathbf{A}) = \sum_{n=1}^{N}\sum_{m\in\mathcal{N}(n)}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_m\|_1 \qquad (7)$$

aimed at promoting the spatial correlation between the neighboring pixels. Here, $\|\cdot\|_1$ denotes the vector l_1-norm and $\mathcal{N}(n)$ is the set of neighbors of pixel n. The neighborhood of pixel n is defined via four closest pixels $n - 1$ and $n + 1$ for row proximity and n - w and n + w for column proximity, where n - w and n + w correspond to the neighboring pixels in the rows below and over the pixel n. Let $\mathbf{H}_{\leftarrow}$, $\mathbf{H}_{\rightarrow}$, $\mathbf{H}_{\uparrow}$ and $\mathbf{H}_{\downarrow}$ be the (N×N) linear operators which calculate the differences between the abundance vector and its left-, right-, top- and down-side neighbors, respectively. With this notation, the regularizer (7) can be rewritten as

$$J_{\text{sp}}(\mathbf{A}) = \|\mathbf{AH}\|_{1,1} \qquad (8)$$

with $\mathbf{H}$ as an (N×4N) matrix defined by the composition $(\mathbf{H}_{\leftarrow}\ \mathbf{H}_{\rightarrow}\ \mathbf{H}_{\uparrow}\ \mathbf{H}_{\downarrow})$ and $\|\cdot\|_{1,1}$ as a sum of the l_1-norms of the columns of a matrix. Therefore, the problem (4) transforms into

$$\hat{\mathbf{A}}, \hat{\boldsymbol{\psi}} = \arg\min_{\mathbf{A},\boldsymbol{\psi}}\sum_{n=1}^{N}\frac{1}{2}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\boldsymbol{\psi}_n\|^2 + \frac{1}{\mu}\|\boldsymbol{\upsilon}_n\|^2\right) + \eta\|\mathbf{AH}\|_{1,1} \qquad (9)$$

$$\text{subject to } \mathbf{A} \geq \mathbf{0};\quad \mathbf{A}^T\mathbf{1}_R = \mathbf{1}_N .$$

To decouple in (9) the non-smooth l_1-norm regularization term from the constrained least squares support vector regression (LS-SVR) term, following [12], we suggest to introduce two surrogate matrices $\mathbf{U}$ and $\mathbf{V}$ to make the overall LS-SVR problem (9) well tractable by relaxing connections between pixels. Next, applying the split-Bregman algorithm [13] to (9), we replace it by the following algebraically equivalent problem

$$\hat{\mathbf{A}}^{(k+1)}, \hat{\boldsymbol{\psi}}^{(k+1)}, \hat{\mathbf{V}}^{(k+1)}, \hat{\mathbf{U}}^{(k+1)} = \arg\min_{\mathbf{A}\in S_{\mathbf{A}},\boldsymbol{\psi},\mathbf{V},\mathbf{U}}\sum_{n=1}^{N}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\boldsymbol{\psi}_n\|^2 + \frac{1}{\mu}\|\boldsymbol{\upsilon}_n\|^2\right)$$

$$+\eta\left\|\hat{\mathbf{U}}^{(k)}\right\|_{1,1} + \frac{1}{2}\left\|\hat{\mathbf{A}}^{(k)} - \hat{\mathbf{V}}^{(k)} - \mathbf{D}_1^{(k)}\right\|_F^2 + \frac{1}{2}\left\|\hat{\mathbf{U}}^{(k)} - \hat{\mathbf{V}}^{(k)}\mathbf{H} - \mathbf{D}_2^{(k)}\right\|_F^2 \qquad (10)$$

where the so-called Bregman decomposition matrices [13] are represented by $\mathbf{D}_1^{(k+1)} = \mathbf{D}_1^{(k)} + (\hat{\mathbf{V}}^{(k+1)} - \hat{\mathbf{A}}^{(k)})$ and $\mathbf{D}_2^{(k+1)} = \mathbf{D}_2^{(k)} + (\hat{\mathbf{V}}^{(k+1)}\mathbf{H} - \hat{\mathbf{U}}^{(k+1)})$, respectively, and $\|\cdot\|_F^2$ defines the Frobenius norm of the corresponding matrix. Thus, the problem at hand (10) could be treated as a multi-stage optimization problem. Here beneath, we construct the three-step iterative procedure that provides a solution to that problem (10).

Step 1. The optimization in (10) with respect to ($\mathbf{A}$, ψ) is performed via decomposing (10) into sub-problems with respect to each one of the abundances vectors $\boldsymbol{\alpha}_n$. This yields the estimate of the complete abundances map $\hat{\mathbf{A}}$ in a fix point iterative form similar to (3).

Step 2. The optimization problem with respect to $\mathbf{V}$ is reduced to the following iterative scheme

$$\hat{\mathbf{V}}^{(k+1)} = (\hat{\mathbf{A}}^{(k+1)} - \mathbf{D}_1^{(k)} + (\hat{\mathbf{U}}^{(k)} - \mathbf{D}_2^{(k)})\mathbf{H}^T)(\mathbf{I} + \mathbf{HH}^T)^{-1} . \qquad (11)$$

Step 3. The solution of the optimization problem with respect to $\mathbf{U}$ can be next transformed into a fix point iterative thresholding-type procedure

$$\hat{\mathbf{U}}^{(k+1)} = \text{Thresh}(\hat{\mathbf{V}}^{(k+1)}\mathbf{H} + \mathbf{D}_2^{(k)}, \eta) \qquad (12)$$

where $\text{Tresh}(\text{x},\tau) = \text{sign}(x)\max(|x| - \tau, 0)$ is the element-wise threshold function applied for each element of matrix $\hat{\mathbf{U}}^{(k)}$ at all performed iterations.

Last, all iterations for computing $\hat{\mathbf{A}}$, $\hat{\mathbf{V}}$ and $\hat{\mathbf{U}}$ are terminated using the stopping rule similar to that explained in the previous Section. To conclude, the composite unmixing problem is presented in a form of the pseudocode summarized in Table 1.

4 Simulations and Discussions

The test was performed on a 75 × 75 pixel framed RS image with five endmembers selected from the USGS digital spectral library [14]. Each pixel has different proportions of abundances. The image was next degraded with zero-mean white Gaussian noise with different signal-to-noise ratio (SNR) values. Five different columns of Fig.1 relate to five abundances maps corresponding to five different tested endmembers. The first row (a) in Fig. 1 presents the noise-free true abundances maps generated for different simulated endmembers. The second row (b) presents the computational results of recovery of the abundances maps from the noise degraded data performed applying the competing pure pixel search based technique [6] combined with the NMF method of [10] for a moderate SNR = 10 dB. The third row (c) shows the same recovery results obtained using the second competing MVC-NMF technique of [11] for the same SNR = 10 dB. The forth row (d) in Fig.1 represents the abundances maps recovered with the proposed here MLl_2l_1SR method for the same SNR = 10 dB.

Table 1. Nonlinear recovery of abundances map with MLl_2l_1SR algorithm

```
Initialization
    Set the regularization parameter $\eta = 0.5$.
    Initialize the zero-step iterations $\hat{\mathbf{A}}^{(0)} = \mathbf{0}$, $\hat{\mathbf{V}}^{(0)} = \mathbf{0}$ and $\hat{\mathbf{U}}^{(0)} = \mathbf{0}$, zero matrices.
Repeat
    Compute current alternating iterations of abundances $\hat{\mathbf{A}}^{(k)}$ using (9).
    Update iteratively matrices $\hat{\mathbf{V}}^{(k)}$ and $\hat{\mathbf{U}}^{(k)}$ using alternating iterations (11) and (12).
    Return to (9) and update $\hat{\mathbf{A}}^{(k+1)}$ using currently updated $\hat{\mathbf{V}}^{(k)}$ and $\hat{\mathbf{U}}^{(k)}$.
until   Stop conditions are satisfied at the specified termination threshold $\varepsilon = 0.05$.
Result Use the last iteration $\hat{\mathbf{A}}^{(K)}$ as the recovered abundances matrix $\hat{\mathbf{A}}$.
```

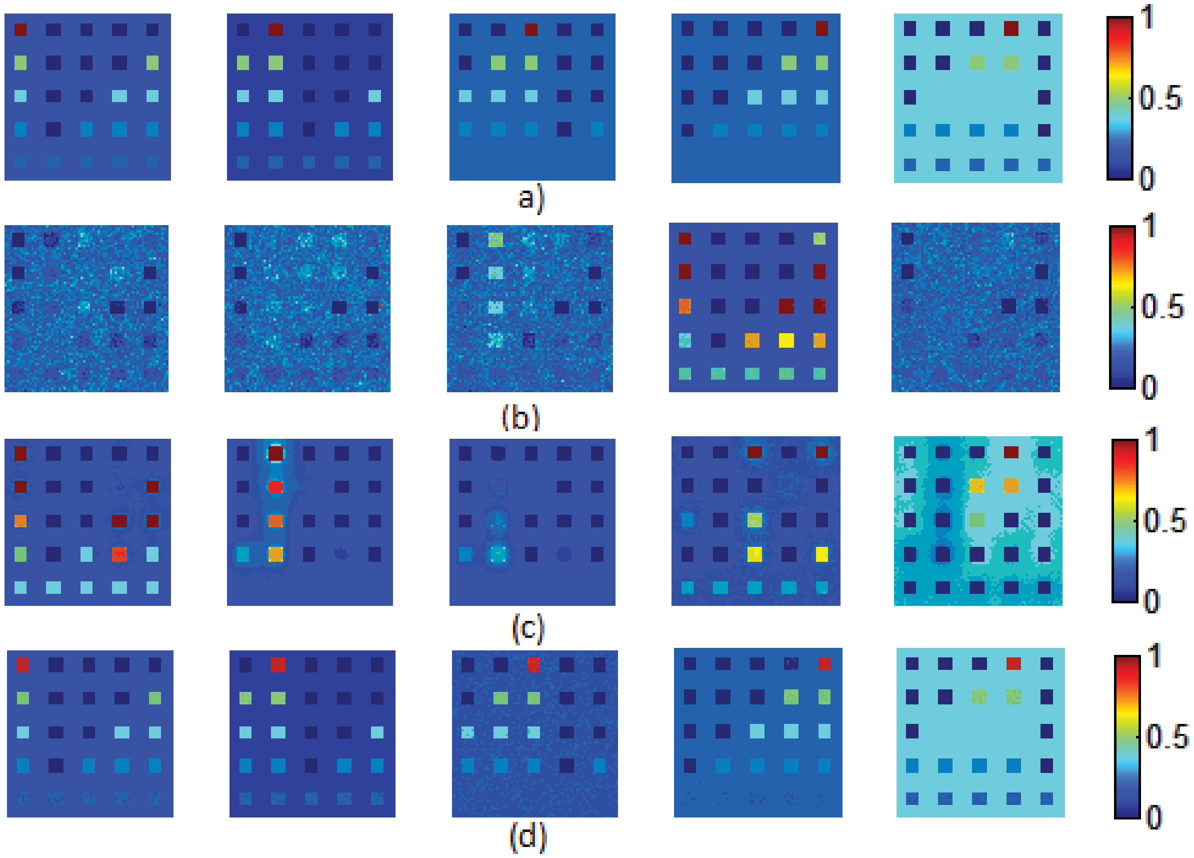


Figure 1: In the horizontal direction, the presented images expose abundances maps corresponding to five tested endmembers. **(a)** True abundances maps, **(b)** Abundances maps recovered applying the pure pixel search-NMF algorithm [6, 10]; **(c)** Abundances maps recovered applying the MVC-NMF algorithm [11]; **(d)** Abundances recovered with the proposed MLl_1l_2SR algorithm. All recovery results are presented for the same SNR = 10 dB.

From the presented simulations it is evident that the competing methods of [6, 10, 11] produce the results that manifest low quality of recovery the abundances maps for the scenes composed of different endmembers. This means that due to imprecise pure pixels based initializations in some scenarios with the noise corrupted data, those methods fail to perform accurate recovery. In contrast, the MLl_2l_1SR technique exhibits robust and much more accurate recovery of the abundances maps in all tested scenarios.

5 References

[1] J.M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N.M. Nasrabadi, and J. Chanussot. Hyperspectral remote sensing data analysis and future challenges. Geoscience and Remote Sensing Magazine, IEEE, 1(2):6–36, June 2013.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE journal of selected topics in applied earth observations and remote sensing, 2012.

[3] J.M.P. Nascimento and J.M. Bioucas Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. Geoscience and Remote Sensing, IEEE Transactions on, 43(4):898–910, April 2005.

[4] Michael E. Winter. N-finder: an algorithm for fast autonomous spectral end-member determination in hyperspectral data, 1999.

[5] Tsung-Han Chan, Chong-Yung Chi, Yu-Min Huang, and Wing-Kin Ma. A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. Signal Processing, IEEE Transactions on, 57(11):4418–4432, Nov 2009.

[6] Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. Geoscience and Remote Sensing, IEEE Transactions on, 45(3):765–777, March 2007.

[7] M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, and J.C. Tilton. Advances in spectral-spatial classification of hyperspectral images. Proceedings of the IEEE, 101(3):652–675, March 2013.

[8] W.-K. Ma, J.M. Bioucas-Dias, Tsung-Han Chan, N. Gillis, P. Gader, A.J. Plaza, A. Ambikapathi, and Chong-Yung Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. Signal Processing Magazine, IEEE, 31(1):67–81, Jan 2014.

[9] Chein-I Chang and Qian Du. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. Geoscience and Remote Sensing, IEEE Transactions on, 42(3):608–619, March 2004.

[10] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. Neural Comput., 19(10):2756–2779, October 2007.

[11] D.C. Heinz and Chein-I Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. Geoscience and Remote Sensing, IEEE Transactions on, 39(3):529–545, Mar 2001.

[12] Jie Chen, C. Richard, and P. Honeine. Nonlinear estimation of material abundances in hyperspectral images with l_1-norm spatial regularization. Geoscience and Remote Sensing, IEEE Transactions on, 52(5):2654–2665, May 2014.

[13] Tom Goldstein and Stanley Osher. The split bregman method for l_1-regularized problems. SIAM Journal on Imaging Sciences, 2(2):323–343, 2009.

[14] R. Clark, G. Swayze, R. Wise, E. Livo, T. Hoefen, R. Kokaly, and S. Sutley. (2007). USGS digital spectral library splib06a: U.S. Geological Survey, Digital Data Series 231. [Online]. Available: http://speclab.cr.usgs.gov/spectral.lib06.

An Autoencoder-Based Image Descriptor for Image Matching

Chenyang Zhao[1], **A. Ardeshir Goshtasby**[1], **Shaodan Zhai**[1]
[1]Computer Science and Engineering, Wright State University, Dayton, OH, USA

Abstract— *Local image features are needed in various computer vision applications. For this purpose, a large number of point detectors and descriptors have been developed during recent years. Nonetheless, creation of effective descriptors is still a topic of research. The Scale Invariant Feature Transform (SIFT) proposed by David Lowe is a widely used image descriptor in image analysis and image matching. SIFT is found to provide a high matching rate and is robust under various image transformations; however, it is relatively slow in image matching. Autoencoder is an effective computational method for representation learning. In this paper, autoencoder is used to construct a low-dimensional representation for a high-dimensional data while preserving the structural information within the data. In many computer vision applications, a high dimensional data implies a high computational cost. The main motivation in this work is to significantly improve the speed of image descriptors without reducing their match ratings noticeably. A new descriptor is designed that is based on the autoencoder concept. The proposed descriptor can reduce the size and complexity of a descriptor significantly, considerably reducing the time required to find an object of interest in an image.*

Keywords: Image matching, keypoints, Image descriptors, Autoencoder

1. Introduction

Local image features are needed in various computer vision applications, such as image matching [1], object recognition [2] and image retrieval [3]. Many point detectors and descriptors have been proposed throughout the years [4], but creating effective descriptors is still a topic of research. The main focus of this work is to create image descriptors that are invariant to rotation, translation, scaling, change in view, and change in illumination.

Image matching is a major area of research in computer vision and image analysis. It involves finding two similar images or image patches using various features. Image features are of two types: global features and local features. Global features are such as statistical properties of an image like standard deviation and variance of intensities. Local features can overcome the limitations of global features [5] by distinguishing patterns that locally differ [6]. Local features require that keypoints in an image be known. They then provide properties of region centered at the keypoints.

Many methods have been proposed for detecting and describing local image features. Mikolajczyk and Schimid evaluate the performances of several feature detectors, descriptors, and matching including steerable filter [7], moment invariants [8], complex filters [9], scale invariant feature transform [2], and cross-correlation [10]. Based on the reported experimental result, the scale invariant feature transform (SIFT) of Lowe (1999) has been found to provide consistently high performance measures while remaining robust under various geometric and radiometric transformations.

SIFT is invariant to translation, rotation, and scale of an image, and for that reason it is widely used in object recognition, image matching, image classification, and image retrieval. SIFT can be used to describe and match images of a scene taken from slightly different views. Due to the high popularity of SIFT, it is of no surprise that several variants and extensions of it have been developed. For example, Ke and Sukthankar described the PCA-SIFT that applies Principal Components Analysis (PCA) to the normalized SIFT descriptor [11]. The Gradient location and orientation histogram (GLOH) [10] changes SIFT quantization grid and uses PCA to reduce the size of the generated description.

Autoencoder [13] can be used in learning and constructing a low-dimensional representation of high-dimensional data. This can be done while preserving the structure and geometry of the data. In many computer vision tasks, the high dimensionality of data implies a high computational cost. Autoencoders are a representation learning method that can learn an effective representation of the data, and transform the data into an easier form or a lower dimension.

As image databases grow in size, modern solutions to local feature-based image indexing and matching must not only be accurate but also highly efficient to remain viable [18]. This paper focuses on the design of a new image feature descriptor using the autoencoder concept and evaluates its application in image matching. The new descriptor can reduce the size and complexity of an existing feature descriptor to improve its performance in image matching.

2. Overveview of SIFT Algorithm

The most widely used image descriptor is the SIFT descriptor [2]. It extracts distinct and invariant features from an image for object recognition. SIFT generates descriptions for local neighborhoods in four steps:

- **Scale-space extrema detection:** First, using a Difference of Gaussian (DOG) operator, it detects keypoints in an image. Given an image $I(x, y)$, a smoothed version of it is obtained by convolution it with a Gaussian filter:

$$J(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$G(x, y, \sigma)$ is a 2-D Gaussian of standard deviation σ:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

The Laplacian of Gaussian is then approximated by a difference of two Gaussians:

$$D(x, y, \sigma) = J(x, y, k\sigma) - J(x, y, \sigma)$$

$k = 1.61$ is a constant and represents the scale ratio of the Gaussians in scale-space. In SIFT, $k = 1.4$ rather than 1.61 is used. In order to detect extrema points from a stack of DOG images, a point in the scale-space that is locally maximum or minimum within a $3 \times 3 \times 3$ in scale-space is taken to represent a keypoint.

- **Keypoint localization:** The location and scale of each candidate point is determined and keypoints are selected based on a measure of stability. Not all detected local extrema are keypoints. Points that have low contrast or fall along an edge are discarded. A keypoint is determined by fitting a 3-D quadratic polynomial using a second order Taylor expansion with the origin at the keypoint.
- **Orientation assignment:** A dominant orientation is assigned to each keypoint based on local image gradient directions. After the location of the keypoint is determined, a dominant orientation is assigned to each keypoint based on local gradient directions. For each pixel of the region around the detected location the gradient magnitude $m(x, y)$ and direction $\theta(x, y)$ are computed using pixel differences :

$$\begin{aligned} m(x, y) \quad = \quad & ((J(x+1, y) - J(x-1, y))^2 \\ & +(J(x, y+1) - J(x, y-1))^2)^{\frac{1}{2}} \end{aligned}$$

- **Keypoint description:** A descriptor is generated for each keypoint using local image gradients at the obtained scale. The window centered at a keypoint is subdivided into a grid of 4× subwindows and gradient directions within each subwindow are quantized into 8 orientations, creating a histogram with 8 bins for each subwindow. Overall, a feature vector with 128 values is generated, representing the descriptor for the keypoint. This descriptor is rotation invariant and is made invariant of illumination for being scaled to have a unit length.

3. Autoencoder based SIFT descriptor

The success of the SIFT descriptor has been the motivation behind this work. To handle the high computational cost of SIFT, we introduce a new keypoint descriptor based on autoencoder, called AED, which has competitive matching performance, while suitable for real-time applications.

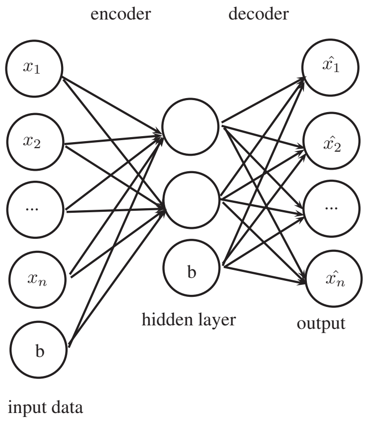


Fig. 1: The structure of a simple 3-layer autoencoder

3.1 Autoencoder

An autoencoder is a specific form of an artificial neural network [15]. The purpose of an autoencoder is to learn another representation of the input data, in compressed or sparse representation. More specifically, an autoencoder is an unsupervised learning method that sets the target values to the input values, i.e., $y_i = x^{(i)}$. Generally, an autoencoder contains one input layer, one or more hidden layers, and one output layer, which has exactly the same number of entries as the input layer, as shown in Fig. 1. Different layers of the network apply a series of transformations (non-linear in most cases) to the input data, and the hidden layers are the different representations of the input data.

Functionally, an autoencoder contains two components in the training process, an *encoder* and a *decoder*. The encoder is used to encode the input data to the desired compressed (or sparse) representation by applying transformations h_j (the jth layer), while the decoder decodes this compressed representation to an approximation of the inputs $\hat{x}^{(i)}$, with $\hat{x}^{(i)}$ as close to $x^{(i)}$ as possible. Usually, the (non-linear) transformation h is a sigmoid function, i.e., the logistic function $h(z) = \frac{1}{1+\exp(-z)}$, where $z = Wx + b$ and W is a weight matrix, b is a bias vector. In the training phase of an autoencoder, the parameters W and b are optimized such that the average reconstruction error is minimized. The reconstruction error is used to measure the similarity of $\hat{x}^{(i)}$ and $x^{(i)}$, which can be measured in many ways. In this study, we simply use the traditional squared error, that is, for any input $x^{(i)}$

$$L(x^{(i)}) = \sum_{k=1}^{d} (x_k^{(i)} - \hat{x_i})^2$$

where we assume $x^{(i)}$ is a d-dimensional vector.

Usually the backpropagation algorithm is applied to train an autoencoder by propagating the error information (gra-

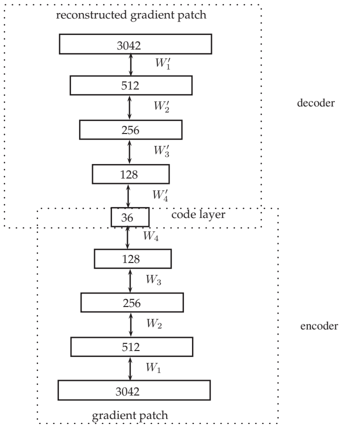


Fig. 2: The structure of the proposed autoencoder.

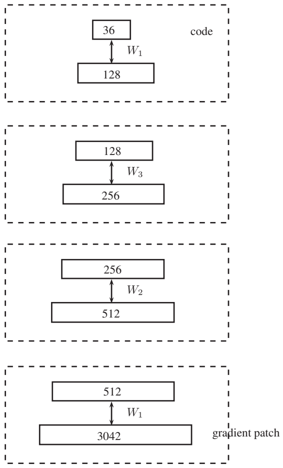


Fig. 3: The pretraining process.

dients) from the output layer back to the input layer [16]. However, for a deep neural network structure, the gradients are smaller, less pronounced since the diffusion of gradients problem. To alleviate this problem, Hinton et. al. [15] proposed a stacking multiple encoders (and their corresponding decoders) when building a deep autoencoder.

3.2 Autoencoder training

An autoencoder enables us to project a high-dimensional data into a compressed low-dimensional representation. For the applications in this research, an autoencoder will be trained offline and stored. The offline training process can be summarized by the following steps:

1) Choose images for offline autoencoder training.
2) Use SIFT detector to extract keypoints in these images.
3) Extract the local gradient patch centered at each keypoint, and generate the gradient vector for each keypoint.
4) Train the autoencoder with the gradient patch as the input.

To prepare data for autoencoder training, first over 40000 41×41 patches from diverse images are collected. These images or patches are not used later in the evaluation. Then, the horizontal and vertical gradients at patches are calculated and stored in a $2 \times 39 \times 39 = 3042$ vector. Finally, this vector is normalized so each feature value falls in the range $[0, 1]$.

The stacked denoising autoencoder is chosen to train the projection model as the compressed representation that learns by a denoising autoencoder, which is robust to small irrelevant changes in input, a very important property of the proposed autoencoder. The autoencoder consisted of an encoder with layers of 3042-512-256-128-36, as shown in Figure 2. The 36-dimensional vector is the code layer of the encoder part of the deep autoencoder.

In the code layer, the 36 real-valued numbers are the encoded version of the 3042 gradient patch. The decoder part (which is symmetric to the encoder) of a deep autoencoder learns to decode the compressed vector, which becomes the reconstructed gradient patch as it makes its way back. Figure 2 illustrates this.

The pre-training process is shown in Figure 3. We first train the bottom-most autoencoder, where the corresponding encoder with layers of 3042-512, and the decoder with layers of 512-3042. The goal is to optimize the weight matrix W_1 such that the reconstruction error, which is measured by

$$L(x^{(i)}) = \sum_{k=1}^{d} (x_k^{(i)} - \hat{x}_k^{(i)})^2$$

is minimized. In this step, the gradient patch is encoded to a 512-dimensional vector, and the transpose of W_1 is used to decode this 512-dimensional vector to the reconstructed gradient patch with 3042 values. After training of this bottom-most autoencoder, a new autoencoder is constructed by taking the 512-dimensional vector as input, with layers of 512-256 in its encoder part and 256-512 in its decoder part. The goal here is to try to reconstruct this 512-dimensional vector such that the reconstruction error is as low as possible. The third and the fourth autoencoders are trained in the same way, and finally the weights W_1, W_2, W_3, and W_4 are obtained. These weights are used to initialize the autoencoder in Figure 2.

After the pretraining process, a global fine-tuning process is applied to fine-tune the weights such that the overall reconstruction error is minimized, as shown in Figure 4. From a high level perspective, the fine tuning process treats all layers of the autoencoder as a single model so that all the weights in the autoencoder are tuned in each iteration. As

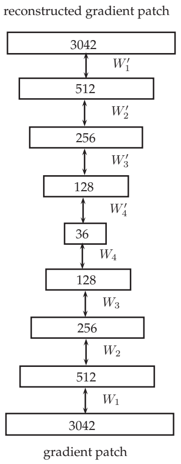


Fig. 4: The fine tuning process.

pretraining, the reconstruction error is measured by squared error too.

The mini-batch stochastic gradient descent (SGD) algorithm is used to perform the optimization, where the batch size is 100. During the pretraining and fine-tuning, the weights are updated after each mini-batch iteration. For pretraining, each hidden layer is trained for 500 epochs through the entire input, weights are initialized with small random numbers subject to Gaussian distribution with 0 mean and 0.1 standard deviation, and the learning rate is set to be 0.05. For fine-tuning, the hidden layers are trained for 700 epochs through the entire training gradient patches, weights are initialized with the resulting weights of pre-training, and the learning is set to be 0.03. For both pretraining and fine-tuning, the logistic function is used as the sigmoid transformation function, and 30% of input are masked as 0 to perform the denoising autoencoder.

3.3 Keypoint description and matching

After offline training of the autoencoder, the encoder is used to project a gradient patch to a compressed representation, which consists of 36 real values. This 36-dimensional vector can be used with the same matching algorithm as SIFT, but note that this compressed vector is significantly smaller than the feature vector of SIFT, which has 128 dimensions. Thus, the proposed AED autoencoder speeds up feature matching by a factor of 3 compared to the original SIFT method.

More specifically, the first step in AED is to detect the keypoints. For that we use the difference-of-Gaussian detector, which is similar to SIFT. For SIFT descriptor, orientation histogram is utilized as the feature vector to represent features at a keypoint. AED uses autoencode of the histogram to represent the gradient patch. For each detected keypoint, the feature extraction window is used to extract the gradient patch centered at the keypoint. Then the patch is rotated so its main orientation aligns with the x-axis, and scaled according to the keypoint scale. For an image patch with size 39×39, the feature vector has 3042 elements $(39 \times 39 \times 2)$. A 3042-512-512-256-256-128-36 autoencoder is trained to learn feature vectors from the image. The trained autoencoder, as described in the last section, is tested on new images. By reducing the dimension of the feature vector to 36, a significantly smaller feature vector compared to standard SIFT feature with 128-element vectors is obtained.

After generating AED feature vectors in an image, nearest-neighbor with distance ratio (NNDR) and Euclidean distance between normalized feature descriptors is used in matching, with the requirement that the nearest neighbor distance must be different by a certain percentage of the second nearest neighbor distance to consider a match unique.

4. Experiments

In this section, we present experimental results comparing the performances of SIFT and AED in different conditions: change in scale and rotation, change in image blur, change in illumination, and change under affine transformation. In order to evaluate the performance of AED, we also add PCA-SIFT to the comparison [14]. PCA-SIFT is a variant of SIFT which tries to handle the high computational cost by mapping the gradient vector to a new smaller vector by the Principal Components Analysis (PCA).

4.1 Data set

The standard Mikolajczyx database[1] was used to evaluate the feature under different transformations, such as image blur, viewpoint change, and degradations caused by jpeg compression.

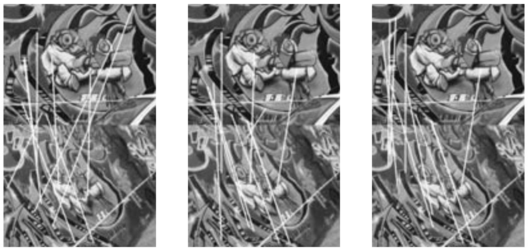
4.2 Evaluation criteria

In recent years, precision and recall have become popular evaluation metrics for image matching [17]. In the AED based image matching, the performance is evaluated in terms of precision and recall of the matching method. Precision and recall are based on the number of correct and false matches between two images. The standard definitions of these two measures are:

$$1 - Precision = \frac{Number\ of\ false - positives}{Total\ number\ of\ matches}$$

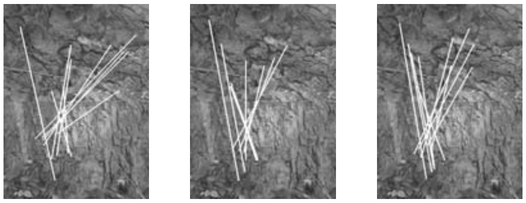
$$Recall = \frac{Number\ of\ true - positives}{Total\ number\ of\ positives}$$

[1] *The dataset available at http: //www. robots.ox.ac.uk./~vgg/research/affine/



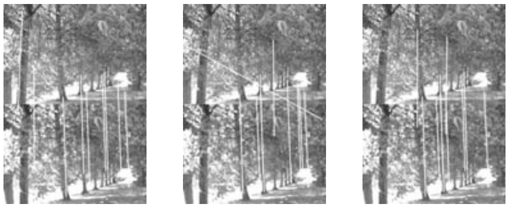
SIFT(4/10) PCA-SIFT(8/10) AED(10/10)

Fig. 5: The first 10 matching key points between images under affine invariance



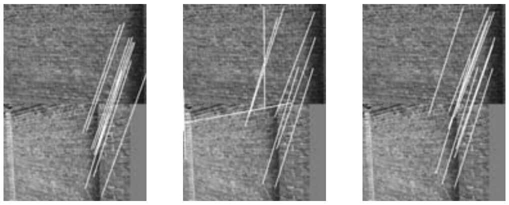
SIFT(9/10) PCA-SIFT(10/10) AED(10/10)

Fig. 6: The first 10 matching key points between images Scale and Rotation invariance



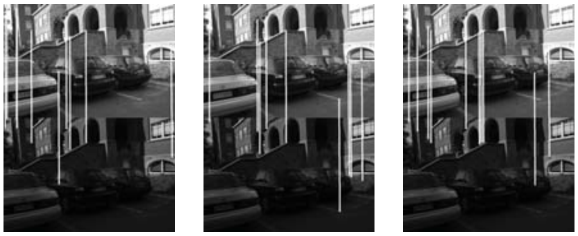
SIFT(10/10) PCA-SIFT(9/10) AED(10/10)

Fig. 7: The first 10 matching key points between images under blur



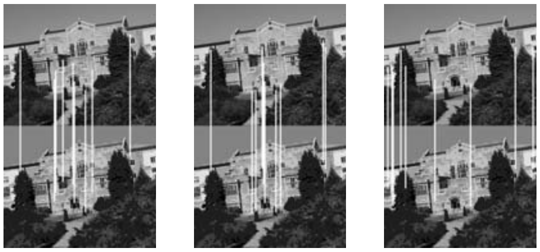
SIFT(10/10) PCA-SIFT(8/10) AED(10/10)

Fig. 8: The first 10 matching key points between images under viewpoint change



SIFT(10/10) PCA-SIFT(10/10) AED(10/10)

Fig. 9: The first 10 matching key points between images under lighting change



SIFT(10/10) PCA-SIFT(10/10) AED(10/10)

Fig. 10: The first 10 matching key points between images under compression

4.3 Performance Evaluation under Different Situations

In our experiments, test images are taken from the Graffiti dataset used to evaluate the descriptors. Six different changes in imaging conditions are evaluated: affine transformation; scale changes and rotation; image blur; illumination changes; viewpoint change; and image compression. In the viewpoint change test, the camera varies from a front-parallel view to one with foreshortening at about 20 degrees. The scale change and blur images are acquired by varying the camera zoom and focus, respectively. The scale changes are by about a factor of up to four. The lighting changes are introduced by varying the camera aperture[10]. In order to make a fair comparison, we determined proper values for the dimensionality of the feature space n for each algorithm. We set n to 128 and 36 for standard SIFT and PCA-SIFT respectively. For our proposed AED algorithm, same as in PCA-SIFT, we set n to 36. Low dimensionality means PCA-SIFT and AED have significantly reduced computational costs compared to that of SIFT. In order to show the matching results, we require each algorithm to return $m = 10$ matching pairs. The effect of feature point matching compared with SIFT, PCA-SIFT and AED is shown in Figure Fig. 5-10.

Affine invariance The first set of experiments is conducted on the set of graffiti images for investigating the performance of SIFT, PCA-SIFT and AED under affine invariance, where each image has a different viewpoint, the experimental results are shown in Fig. 5.

Fig. 5 shows the result of the first 10 matching keypoints between the images for SIFT, PCA-SIFT and AED respectively. For the sake of clarity, we only choose the first 10 matching keypoints. As we can see from Fig. 3, the numbers of matched keypoints extracted by SIFT, PCA-SIFT and AED are 4, 8 and 10, respectively.

Scale and Rotation invariance The second set of experiments is conducted on a set of bark images for investigating the performance of SIFT, PCA-SIFT and AED under Scale and Rotation invariance. Fig. 6 shows that AED and PCA-SIFT have 10 matches, SIFT has 9 matches. So both the AED and PCA-SIFT are slightly better than SIFT.

Blur invariance The third set of experiments is conducted on a set of tree images for examining the impact of the image blur on the performance measures. Fig. 7 shows the results of the SIFT, PCA-SIFT and AED, all of them have 10 matches. Thus, all 3 algorithms are well-suited for matching of images with blurring differences.

Viewpoint change The fourth set of experiments is conducted on a set of wall images where the images are taken from different viewpoints. The results show that SIFT and AED have 10 matches which are better than 8 matches by PCA-SIFT. The experimental results are shown in Fig. 8. As we can be seen in Fig. 6, the number of matched keypoints obtained by SIFT, PCA-SIFT and AED are 10, 8 and 10, respectively.

Light change The fifth set of experiments is conducted

Table 1: Experimental results on various transformed images

Image	SIFT			PCA-SIFT			AED		
type	total	$r(\%)$	$1-p(\%)$	total	$r(\%)$	$1-p(\%)$	total	$r(\%)$	$1-p(\%)$
Walls	7890	36.5	52.8	6830	28.5	76.3	7910	41.9	48.9
Bikes	3105	23.9	69.5	3562	33.8	67.9	4021	41.8	49.5
Leuven	2234	40.8	49.8	3445	38.7	51.8	2876	55.8	39.7
Trees	11321	9.23	87.5	7561	11.4	79.7	7443	17.3	78.3
Bark	4267	15.1	77.6	3598	7.81	88.4	3601	23.3	78.9
Graf	2843	33.6	58.2	3354	15.6	76.1	2581	35.6	59.9

on a set of Leuven images. Fig. 9 shows the results of the SIFT, PCA-SIFT and AED, all of them have produced 10 matches, so they are well-suited for matching images with illumination changes.

Compression The sixth set of experiments is carried out using a set of UBC images for investigating the performances of SIFT, PCA-SIFT and AED under image degradations caused by JPEG compression. Fig. 10 shows that all of the methods work well for the image compression. Since they all return 10 matched keypoints.

Considering the results obtained by SIFT, PCA-SIFT and AED under image blur, changing in lighting, and compression distortion, we see that SIFT, PCA-SIFT, and AED all have similar performances. AED produces the best result in affine transformation. AE-SIFT is better than SIFT when changing the scale and rotation. Therefore, among all the tested method, AED is the most distinctive, and is more robust to image deformation and more compact than the SIFT descriptor. It is also better in true positives and precision compared to the other methods. The precision of the proposed method is never lower than those of other methods. A detector with a higher precision is evidence that the detected keypoints are more accurate and stable than those obtained by other methods. Results for AED are given in the Table 1. The total number of features detected, true positive, and precision are calculated for comparison and listed in Table 1.

5. Conclusions

A rotation and scale invariant feature detector based on autoencoder was proposed. Instead of using SIFT's weighted histograms, an autoencoder is used to generate local image descriptors. The advantage of the proposed descriptor is its high precision and low dimensionality, resulting in fewer incorrect matches. Unlike SIFT, the proposed method uses the autoencoder to generate a feature vector, which has only 36 dimensions. As a result, it speeds up image retrieval while increasing repeatability. The presented preliminary results show the effectiveness of the proposed method over state-of-the-art methods.

References

[1] Tuytelaars T, Van Gool L. Matching widely separated views based on affine invariant regions[J]. International journal of computer vision, 2004, 59(1): 61-85.

[2] Lowe D G. Object recognition from local scale-invariant features[C]//Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Ieee, 1999, 2: 1150-1157.

[3] Mikolajczyk K, Schmid C. Indexing based on scale invariant keypoints[C]//Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, 2001, 1: 525-531.

[4] Mikolajczyk K, Schmid C. Scale & affine invariant keypoint detectors[J]. International journal of computer vision, 2004, 60(1): 63-86.

[5] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[6] Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey[J]. Foundations and Trends® in Computer Graphics and Vision, 2008, 3(3): 177-280.

[7] Freeman W T, Adelson E H. The design and use of steerable filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1991 (9): 891-906.

[8] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In Proceedings of European Conference on Computer Vision, 1996.

[9] F. Schaffalitzky and A. Zisserman. Multi-view matching for un- ordered image sets. In Proceedings of European Conference on Com- puter Vision, volume 1, pages 414, C431. Springer-Verlag, 2002.

[10] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005, 27(10): 1615-1630.

[11] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors[C]. Proceedings of the IEEE computer society conference on Computer vision and pattern recognition, 2004 Pages 506-513.

[12] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005, 27(10): 1615-1630.

[13] Liang J, Kelly K. Training Stacked Denoising Autoencoders for Representation Learning[J].

[14] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors[C] Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, 2004, 2: II-506-II-513 Vol. 2.

[15] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[16] Hagan M T, Menhaj M B. Training feedforward networks with the Marquardt algorithm[J]. Neural Networks, IEEE Transactionson, 1994, 5(6): 989-993.

[17] Smith J R, Chang S F. Tools and techniques for color image retrieval[C] Electronic Imaging: Science & Technology. International Society for Optics and Photonics, 1996: 426-437.

[18] Trzcinski T, Christoudias M, Lepetit V. Learning image descriptors with boosting[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(3): 597-610.

A New Image Descriptor for Image Retrieval

Chenyang Zhao[1], **A. Ardeshir Goshtasby**[1], **Shaodan Zhai**[1]
[1]Computer Science and Engineering, Wright State University, Dayton, OH, USA

Abstract— *Development of various image descriptors has greatly contributed to the advancement in image retrieval. Image descriptors can be made invariant to various image changes. In this paper, a new image descriptor is described and its application in image retrieval is evaluated. The descriptor uses the autoencoder concept to reduce the dimension of the feature vector measuing various properties of a keypoint. The new descriptor is found to have higher precision and recall rates when compared to the SIFT descriptor in image retrieval. Moreover, the descriptor is about three times faster than the SIFT descriptor, and by using the codebook concept, it can be made even faster.*

Keywords: Image retrieval, Image descriptors, Autoencoder, Codebook representation

1. Introduction

With advancements in digital imaging technology, content-based image retrieval has received much attention during recent years. Image retrieval is the process of finding images in a database that are similar to a query image. Image retrieval methods can be categorized into Text-Based Image Retrieval (TBIR) and Content Based Image Retrieval (CBIR). Text-based image retrieval can be traced back to the late 1970s [1].

In TBIR, images are manually annotated and then searched by text. If images are annotated correctly, search results can be quite accurate; however, TBIR has some limitations. First, the amount of labor required to manually annotate all images in a database can be tremendous. Second, the inaccuracy caused by the subjectivity of human perception when generating the descriptions can be a source of error in image retrieval. Different individuals can have different interpretations of an image.

To overcome these limitations and drawbacks of TBIR, content-based image retrieval (CBIR) has been suggested. CBIR uses an images contents to search for similar images in a database [2]. An image in a database is represented by a feature vector. The size of the feature vector is usually much smaller than the size of the image it is representing. The feature vector is used to compare and find similar images in a database.

Content-based image retrieval plays an important role in multimedia database systems. In CBIR, low-level features (such as colors, textures, and shapes) are used to describe an images contents. However, these low-level features can hardly describe the semantic concepts of an image [3]. In recent years, the mid-level features, such as SIFT descriptors, have attracted much attentions in image retrieval.

A feature measures a property of an image, either globally or locally [4]. Image features determine the performance of an image retrieval system; therefore, use of appropriate features is important in the success of CBIR. Image features such as color, shape, and texture can be used to match a query image to other images. Features are extracted automatically using computer vision techniques, and the similarity of the features is determined with those of features in images in the database [5]. In order to improve the retrieval performance, various combinations of features have been proposed.

In this paper, a new compressed image descriptor is proposed. Keypoints are extracted in the query image and in images in the database using the difference of Gaussian (DoG) operator, a descriptor is generated for each keypoint in the query image and in images in the database, the stacked autoencoder method is used to reduce the size of the image descriptors, and finally, the codebook method is used to speed up the retrieval process.

The remainder of this paper is organized as follows. Section 2 provides the background information about image retrieval, Section 3 lays out details of the proposed descriptor and provides information about the retrieval procedure and the codebook idea, Section 4 presents the results obtained by the proposed descriptor in image retrieval, and finally, Section 5 discusses strengths and weaknesses for the proposed descriptor.

2. Background

When a query image is provided, CBIR requires searching the database for the relevant images by using the contents of the images rather than relying on human-input metadata (such as captions or keywords). Image features and feature matching are important in CBIR. The performance of an image retrieval system depends on two factors: a suitable feature descriptor and a powerful feature matching strategy [6].

2.1 Keypoints detection

The first step in image retrieval is to detect keypoints in images. Various keypoint detectors have been proposed. In this paper, we use the DoG detector. The detector has three steps:(1) Scale-space extrema detection; (2) Keypoint localization; (3) Orientation assignment.

In the first step, keypoints are searched in scale-space by using the DoG operator. In order to detect extrema points from a stack of DOG images a point in the scale-space that is locally maximum within a $3 \times 3 \times 3$ neighborhood is taken to represent a keypoint. In the second step, a keypoint is determined by fitting a 3-D quadratic function using a second order Taylor expansion with the origin at the point of interest. Then, local extrema that correspond to weak edges are discarded. In the third step, a dominant orientation is assigned to each keypoint using the histogram of gradient directions in the neighborhood of the keypoint.

2.2 Creating a compressed image descriptor using a stacked autoencoder

An autoencoder is a specific form of an artificial neural network [7]. It is an unsupervised learning method that sets the target values to the input values. An autoencoder is composed of input layers, hidden layers, and output layers. A backpropagation algorithm is usually applied to train an autoencoder by propagating the error information (gradients) from the output layers back to the input layers [8] by using a gradient descent approach.

For an autoencoder with multiple hidden layers, it is difficult to optimize the weights by the back propagation algorithm. First, the back propagation algorithm does not work well in a deep neural network structure due to the diffusion of the gradients. Second, the optimization may get stuck at a false local minimum when the neural network is initialized with random weights. To alleviate this problem, Hilton [7] proposed a stacked autoencoder approach.

The motivation behind this is, gradient descent tends to work well when the initial weights are close to a good solution. To find initial weights for this *good solution*, a layer-wise pre-training process is used. This specifically consists of the following steps:

1) Train the bottom-most autoencoder, which consists of the input layer and the bottom-most hidden layer.
2) Remove the decoder layer of the trained autoencoder; and then construct a new autoencoder by taking the hidden layer of the previous autoencoder as input.
3) Train the new autoencoder.
4) Repeat Step 2–3 until all weights are pre-trained.

After the pre-training process, the next stage is to fine tune the weights of the network in a supervised fashion using the back propagation algorithm. In this stage, the weights obtained from pre-training are assigned as the initial weights, and the goal is to minimize the reconstruction error.

After detecting the keypoints, a local gradient patch centered at each keypoint is selected. Then, the image descriptor for the gradient patch is generated. For offline training, 40000 41×41 patches were collected from diverse images. For each detected keypoint, the gradient patch centered at the keypoint is extracted. Then, the patch is rotated so its main orientation aligns with the x-axis, and scaled according to the keypoints scale. For an image patch with size 39×39, the feature vector has 3042-values $(39 \times 39 \times 2)$. A 3042-512-512-256-256-128-36 stack autoencoder is trained to extract feature vectors from the image. The feature vector can be computed by the local gradient image of the patch. We use the stacked autoencoder to project the feature vector from 3042 to 36. So our compressed autoencoder descriptor(AED) is significantly smaller than the standard SIFT descriptor with 128 values.

3. The retrieval process

The image retrieval process consists of two steps: the offline step and the online step. The offline step can be performed once ahead of time, and updated as new images arrive periodically. Thus, the efficiency of the offline step is not crucial. Since the online step will be performed when a query image arrives, this step should be carried out very efficiently. Figure 1 illustrates the image retrieval task. In the following, details of the offline step and the online step are provided.

3.1 Offline step

Given a database containing a large number of images, the offline step proceeds as follows:

- Using the SIFT detector extract keypoints in all images in the database.
- For each image, extract the local gradient patch centered at each keypoint and generate the gradient vector for each keypoint.
- Project each gradient vector to a compressed feature vector using the pre-trained autoencoder.
- Store the compressed feature vectors with the images.

3.2 Online step

In the online step, given a query image it is required to find all relevant images in the database. This is achieved as follows:

- Using the SIFT detector extract keypoints in the query image.
- Extract the local gradient patch centered at each keypoint, and generate the gradient vector for each keypoint.
- Project each gradient vector to a compressed feature vector by using the pre-trained autoencoder.
- For each image in the database:
  - Compute the similarity between this image and the query image. In the two images, compare each feature vector in one image with all feature vectors in the other image. If the Euclidean distance between two feature vectors is smaller than a required threshold value, declare a match. The similarity of the two images is calculated by the number of

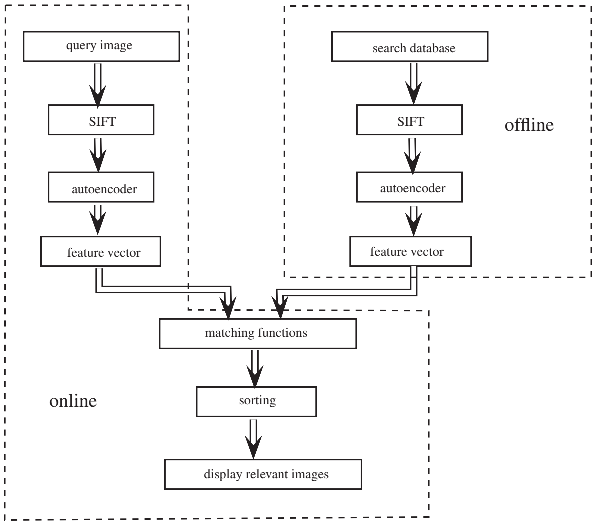


Fig. 1: Flowchart of content-based image retrieval system using the AED method.

matches. If the number of matches is greater than 0, return the image.

- Sort the retrieved images in descending order of their similarities with the query image.
- Display the relevant images.

In this image retrieval process, the bottleneck in computations is in the computational of the similarity between the query image and all the images in the database. By employing AED, the relevant images are obtained more efficiently than by SIFT.

3.3 Codebook image representation

Consider the matching process shown in Figure 1. Each image in the search database is compared with the query image. This is intractable in practice because a search database often contains a very large number of images. In order to speed up the matching process, a codebook mapping method [9], [10] is employed. The main idea is to map each image into a fixed-length vector, so that the similarity of two images can be efficiently determined by two vectors. Matching is then performed only on those images that produce the highest similarity scores.

Figure 2 shows the image retrieval process using the codebook mapping approach. In order to map an image to a fixed-length vector, we first cluster the (36-dimensional) AED feature vector of each keypoint for all images in the search database. We employ k-means clustering and set the number of clusters to L [1].

Each AED feature vector can then be assigned to one of the L clusters. In this manner, an image can be described by a frequency distribution of these L labels. This L-dimensional vector is called *codebook* or *bag-of-words*. This notion comes from the natural language processing area, and is a popular way of representing a document by it's word frequency distribution, ignoring orders.

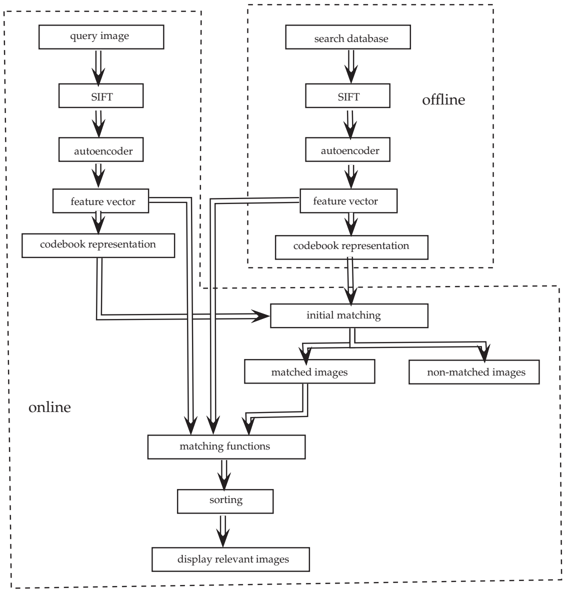


Fig. 2: Flowchart of content-based image retrieval system using AED algorithm and codebook representation.

To employ this codebook mapping, there is one more step in the image retrieval's offline step. After AED vectors of all images in the search database are obtained and stored, k-means clustering is performed on the AED feature vectors and group them into L clusters. Then, these L-dimensional codebook vectors and the clustering model are stored.

In the new image retrieval system, the online step is updated as follows:

- Using the SIFT detector extract keypoints of the query image.
- Extract the local gradient patch centered at each keypoint, and generate a gradient vector for each keypoint.
- Project each gradient patch to an AED vector using the pretrained autoencoder.
- Map the query image to an L-dimensional codebook representation. More specifically, compute the distance between each AED vector in the query image and the L cluster centers generated by the k-means algorithm in the offline step. Then, assign each AED vector to the cluster with its center closest to the AED vector.
- Compute the similarity between the query image and all the images in the database using their codebook representations. Select N images with the highest similarity scores as the initially matched images.
- For each initial matched image in the search database:

[1] We choose L to be 1000, suggested in [9]

   - Compute the similarity between this image and the query image. For two images, we compare each feature vector in one image with all feature vectors in the other image. If the Euclidean distance between two feature vectors is smaller than the chosen threshold, a match is declared. The similarity of the two images is calculated by the number of matches. If the number of matches is greater than 0, the image is returned.
- Sort the retrieved images in descending order of their similarity with the query image.
- Display the relevant images.

4. Evaluation Metrics

In the experiments, we use the popular metrics Precision vs. Recall to evaluate the performance of an image retrieval system. For any query image, Precision is the ratio of the number of relevant images retrieved to the total number of images (including relevant and irrelevant) retrieved:

$$Precision = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}$$

where an image is considered relevant if and only if it is in the same group as the query image. Intuitively, Precision measures the quality of the retrieved images. The larger its value, the better the quality of the retrieved images. For simplicity, Precision of a query image is set to 1 if no images are retrieved.

Recall is the ratio of the number of relevant images retrieved to the total number of relevant images retrieved in the entire search database:

$$Recall = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images}}$$

In some real-world applications, the total number of relevant images is usually unknown. For example, if the search space is the entire Internet, it is hard to know how many images are relevant to a query images. However, in our application, Recall is computable as we assume that only if images in the same group are relevant. Recall measures the ability to find relevant images by the image retrieval system.

With the Precision and Recall for a query image, we can compute the average Precision and Recall over all the queries. Ideally, it is desired that an image retrieval system provide both a high Precision and a high Recall rate, but this is often an unattainable goal in practice. For the extreme cases, if no images are retrieved, the Precision is set to 1, but obviously this image retrieval system is useless. On the other hand, if an image retrieval system retrieve all the images in the database, then Recall is always 1, but Precision can be close to 0. Therefore, there is usually a trade-off between precision and recall, and we employ the Precision vs. Recall curves to measure the performance of an image retrieval system. The closer the curve is to the top of the chart it indicates a better performance.

5. Results

In order to evaluate the performances of the proposed descriptor, the INRIA Holidays dataset and ORL database are used. The experiment is to compare the retrieval performance between our descriptor and the SIFT and PCA-SIFT [11] descriptors. For the image retrieval systems that use SIFT and PCA-SIFT, they are very similar to the process of Figure 1. The only difference is replacing the AED module to SIFT or PCA-SIFT.

For a given dataset, we first divide a dataset into two parts. The first part contains all the query images, and the rest is used to search the database. For a query image, the goal is to try to find all relevant images for the query. Taking the ORL dataset as an example, there are 40 query images in total, and the remaining 360 images represent the database. For a query image, we intend to find the other 9 images of the same person from the entire 360 images.

5.1 Image retrieval results using the Holidays dataset

The Holidays dataset [2] was created for the ANR RAFFUT project. It contains 1491 personal vacation photos with a very large variety of scene types, including natural, food, water and building. There are totally 500 image groups in this dataset, each of them represents a distinct scene of object. For each image group there are 2 to 10 images, and these images were taken from different viewpoints or by varying the lighting. We choose the first image of each group as the query, and 1 to 9 images of the group as the correct retrieval results.

Figures 3 and 4 illustrate the retrieved images of SIFT, PCA-SIFT and AED algorithms. For figures 3 and 4, AED descriptor performs especially well, finding all retrieved images.

The Precision vs. Recall curves of SIFT, PCA-SIFT, and AED are shown in Figure 5. For each query image, each algorithm is allowed to return at most 9 images. This result shows that AED is comparable or better than SIFT and PCA-SIFT in image retrieval when using the Holidays dataset. We believe this is due to AED's high matching accuracy at the keypoint level, which also translates to high retrieval results.

Since the codebook mapping is an approximate method, the matching performance may worsen when compared to the naive matching approach. Figure 6 shows the Precision vs. Recall curves of AED and AED with codebook mapping. After initial matching with codebook representation, some relevant images are dropped incorrectly. However, this loss of accuracy may be acceptable when considering improvement run time.

[2] https://lear.inrialpes.fr/ jegou/data.php

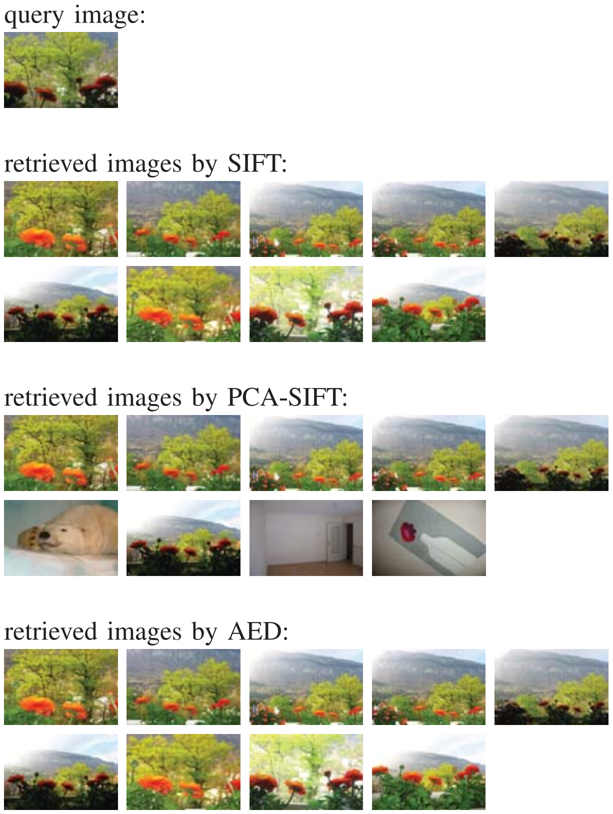


Fig. 3: Retrieved images by SIFT, PCA-SIFT, and AED for the query image 136000.pgm.

5.2 Image retrieval results using ORL Database of Faces

The ORL database was created by AT&T Laboratories [3]. It contains a set of face images of 40 individuals taken between April 1992 and April 1994 at the lab. For each person, there are 10 different face images, and each image contains just one face. The 10 face images for a person are different due to four factors: 1) images were taken at different times; 2) varying the lighting; 3) facial expressions, such as open/closed eyes, smiling/not smiling; 4) facial details, such as glasses/no glasses. All the images have black homogeneous background. The size of each image is 92×112 pixels, with 256 grey levels per pixel. We divide the face images into 40 groups, each group contains 10 images for one person. For each group, we choose the first image as the query image, and thus there are 9 relevant images for any query image.

From the experimental results using the ORL dataset, the Precision vs. Recall curves of SIFT, PCA-SIFT, and AED are plotted in Figure 9. As before, at most 9 images are retrieved for each query image. The image retrieval task on

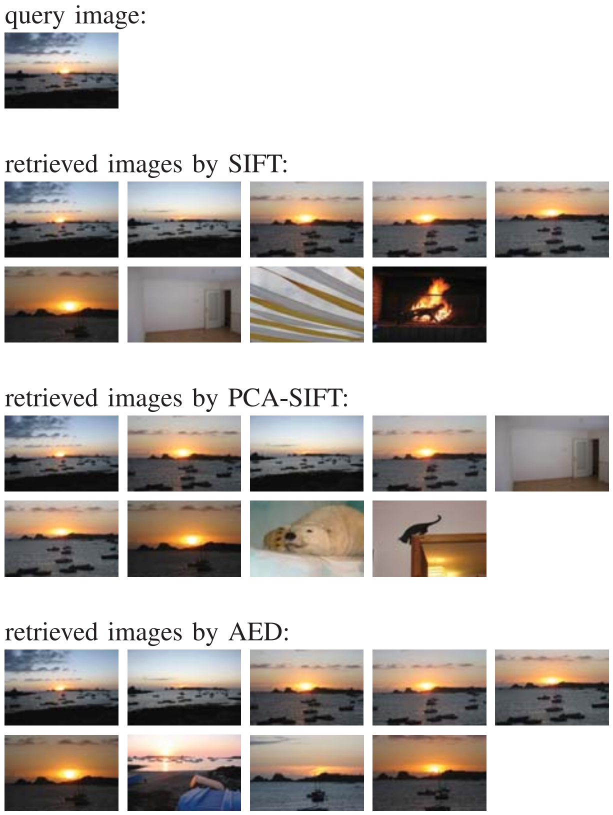


Fig. 4: Retrieved images by SIFT, PCA-SIFT, and AED for the query image 132500.pgm.

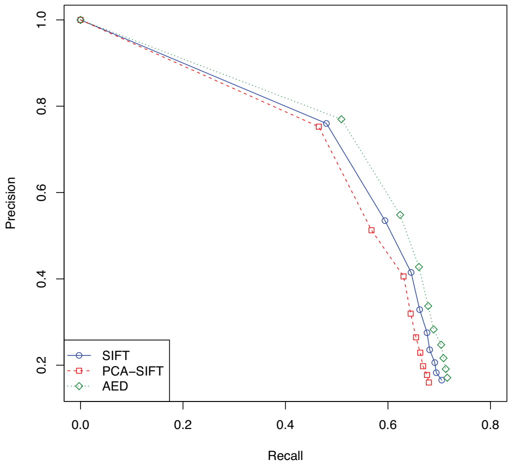


Fig. 5: Precision vs. Recall curves of SIFT, PCA-SIFT, and AED using the Holidays dataset.

[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

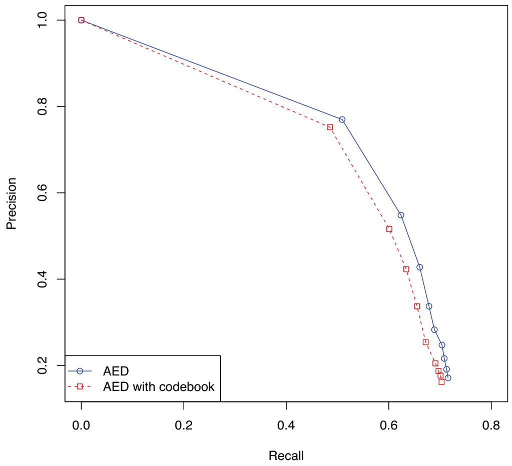


Fig. 6: Precision vs. Recall curves of AED and AED with codebook mapping when using the Holidays dataset.

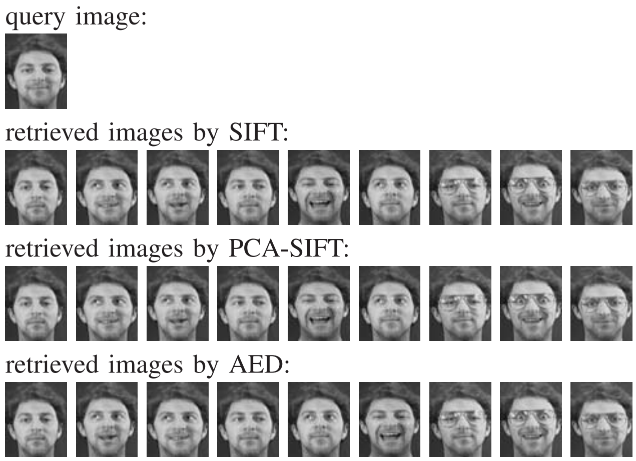


Fig. 7: Retrieved images by SIFT, PCA-SIFT, and AED for query image of the 7th and 2nd persons.

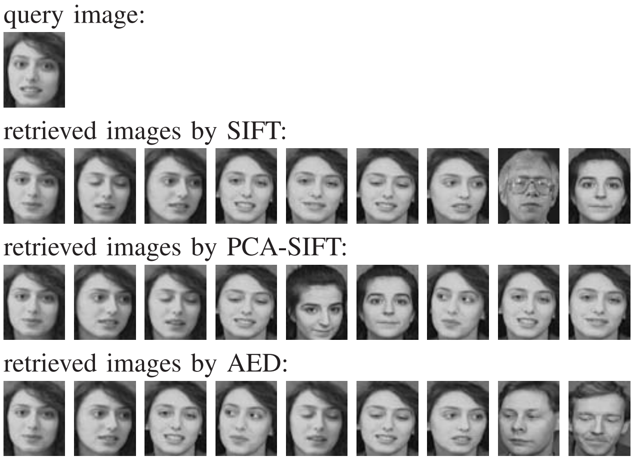


Fig. 8: Retrieved images by SIFT, PCA-SIFT, and AED for the query image of the 8th and the 3rd persons.

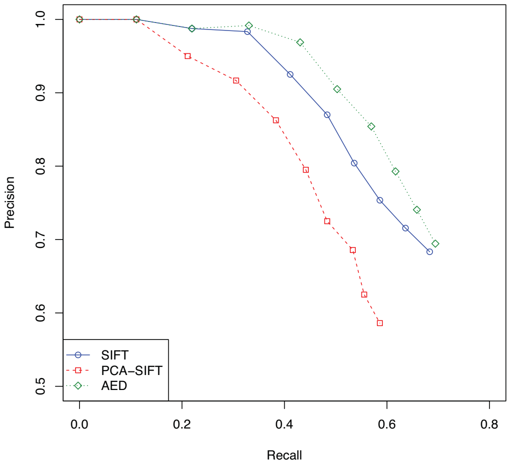


Fig. 9: Precision vs. Recall curves of SIFT, PCA-SIFT, and AED when using the ORL dataset.

this dataset is easier than the previous Holidays dataset. The retrieved results are perfect for each algorithm if we only consider the first retrieved image. Considering the overall retrieval results, AED performs better than SIFT and PCA-SIFT on this dataset.

Figure 8 shows the retrieved images for the query image of the 8th person and the query image of the 3rd person. The image retrieval tasks for these two query images are more difficult, and each algorithm retrieves some irrelevant results. For the query image of the 8th person, 7 relevant images are retrieved by all the three algorithms. However, PCA-SIFT performs worse than the other two methods since the irrelevant images rank higher. For the query image of the 3rd person, 6 relevant images are retrieved for both SIFT and PCA-SIFT. AED performs better for this query image since 8 relevant images are retrieved.

5.3 Run-Time Analysis

Besides the accuracy, an important performance measure of an image retrieval system is its speed. As was discussed before, the matching process is the bottleneck in image retrieval since it has to be executed online. To evaluate the time performance of the image retrieval time, each method was tested on a Red Hat Linux server with Intel Xecon with 5650 CPUs. The algorithm was implemented in C++. Here an example using two images from the Holidays dataset

	Run time per image	Run time per point
SIFT	7.246	7.44e-7
PCA-SIFT	2.206	2.26e-7
AED	2.182	2.24e-7

Table 1: Matching run time in seconds for SIFT, PCA-SIFT, and AED when using two images from the Holidays dataset.

is given. Overall, 2967 and 3284 keypoints were detected by the SIFT detector, requiring about 9.7 million point comparisons during the matching process. Table 1 shows the run times in ms for SIFT, PCA-SIFT, and AED. PCA-SIFT and AED spent almost the same time in matching as the feature vector in each keypoint had 36 dimensions by both methods. SIFT is about 3 times slower than PCA-SIFT and AED. This is due to associating a 128-dimensional feature vector to each keypoint. For each query image, there was a need to use all 1491 images in Holidays dataset, requiring about one hour matching time for AED and PCA-SIFT, and three hours for SIFT. Obviously this run time is not practical in real applications.

When we use the codebook method, and set N to 100 in our image retrieval system, the query image is compared to 100 images instead of all images in search database. For the Holidays dataset, this speeded up the process by about 15 times.

6. Conclusions

With development of various image descriptors, significant progress has been made in image retrieval. Image features generated by local image descriptors are generally invariant to different image transformations. This paper described and evaluated a new image descriptor using the autoencoder concept.

The stack autoencoder is used to reduce the dimension of the feature vectors describing the properties of the neighborhood of a keypoint. Compared to the SIFT descriptor, which produces a 128-dimensional feature vector, the proposed encoder produces a 36-dimensional feature vector. As a result, the proposed descriptor has a considerably lower computational requirement than the SIFT descriptor.

When used in image retrieval, the proposed descriptor is found to have a higher combined precision and recall rate than SIFT. Moreover, the proposed descriptor is about three times faster than the SIFT descriptor. By using the codebook method, the speed of the proposed descriptor in image retrieval is further increased.

References

[1] Rui Y, Huang T S, Chang S F. Image retrieval: Current techniques, promising directions, and open issues[J]. Journal of visual communication and image representation, 1999, 10(1): 39-62.

[2] Smeulders A W M, Worring M, Santini S, et al. Content-based image retrieval at the end of the early years[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000, 22(12): 1349-1380.

[3] Yuan X, Yu J, Qin Z, et al. A SIFT-LBP image retrieval model based on bag of features[C]// IEEE International Conference on Image Processing. 2011.

[4] Datta R, Joshi D, Li J, et al. Image retrieval: Ideas, influences, and trends of the new age[J]. ACM Computing Surveys (CSUR), 2008, 40(2): 5.

[5] Rui Y, Huang T S, Chang S F. Image retrieval: Current techniques, promising directions, and open issues[J]. Journal of visual communication and image representation, 1999, 10(1): 39-62.

[6] Jegou H, Douze M, Schmid C. Hamming embedding and weak geometric consistency for large scale image search[M] Computer Vision?ECCV 2008. Springer Berlin Heidelberg, 2008: 304-317.

[7] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[8] Hagan M T, Menhaj M B. Training feedforward networks with the Marquardt algorithm[J]. Neural Networks, IEEE Transactions on, 1994, 5(6): 989-993.

[9] Collins J, Okada K. A Comparative Study of Similarity Measures for Content-Based Medical Image Retrieval[C] CLEF (Online Working Notes/Labs/Workshop). 2012.

[10] Yuan X, Yu J, Qin Z, et al. A SIFT-LBP image retrieval model based on bag of features[C] IEEE International Conference on Image Processing. 2011.

[11] Ke Y, Sukthankar R. PCA-SIFT: A more distinctive representation for local image descriptors[C] Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, 2004, 2: II-506-II-513 Vol. 2.

SESSION

IMAGING SCIENCE AND MEDICAL APPLICATIONS + ASSISTIVE TECHNOLOGIES

Chair(s)

TBA

Toward Device Assisted Identification of Grocery Store Sections and Items for the Visually Impaired

Wadee Alhalabi [1], **Dalia Attas** [1]

[1]Department of Computer Science, Faculty of Computing and IT, King Abdul-Aziz University, Jeddah, KSA

Abstract - *There are systems developed to assist the visually impaired in grocery shopping. The developed systems require physical work from the users, wireless connections and products database to obtain the products information. Here comes the need for a system that assist the visually impaired person in grocery shopping without any additional devices. The system should use object recognition algorithms instead of wireless connections and database to recognize products. In the paper, a software system was created to solve the problem of assisting the visually impaired in grocery shopping. The created software accomplished only the mission of recognizing the grocery products on shelves. The paper recommends implementing a shopping cart consists of three cameras installed vertically on one side of the cart. We conducted a comparison between two object recognition algorithms to recognize products on the aisle shelves. A multimodal system created to fuse the results of the used object recognition algorithms.*

Keywords: Grocery Shopping; Visually Impaired; Object Recognition Algorithms; Interest point Matching; Optical Character Recognition

1 Introduction

The World Health Organization stated that there are 285 million people worldwide visually impaired (39 million blind and 246 with low vision), from which 82% are in their 50s or above [1]. The visually impaired people in the age of 50 and above are considered to require assistance for the daily routine tasks such as grocery shopping. The more convenient way for the visually impaired to shop in the grocery store is to go the store themselves. That gives them the freedom to select the products without the need for a previous task such as filling an electronic shopping list. Conversely, there are many obstacles that face the visually impaired in attending the grocery store such as the lack of assisting tools that can help them performing such trivial tasks. Furthermore, there are no helping resources in locating items that the visually impaired desire to purchase.

There are several institutions active in research and development for systems assisting the visually impaired persons to perform individually grocery shopping, e.g., shopping in supermarkets. Some of these efforts deal with helping the person to navigate inside the supermarket while others focus on assisting the person in locating the required product. Some systems require physical work from the user. Additionally, some systems rely on wireless connections and databases to recognize products.

There are different object recognition algorithms that can be used to detect objects using image features. Usually, the grocery products images show the text written on the products that can be recognized as text features, and the overall shape of the products that can be recognized by visual features. The visual features can be used for recognition by tools such as feature matching algorithms. On the other hand, text features can be used to recognize the text written on products through optical character recognition (OCR) algorithms.

The main paper objectives are to develop a software system that can perform three essential tasks: i) Announcing the aisle category name to the user, ii) Finding the user desired product on the shelf, iii) Guiding the user to the product location. To recognize the products (second task), we compared the visual features recognition and textual features recognition. The algorithms implemented under 1550 product images collected from a dataset on the web. The visual features are compared using Interest Point Matching (IPM) algorithm called Speeded-up Robust Features (SURF) with product images on a dataset for recognition. We compared the textual features with a name of the user desired product using Optical Character Recognition (OCR) engine. Also, a multimodal system consisting of the fusion between the visual feature recognition and textual feature recognition via an enhanced fusion level procedure was developed.

The rest of the paper organized as follows: Section 2 describes related systems implemented to assist the visually impaired in grocery shopping and the common algorithms used for object recognition. Section 3 described the workflow of the developed system and the dataset used for implementation. In section 4, the results are showed using confusion matrix rates and the performance rates. Additionally, section 4 shows the fusion results between IPM and OCR. Furthermore, Section 5 discuss the obtained results.

Finally, section 6 concludes the paper and suggest future work.

2 Literature Review

2.1 Related systems

There are several institutions active in research and development for assisting the visually impaired in grocery shopping. Some of these efforts deal with assisting the person to navigate inside the supermarket while others focus on assisting the person in locating the required product.

One of the leading efforts in this domain is the RoboCart [2]. RoboCart is a robot that assists the visually impaired by helping the user to navigate the grocery store to purchase the desired items. RFID (Radio Frequency Identification) tags were attached to the store shelves to help localize the products in the store. According to the RoboCart developers, the system provides additional assistance but does not replace the white cane and guide dogs. RoboCart can assist the user to reach the grocery store sections, but cannot help the user in retrieving a particular product from the store shelves. The developers of RoboCart outspread the project to involve a new system called ShopTalk [3]. ShopTalk is a wearable system developed to assist the visually impaired people to find a product on the grocery store shelves. It uses speech directions and a map of the grocery store to find the aisle of the required item. The testing results showed that the users were able to find all the required products using store map and barcode scans. The system cannot work in other stores because it is restricted to a particular store map. In 2006, a system called Trinetra [4] was developed to assists the visually impaired in shopping at grocery stores independently and cost-effectively using COTS (Commercial off the shelf) products. The system used both RFID tags and UPC (Universal Product Code) barcode for product identification. The users will need to seek for help in determining the required aisle and shelf. UPC barcode is tagged in every product, in the grocery store, by the product manufacturer. There is an online database that provides the integration between the UPC and their corresponding description of the product [4] [5]. The barcode solution considered to be inconvenient because the user will need help in locating the required aisle and shelf. By the year 2007, a system called GroZi [6] developed to use computer vision techniques for assisting the visually impaired in locating grocery items. GroZi consists of a glove containing vibrating motors sensible of the four directions, a camera placed above the glove, Bluetooth headset for feedback, and a battery pack placed on a belt. To test the system, the authors replaced the computer vision software with a Remote Sighted Guide (RSG). The RSG contains a person viewing what the camera is showing and control box aimed to provide audio and haptic feedback. The experiments of the device showed that it was hard for the user to find the items primarily. After the user had adapted to the system, he was able to find the items [6]. GroZi only specifies to the user the aisle section and the location of the requested product on the shopping list. Furthermore, the RSG technique is not convenient because it cannot work without human control. Sreekar Krishna created a system called "Wearable Wireless RFID System for Accessible Shopping Environments" [7] in 2008. The system aims to navigate the visually impaired persons in grocery shopping by retrieving the product information. The system first part uses the RFID tags to identify the products. The second part recognizes the product using a centralized store server. The system experimented according to the RFID tags detection [7]. The main disadvantage of the RFID tags is that they require many modifications in the store. Also, the store server may need a lot of data to be collected. By the year 2011, a system called BlindShopping [8] was developed to customize mobile techniques that assist the visually impaired at grocery shopping. The authors consider the system to have a minimal cost and easily deployable. It consists of a white cane holding RFID reader, RFID tags attached to the supermarket floor and an application for the Smartphone [8]. The system was considered to be high cost and rough to deploy due to the number of equipment that need to implement in the supermarket such as the RFID tags on the floor, the QR codes or barcodes on the aisle shelves.

2.2 Interest Point Matching (IPM)

There are some algorithms developed for Object Recognition. The latest algorithms rely on image feature matching methods. The features were detected from both the images in the real scenes and the objects images and stored in a database. Then, image feature matching techniques are be used to match the features of the scenes against the objects features database. When an appropriate number of the match occur, a geometric transformation aligns the matched features together [9]. One of the images features used for matching images is interest points. Interest point matches the geometry of the comparable images using a transformation function that inputs the coordination of the matching points in both images. The point features can be called: interest point, the key point, the corner point, and control point. We will use the term interest point [10]. The interest point matching algorithms have three main steps [11]: Interest Point Detection, Interest Point Description, and Interest Point Matching. The most leading two IPM algorithms are Scale Invariant Feature Transform (SIFT) and Speeded-up Robust Features (SURF). The authors in [12] compared SURF with SIFT. The results showed that SURF performance in all the comparisons was better than the others. Based on the average recognition rate, SURF obtained 82.6%, and SIFT gained 78.1%. SURF proved to be a fast and accurate interest point detector and descriptor. For that reason, we tested the system using the SURF algorithm.

2.3 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technique used to detect and recognize texts in document images or scene images. There are a number of engines developed for OCR. The scanned text images are different from the scene text images according to the differences in font size, color, orientation, and the background disruption. K. Wang and S. Belongie [13] compared the performance of the two most known OCR engine (Tesseract and ABBYY Fine Reader) on ICDAR 2003 robust reading dataset and Street View Text dataset. The Street View Text dataset is outdoor image text dataset that shows high inconsistency and usually has low resolution. The results of the test showed that the accuracy of the Tesseract was 31.5 % while the ABBYY was 47.7% [13]. In the paper, we tested the OCR results using the ABBYY OCR engine.

3 Methodology

Our aim is to solve some issues that can disrupt the user while using the system. The proposed objective is to develop a software system that can detect and recognize grocery store sections and products on shelves for the visually impaired persons. The novel study is to compare two object recognition algorithms: OCR and IPM algorithms results on detecting grocery store products dataset. The comparison is aimed to justify which method is applicable for implementation and to show the result of both methods. A fusion multimodal was created to fuse between the IPM and OCR. Furthermore, to state if the fusion multimodal will perform better results than each algorithm alone. The user is supposed to push a shopping cart consisting of three cameras installed vertically on the right side of the cart and a laptop at the bottom of the cart as shown in Fig. 1. This mechanism allows to scan all the aisle products from top to the down of the aisle. Each camera is named to indicate its position in the cart (Top, Middle, and Bottom) for later processing. We did not implement the shopping cart in the system.

The system is started by the input of the three camera views of the supermarket shelves from the cart cameras in the form of video clips. The system convert each video clip to a number of image frames which do not contain redundant frames. Each frame is connected with the name of the camera that captured the view either Top Camera, Middle Camera or Bottom Camera. The system workflow is shown in Fig. 2. When the user enters the aisle, he or she does not know the aisle category. Therefore, the first task for the system is to announce the name of the aisle category. After that, the user has the option of either proceeding to a new aisle or selecting a product from the current aisle. The second task for the system is to retrieve the user desired product from the shelves. This task consists of two parts. First, the system must find the user required product on the shelf by object detection algorithms. The implemented algorithms are IPM and OCR. Second, the location of the found product on the shelf must be retrieved. The location can help the system to guide the user in front the shelf.

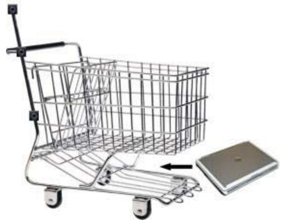


Fig. 1. System Shopping Cart

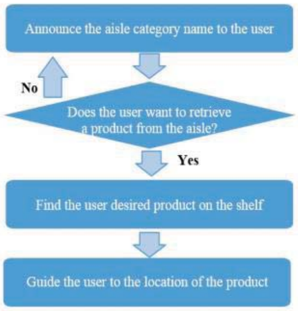

Announce the aisle category name to the user
No
Does the user want to retrieve a product from the aisle?
Yes
Find the user desired product on the shelf
Guide the user to the location of the product

Fig. 2. System Workflow

3.1 IPM

In the pre-processing stage, the scope is to detect the interest points and extract feature descriptors at the interest points of all the product images in the database as shown in Fig. 3. The feature descriptor provides the feature vector and the feature location. The system stores both values in the product images database with the relevant image record.

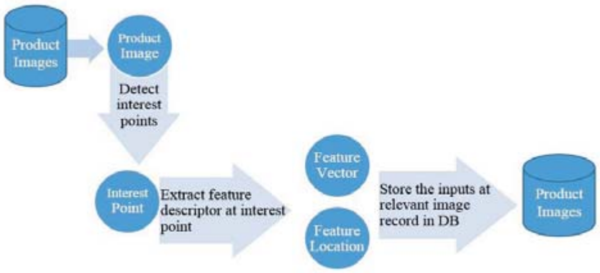


Fig. 3. Preprocessing Stage

The system starts by waiting for the user to announce the name of the product. The speech is then converted to text using speech to the text tool. When the system gets a shelf view image from the camera, the image is processed as shown in Fig. 4. If the system did not find a match, a new shelf view image is retrieved and the process is repeated until the user enters a new category area.

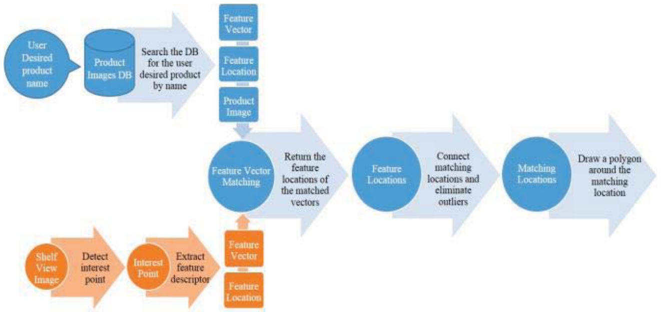

User Desired product name
Product Images DB
Search the DB for the user desired product by name
Feature Vector
Feature Location
Product Image
Feature Vector Matching
Return the feature locations of the matched vectors
Feature Locations
Connect matching locations and eliminate outliers
Matching Locations
Draw a polygon around the matching location
Shelf View Image
Detect interest point
Interest Point
Extract feature descriptor
Feature Vector
Feature Location

Fig. 4. Object Detection using IPM workflow

To detect the interest points, we used the SURF algorithm. To match between the descriptors, we use two matching strategies Nearest Nighbor (NN) and Nearest Nighbor Distance Ratio (NNDR) [14]. The threshold scale is from zero to one. We experimented 50 testing samples under different threshold values. When assigning the threshold too high, it will result in a many false positive and if we assigned the threshold too low, it will result in many false negatives. The Computer Vision Algorithms and Applications book [11] agreed on the same point of view. Therefore, we assigned the threshold to (0.6) because it gave us better true matching results. The matching was done based on matching interest point in objects and not per interest point. The IPM system was implemented using an MATLAB code performing the earlier described IPM procedure.

3.2 OCR

First, the system receives the shelf view image from the camera and the name of the desired product from the user. A speech to text tool is used to convert the speech to text. Then, the system extracts each product image from the shelf image view separately, to store it in its corresponding location in the image. The extraction is performed in two forms: the marketing form (MK) of the products and the planogram front face of products (PF). The complete process is shown in Fig. 5.

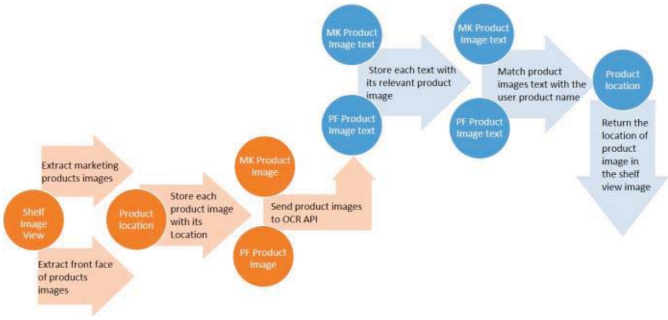

MK Product Image text
Store each text with its relevant product image
MK Product Image text
Match product images text with the user product name
Product location
PF Product Image text
PF Product Image text
Return the location of product image in the shelf view image
Extract marketing products images
MK Product Image
Shelf Image View
Product location
Store each product image with its Location
Send product images to OCR API
Extract front face of products images
PF Product Image

Fig. 5. OCR Procedure

When matching the product name given by user input with the retrieved OCR results from the engine, we used a threshold (0.6) similar to the IPM threshold. We will match the user desired product name under two circumstances: match the product Brand Name (BN) along with the Product Description (PD), and match the PD alone. For clarification, if 60% of the product description text exists in the OCR results it was accepted as a match in case of product description matching. If we matched the brand name and product description with the text retrieved from the OCR, the full brand name must be retrieved from OCR and the product description must apply under the threshold rule. A total of 1550 images was sent to the ABBYY OCR engine using an interface. The interface was built using Java and a login credentials to enter the ABBYY engine.

3.3 Dataset

We used item master dataset [15] instead of capturing product images from a real grocery store. The dataset contains more than 20,000 product images. We used 1550 images from the dataset. The dataset contains different product images from 688 categories. Examples of category names are Milk, Cheese, Syrups, and Herbs. The dataset classifies the products based on the identification number (ID), and Universal Product Code (UPC). Each product record contains the product name, brand, manufacturer, and more. Each product has a number of images from different views: planogram front (PF) as shown in Fig. 6, back, top, bottom, left, and right. We used 775 PF images for IPM evaluation. The IPM method used a database contained PF view images. In addition, it includes product images for marketing and commercial use that are photographed from ideal views as shown in Fig. 6. The MK images are also photographed from different views: front, left, and right. We used the MK images to resemble the shelf view image instead of photographing images from real grocery store shelves. The MK images used to create montage images as shown in Fig. 7. We used 775 MK images to create 100 montage images from different 100 categories. Each montage image contained eight products MK images from the same category. We used the product name in the dataset as the user desired product name to be matched with the OCR result. The product name in the dataset contains two parts: Brand Name (BN) and Product Description (PD).For example: if the product name is Starbucks Hot Cocoa Peppermint, BN is Starbucks and PD is Hot Cocoa Peppermint. The dataset was collected using an MATLAB code to store the images from URL links typed in the dataset excel sheet along with the required information. Furthermore, an MATLAB code was developed to create a 100 montage images from the stored product images using the montage function.

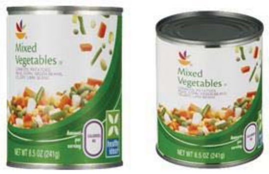


Fig. 6. Left: PF View product Image. Right: MK Product Image [15]

Fig. 7. Montage Image grouped from item master dataset [15]

We grouped the dataset to four situations that the user might encounter, each containing 25 montage images: If the user desired product exist in the montage image, we call the situation “exist”. If the user desired product does not exist in the montage image, we call the situation “not exist”. If a similar to the user desired product exist in the montage image, we call the situation “similar”. The similarity may be in the BN or PD. If the user desired product exist twice in the montage image, we call the situation “twice”.

4 Results

As mentioned in the dataset section, the IPM and OCR algorithms were applied to the item master dataset in order to create a system for visual impaired persons. The IPM matched the interest point features of the product images in the dataset with the images retrieved from the camera. On the other hand, OCR matched the text retrieved from the OCR engine with the name of the desired product. Each algorithm was tested under different conditions in order to ensure its correct functionality. We tested the IPM with two different matching strategies. The OCR was tested for two different image conditions and two different matching criteria. The analysis used 1550 images from the dataset belonging to 775 products. Four situations that may occur to the user in shopping were tested.

In order to develop the proposed system and to apply it to the real-world, six different matching algorithms strategies were used. These strategies are:

- *IPM NN:* The IPM SURF algorithm will be used to match between the montage images and the user desired product PF images. The NN strategy will be used to match between the two images.
- *IPM NNDR:* The IPM SURF algorithm will be used to match between the montage images and the user desired product PF images. The NNDR strategy will be used to match between the two images.
- *OCR MK BN+PD:* The ABBYY OCR engine will be used to recognize text in the eight MK product images of one montage image. Then, match between the retrieved text of the eight images and user desired product BN and PD text.
- *OCR PF BN+PD:* The ABBYY OCR engine will be used to recognize text in the eight PF product images of one montage image. Then, match between the retrieved text of the eight images and user desired product BN and PD text.
- *OCR MK PD:* The ABBYY OCR engine will be used to recognize text in the eight MK product images of one montage image. Then, match between the retrieved text of the eight images and user desired PD text.
- *OCR PF PD:* The ABBYY OCR engine will be used to recognize text in the eight PF product images of one montage image. Then, match between the retrieved text of the eight images and user desired PD text.

The performance of the experiment results were measured by calculating the performance rates: Precision, Recall, Fall out, Negative Predicted Values (NPV) and Accuracy. In order to determine the latter measures, we need to calculate the following four values from the confusion matrix: TP, FP, TN, and FN. In this context: i) TP means that the product is available, and the system found it; ii) FP means that the product is not available, but the system found it; iii) TN means that the product is not available, and the system did not found it; iv) FN means that the product is available, but the system did not detect it.

The confusion matrix results for the four situations is shown in Fig. 8. In the not exist and similar situations, the user desired product does not exist on the shelves. That means we can not calculate the TP and FN. The performance rate results for the four situations is shown in Fig. 9. In the not exist and similar situations, we can not calculate the precision, recall, and NPV because there is no value for TP and FN.

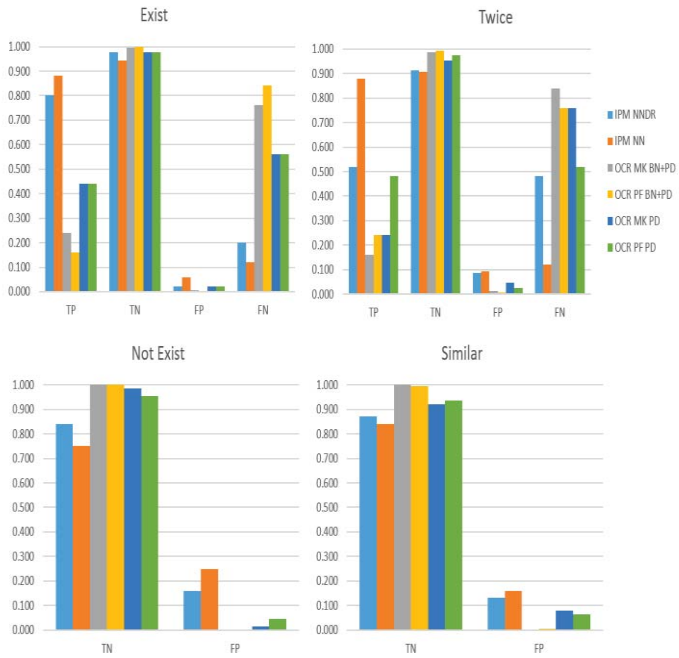


Fig. 8. Confusion Matrix Results for the four situations

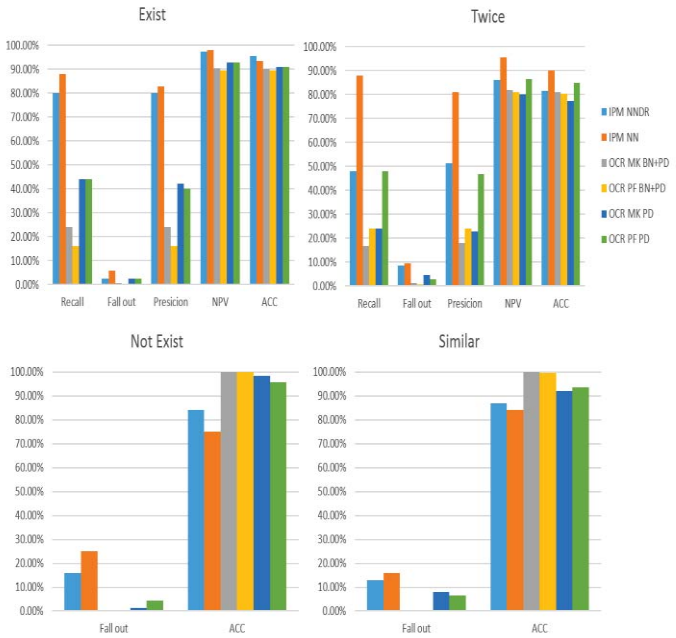


Fig. 9. Performance Rates for the four situations

In order to construct a multimodal system of multiple algorithms used in our system, the Decision Level Fusion in the parallel mode was applied using the Majority Voting method. The multimodal system starts by extracting the features either by IPM or OCR from the shelf view images. Then, the extracted features are matched with the user desired product image. Finally, the system generates a decision regarding the match. According to the matching results, the (OCR MK BN+PD) and (OCR PF BN+PD) algorithms were excluded from the fusion due to the large number of the false matching results. In addition, not exist and similar situations, cannot fused due to a large number of the false match resulted from comparing with IPM. Fig. 10 and Fig. 11 shows the fusion results for exist and similar situations.

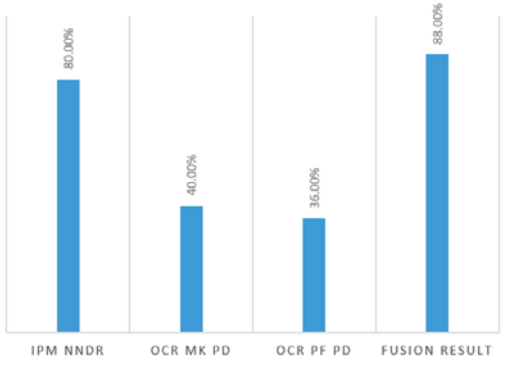


Fig. 10. Fusion Results for exist Situation

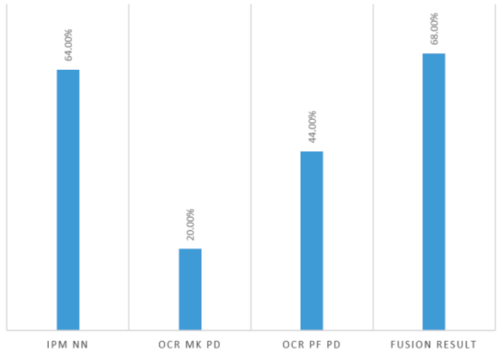


Fig. 11. Fusion Results for twice Situation

5 Discussion

The confusion matrix results showed that the IPM is better than the OCR when the system computes the positive results (TP and FN) in exist and twice situations. On the other hand, OCR is better for measuring the negative results (TN and FP) in the four situations. In case the desired product is available on the shelf, the performance rates showed that the IPM is more accurate than the OCR because IPM is capable to bring more either matching or not results. IPM using NNDR strategy is more accurate than NN in exist situation because the number of FP in NNDR strategy is less than in NN strategy. On the other hand, in not exist and similar situations, OCR algorithm is better than the IPM because it is capable to accurately distinguish between the TN and FP. The IPM using the NN strategy in exist situation resulted in a higher Recall, Precision, Fall out, and NPV, but the higher accuracy algorithm was IPM using the NNDR strategy because the number of the fall out that indicates the number of the false positives for IPM NN was low comparable with the other algorithms.

In the performance rates for the twice situation, the most efficient is IPM using NN strategy because the desired product is mentioned two times in the image and the NN strategy matched the montage image with the nearest neighbors of the desired product image that is under the threshold. Thus, the probability of finding the desired product is higher than in the exist situation. The rates of the fall out in the not exist and similar situations using IPM is higher than using OCR. That will lead to less accurate IPM than OCR. The fusion results for exist situation between the three algorithms are 8 % better than the higher result algorithm (IPM NNDR). The fusion results for the twice situation between the three algorithms are 4 % better than the higher result algorithm (IPM NN).

6 Conclusion and Future Work

The common problem in the related systems is portability such that many systems overload the user with equipment's or operations. Additionally, the wireless and database techniques may prevent the systems from being used in any grocery store.

According to the problem raised in the related systems, it is important to develop a system that does not require a lot of portable equipment and that minimizes the system operating missions for the user. Additionally, the system should operate in any grocery store without restrictions to a wireless connection or database to recognize products, but uses object recognition algorithms to recognize products. Therefore, in the paper we assumed that the system consists of a shopping cart with three cameras installed vertically on one side of the cart. The software system compared the visual features of the products using Interest Point matching (IPM), while text features can be recognized using Optical Character Recognition (OCR). The evaluated IPM algorithm is SURF descriptor. The SURF algorithm was evaluated using two different matching strategies. The OCR results were obtained using OCR engine called ABBYY. In order to evaluate the matching results, the confusion matrix and performance rates were used to measure the results. The test was conducted on dataset called ItemMaster [15] rather than taking real images from the grocery store. A fusion was made between the IPM and OCR results. The results showed that the IPM is more accurate than the OCR in case the user desired product is available on the shelf. While the OCR is more accurate in case the user desired product is not available on the shelf. This due to, the IPM ability to match the products images with the most matching product even if it was not the same product. While, the OCR matches the retrieved text from the OCR engine with the retrieved product name from the user. For that reason, OCR will not indicate a product match until all the text is matched under the specified threshold. In the future, we can develop an OCR algorithm that is specially created for detecting and recognizing the text written on grocery store products. Furthermore, we can use a database containing images for the products brand names logo.

7 References

[1] World Health Organization, "Visual impairment and blindness," August 2014. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/. [Accessed 28 September 2015].

[2] V. Kulyukin et al., "RoboCart: Toward Robot-Assisted Navigation of Grocery Stores by the Visually Impaired," in IEEE International Conference on Intelligent Robots and Systems (IROS), Alberta, 2005.

[3] J. Nicholson et al., "ShopTalk: Independent Blind Shopping Through Verbal Route Directions and Barcode Scans," The Open Rehabilitation Journal, vol. 2, 2009.

[4] P. Lanigan et al., "Trinetra: Assistive Technology for Grocery Shopping for the Blind," in 10th IEEE International Symposium on Wearable Computers, Montreux, 2006.

[5] P. Narasimhan, "Assistive Embedded Technologies," IEEE Computer Magazine, vol. 39, pp. 85-87, 2006.

[6] L. Carlson et al., "GroZi Shopping Assistant," California Institute for Telecommunications and Information Technology, San Diego, 2007.

[7] S. Krishna et al., "A Wearable Wireless RFID System for Accessible Shopping Environments," in 3th International Conference on Body Area Networks (BodyNets08), Arizona, 2008.

[8] D. Ipina et al., "BlindShopping: Enabling Accessible Shopping for Visually Impaired People through Mobile Technologies," in 9th International Conference on Smart Homes and Health Telematics, Montreal, 2011.

[9] A. Baumberg, "Reliable Feature Matching Across Widely Separated Views," in IEEE Conf. Computer Vision and Pattern Recognition, 2000.

[10] A. Goshtasby, 2D and 3D Image Registration for Medical, Remote Sensing, and Industrial Applications, New Jersey: John Willy & Sons, 2005.

[11] R. Szeliski, Computer Vision Algorithms and Applications, London, UK: Springer-Verlag, 2011.

[12] H. Bay, et al, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding (CVIU), vol. 110, no. 3, p. 346–359, 2008.

[13] K. Wang and S. Belongie, "Word Spotting in the Wild," in 11th European Conference on Computer Vision, Greece, 2010.

[14] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Transaction on Pattern Analysis and Matching Intelligence, vol. 27, no. 10, p. 1615–1630, 2005.

[15] "ItemMaster," 2012. [Online]. Available: http://www.itemmaster.com. [Accessed 31 December 2013].

Improving Word Recognition with a Time of Flight 3D Camera

Luis Galarza[1], **Harold Martin**[1], **Malek Adjouadi**[1]
[1] Department of Electrical and Computer Engineering, Florida International University, Miami, Florida, United States of America

Abstract - *This study introduces a modification to a previous method for dewarping book spread images in the design of an automated book reader for persons with visual impairment and blindness. This design concept could also be applied to the challenging book digitization process. In particular, we will focus on contrasting the effects of using uniform and full height maps, obtained through a time of flight device, when performing the necessary image corrections. The experiments conducted to prove the merits of this approach were performed on a testing dataset consisting of 142 pages with their corresponding depth maps that were extracted using the time of flight 3D camera. These 3D maps of book spreads are made available to other researchers as an open source resource for developing other dewarping mechanisms and character recognition algorithms. The results were quantified and measured by introducing the corrected images to an Optical Character Recognition (OCR) engine. Lastly, the robustness of the approach utilizing these height maps (uniform and full height) is also put to test by introducing unforeseen rotation on the book spreads which could happen when the book is not place properly*.

Keywords: Book reader, curvature correction, depth map, digitization of text, optical character recognition (OCR), time of flight (ToF)

1 Introduction

There is a great deal of information that is yet to be digitized, not only as part of a practical solution to the remarkably challenging book digitalization project, which benefits the general public at large, but also as means for universal access to reading materials for persons with visual impairment and blindness. For the proposed book reader design, although it could augment the book digitalization project, its main intent is for this system to be used as an assistive technology too for blind individuals in the comfort and privacy of their own homes or office and be able to use any bound book they wish to read. Flat documents in this case would not necessitate any corrections as they already yield very high reading accuracy with existing OCRs. Several designs of book readers have been proposed using different system set ups and architectures and different approaches at dewarping the image capture of bound book spreads. Our early attempts at such a design involved the use of a lateral camera to estimate the page curvature, and a top camera with a higher resolution to read the text once the dewarping is accomplished by flattening mathematically the curvature captured by the lateral camera [1, 2]. With the same intent to help persons with visual impairment and blindness, other related studies have looked into different design alternatives and methodologies. Interestingly, the work in [3] uses a single high resolution camera and dewarping is accomplished by taking into consideration cues present in the image by supposing baseline fitting assumed to be straight, horizontal and vertical vanishing point estimations on the perspective transformation on parallel lines assumed for the text. In this study, although they report a high reading accuracy of nearly 96%, the rectified text seemed skewed with some words that were difficult to read with the naked eye, while other words in a same paragraph appeared in italic next to others that looked rectified more correctly. Studies in [4, 5] are rather unique as they propose to convert e-books or e-documents available in PDF into formats that are easily accessible by persons with visual impairments (i.e., Braille, audiotape, or large print format). Understandably, descriptions of figures and graphs have to be added manually in a semiautomatic process [4]. In a sophisticated image analysis strategy, the study reported in [6] integrates the so-called iconic model (character-image classifier) and a linguistic model (word occurrence probability) in seeking a higher reading accuracy, with mutual entropy being the measurement that assess the disagreement between the two models (high mutual entropy to mean disagreement or that the answer given out by the iconic model has no corresponding entry in the dictionary). The hamming-distance-based template matching is used for classification in the iconic model relying on the character classifier. Other research groups have looked into automating reading and text tools for visually impaired persons [7] to be able to interact effectively and construct mathematical expressions focusing in this case on linear algebra as a good example to facilitate access to mathematics. The work in [8] focuses more on the removal of the so-called moiré-pattern noise and specular highlight present in document images acquired through a smartphone to benefit OCR reading outcome. The challenge of recognizing and then reading text extends of course to other languages [9] where the interesting case of Arabic text, which is cursive in its writing nature, and where characters assume different shapes as a function of their location in a word (start, within, and ending of a word) is explored. The authors in this study propose using 16 features on the basis of sliding windows and applying the hidden Markov model in its application in this case as a probabilistic pattern recognition process, bypassing altogether any segmentation process.

However, there is still room for improvement in these systems, especially in terms of recognition rate and reading accuracy. For instance, during image capture, warping effects occur due to curvature of the book. These aberrations make the character recognition process challenging and error prone. Methods using special dewarping or flattening mechanisms extract orientation information from cues implied directly from the printed text or from shading information to rebuilt the 3-D surface [3]. A thorough and interesting survey [10] highlights the merits and challenges of the different exiting systems and approaches used for reading documents.

Several studies have looked into the challenge of dewarping documents in order to improve text recognition and ultimately the reading accuracy by means of existing OCR. Interesting studies have been reported on means to rectify distorted text by either estimating the curled text lines [11, 12], looking into the geometric properties of the captured images [13], or using two images from a single camera at different angles to estimate document surface from corresponding points akin to stereo vision. Several studies proposed different flattening mechanisms through ether grid modeling [14], 3D shape modeling [15], by considering parallelism and equal line spacing [16], looking into slopes in words for realignment [17], or by using curved surface projection modeling and baseline estimation as a goal-oriented rectification [18]. The concepts of dewarping and flattening have also been explored for other applications that include the restoring and enhancement of historical documents [19, 20], and for recognizing street signs from 3D scenes [21]. Cylindrical surface modeling has also been considered through the use of an isometric mesh [22] or through optical flow and structure from motion by using the camera of a mobile phone which is swept across the book spread for image capture [23].

This study focuses on approaches that work based on the book's surface geometry which we obtained directly from depth extraction, similar to [24] - [26]. With notable advances in Time of Flight (ToF) sensors and with their increased range of applications [27] - [31]; it is possible to obtain accurate depth data and thereby 2-D height profile maps that reflect the book curvature. Therefore, improvements on character recognition from distortion correction algorithms relying on height maps can now be further explored. This paper will particularly focus in using two height maps from multiple book spreads, the uniform height map will be obtained by repeating a single surface curve in the book spread's surface, and the full height maps will be obtained from a full scan of the book's surface by a ToF sensor. Both of these maps will be utilized by different approaches and the reading accuracy will then be measured by introducing the resulting images into an OCR engine.

The approach used in this paper was chosen for its dewarping characteristics via lens correction from a surface map. The uniform map was initially developed by obtain the book's curvature information (height map) from a single side view (lateral) image, and assumed uniformity through the rest of the book's surface. However, ToF devices can be used instead to reduce the size of the book reader. The uniform map is then made possible by simply replicating the center curvature. Whereas, the full height map can be obtained from the raw depth data provided by the ToF device. This height information was then used to perform the desired corrections.

The following sections will go into more detail as to how the corrections are completed, not only using the algorithm's inherent height maps, but also their adjusted versions to allow for the use of both height profiles; the effects of which will be noted in the results. Furthermore, the resilience of the algorithms will be explored by rotating the book spreads to introduce misalignment.

2 Curvature Correction Method

The location of a given object relative to the camera can cause the so-called warping effect. The first step of this method involves correcting the barrel distortion introduced by the camera lens as addressed in [1], [2] and [31]. This distortion is created because the magnification of the lens is inversely proportional to the distance from the optical axis. This makes a straight line in the real world seem as a curved line in the image plane as it moves away from the optical axis. Fig. 1 provides a visual representation of how a distorted straight line is corrected, whereas equation (1) provides the mathematical expression describing the distorted radius (r_{dist}). In equation (1) P^C is a vector containing the inherent parameters of the camera lens and R_{cor}^T is the transpose of a vector containing the radius of each pixel in the image. The radius of the distorted pixel's location in terms of its x and y coordinates (x_{dist} and y_{dist}, respectively) is described by equation (2).

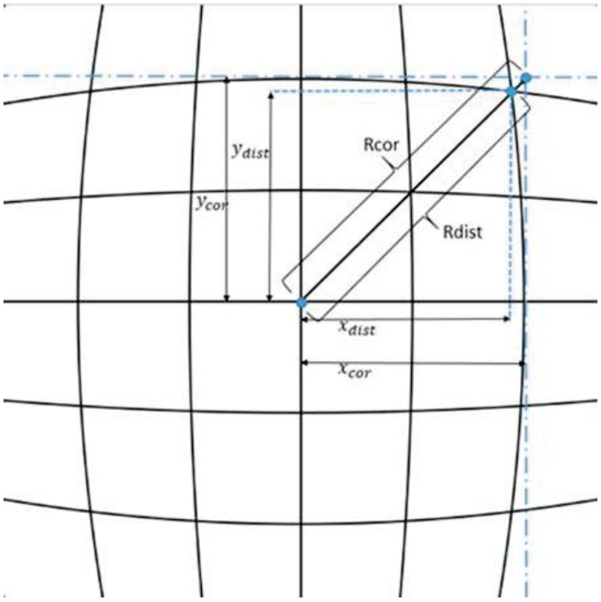


Fig. 1. Barrel Distorted Image and its Correction

$$r_{dist} = P^C . R^T_{cor} \tag{1}$$

$$r_{dist} = ({x_{dist}}^2 + {y_{dist}}^2)^{\frac{1}{2}} \tag{2}$$

In the implementation phase, an expansion of equation (1) as given by equation (3) is used. In this expression, term P^C_1 represents the first coefficient of the P^C vector, P^C_2 the second, and so on for the other terms. The r_{cor} terms are the elements of the R^T_{cor} vector.

$$r_{dist} = P^C_1 * {r_{cor}}^4 + P^C_2 * {r_{cor}}^3 + P^C_3 * {r_{cor}}^2 + P^C_4 * {r_{cor}}^1 \tag{3}$$

Thus, after the barrel correction is completed, a perspective transformation is applied to the image, referred to in this study as a "push operation". This operation corrects the perspective transformation, which causes straight lines to appear curved due to the camera's point of view as seen in Fig. 2.

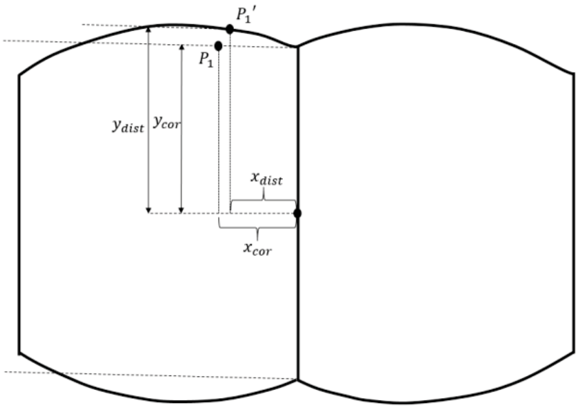


Fig. 2. Push Operation

The push operation as described by equation (4) is used to obtain a pixel's undistorted vertical distance from the camera's center axis (y_{cor}). In this equation, *f* is the camera focal length, $h(x_{dist}, y_{dist})$ is the height of the distorted pixel ($i.e., P_1'$ in Fig.) from the book stand surface in the real world, and y_{dist} is the distorted vertical distance of a given pixel from the camera's optical axis.

$$y_{cor} = (f . y_{dist})/(h(x_{dist}, y_{dist}) - f) \tag{4}$$

The last step for this method is an extension operation along the x-axis to eliminate the horizontal distortion that the book's curvature introduces in the image. This operation, originally introduced in [9] can be derived from Fig. to yield equation (5).

$$r = (\Delta x^2 + (h_2 - h_1)^2)^{\frac{1}{2}} \tag{5}$$

From equation (5), the flattened distance between two pixels, as represented by r, can be approximated by a triangle when the horizontal distance between said points (Δx) is sufficiently small, and where h_1 *and* h_2 represent the respective heights of the two points with respect to the book holder's surface.

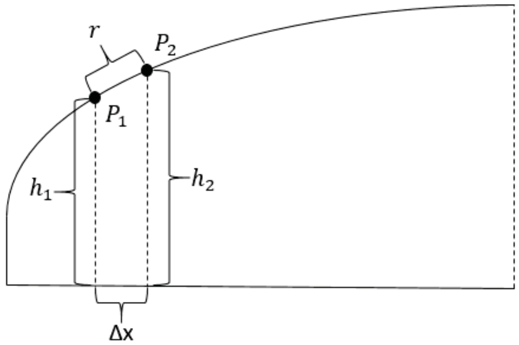


Fig. 3. Section View of the Page Curvature

This method originally obtained the pixels' height values by extracting the book curvature from a side picture of the book spread which is then uniformly distributed throughout the entire surface thereby generating what we refer to as a uniform height map. For this study, the height maps will be instead obtained using the experimental setup described in section 3. This setup will generate the uniform and full height maps which will then be used as the new values for h and h_i respectively, as detailed next.

3 Experimental Setup and Adjustments

We use the Bluetechnix Argos 3D-P100 depth sensor to capture the book's surface elevation information (or depth map). The information provided by this sensor is captured based on the principle of time of flight (ToF). This sensor by itself does not contain a high resolution RGB camera and therefore an external one needs to be use for obtaining the detailed image required for the OCR engine. Nonetheless, this sensor can produce a grayscale low resolution image (160x120 pixels) which was used when matching the height map to the high resolution image. The high resolution camera used in our setup is the Canon G6 which offers a resolution of 3 megapixels. The devices were setup as seen in Fig. 4.

With this setup in place, the full height map of the book's surface was obtained by subtracting the average depth of 50 frames of the book stand from the average depth of 50 frames of the book's surface. The uniform height was created by only using the center row of the full height map and replicating. Both these type of maps were obtained for each of the book spreads. As previously mentioned, due to the resolution discrepancy we then proceeded to register the high resolution image with the low resolution image obtained from the ToF device by using three known points as reference points (two points along the spine of the book and one on a corner). Once the matching was in place, we were able to scale the height map *(h)* to match that of the high resolution image. This was done for a chapter of a book comprised of 142 pages with their corresponding full height maps. The uniform height maps were then obtained by replicating the center row of the height map (*h*) on all the other rows. Lastly, the book was physically rotated by 30 degrees and 74 new rotated book spread images and matching height maps were once again obtained.

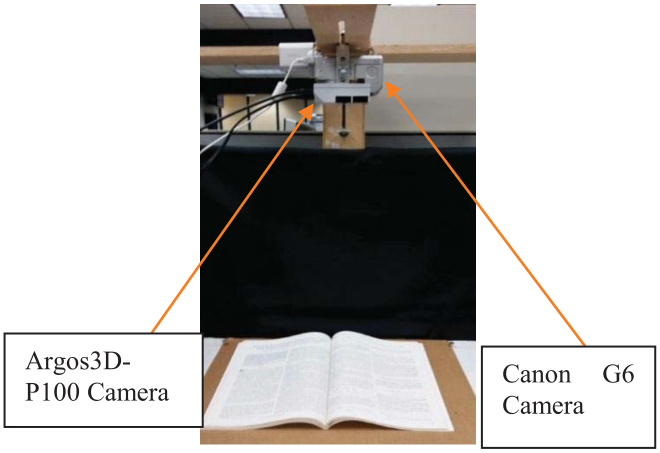


Fig. 4. Experimental Setup

Once the images were corrected they were then fed into the latest version of ABBYY FineReader, v12.0 OCR engine. The resulting digital file is then compared on a character by character basis with the actual digital version of the book's chapter. The character by character comparison was made possible by utilizing the Myers' Difference Algorithm which finds the longest common chain of matching characters in two sequences (control and test sample). Utilizing this algorithm allowed us to ease and expedite the comparison process, and was especially useful with the larger datasets used in this study.

4 Results

The following results demonstrate the curvature corrections that were made possible through the use of both uniform and full height maps on dewarping the pages of bound books. In addition, to further appreciate the impact of the depth maps, the book spreads were place in standard and rotated positions. The accuracy of the recognized characters is shown in the following sections for each of the 142 pages at an average of about 3245 characters per page.

4.1 Standard Placement

The standard placement is achieved by positioning the book spread so that it is aligned with the base of the book reader which yields book spreads as the one seen in Fig. 5.

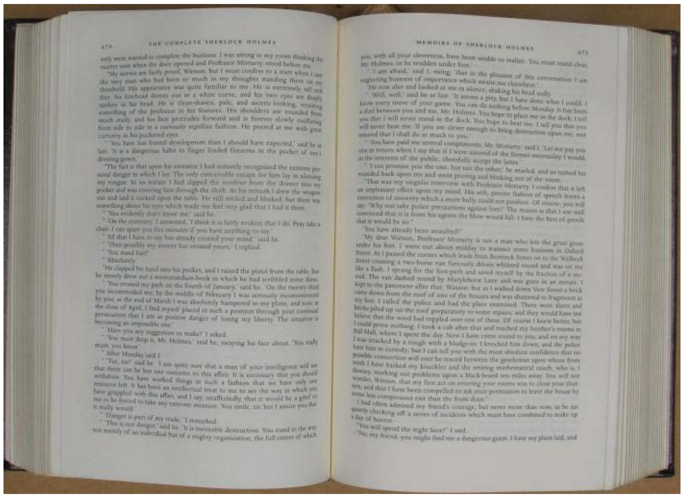
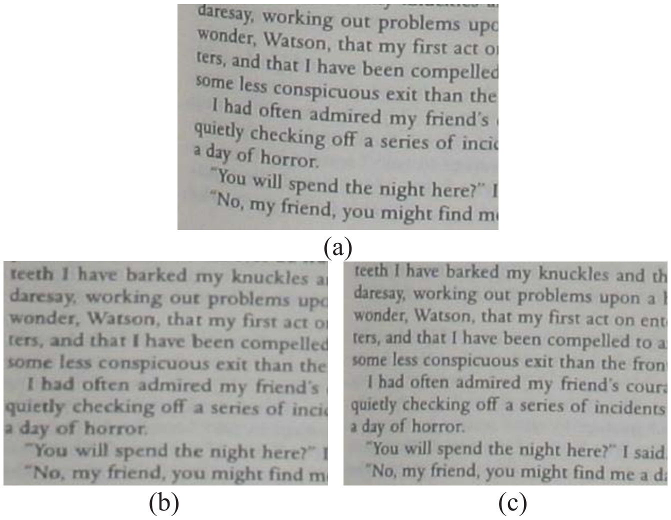


Fig. 5. Standard aligned book spread



daresay, working out problems upc
wonder, Watson, that my first act o
ters, and that I have been compelled
some less conspicuous exit than the
I had often admired my friend's
quietly checking off a series of inci
a day of horror.
"You will spend the night here?" I
"No, my friend, you might find m

(a)

teeth I have barked my knuckles a
daresay, working out problems upc
wonder, Watson, that my first act o
ters, and that I have been compelled
some less conspicuous exit than the
I had often admired my friend's
quietly checking off a series of inci
a day of horror.
"You will spend the night here?" I
"No, my friend, you might find m

(b)

teeth I have barked my knuckles and th
daresay, working out problems upon a l
wonder, Watson, that my first act on ent
ters, and that I have been compelled to a
some less conspicuous exit than the fron
I had often admired my friend's coura
quietly checking off a series of incidents
a day of horror.
"You will spend the night here?" I said.
"No, my friend, you might find me a d

(c)

Fig. 6. Examples of text: (a) original warped, (b) corrected with uniform map, and (c) corrected with full map

If we focus on a section of the book spread, it is possible to see the warping effect that occurs as demonstrated in Fig. 6(a). This arching effect occurs in the majority of the tested book spreads. Fig. 6(b) shows when the result of the usual correction of the warped text using the uniform height map, while Fig. 6(c) illustrates the correction with the full height map. The OCR results of the warped images can be seen in Fig. 7. It should be noted from the results that for this type of correction, the OCR engine performs quite well. The results of the corrections using both the uniform map and full map are also depicted in Fig. 7, both of which performed better than on the original image. Table 1 shows a statistical summary of the results for all the methods.

It is observed that all the different curvature correction procedures led to enhanced reading accuracy. Furthermore, using the full height map yielded better results than the uniformly distributed height map. This outcome was anticipated as the full height map reflects better the true nature of the page curvature, which may not necessarily be replicated by simply assuming a uniformly distributed curvature from single row of text or from a lateral view of the book as in [29]. Correction errors most certainly occur due the complex nature of book curvatures as we flip through the pages. A smarter expansion process (i.e. recognizing key points as references from where to start the expansion) could improve these results. Nonetheless, we still have to contend with the imperfections of the ToF technology in extracting depth information and the nonlinear nature of page curvatures.

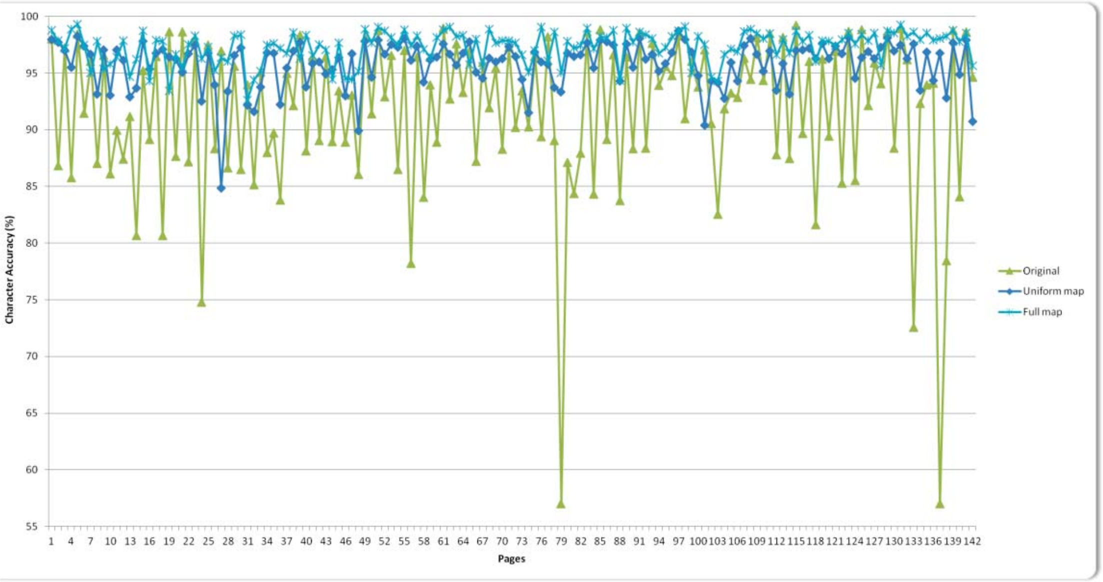


Fig. 7. Accuracy of the original and corrected book spreads

Table. 1 Summary of the standard positioning of book spreads

Procedure	Number of samples	Minimum accuracy	Maximum accuracy	Mean accuracy	Variance	Standard deviation	Median accuracy (after sort)
Original – no correction	142	56.99%	99.18%	91.57%	48.17	6.94	93.25%
Uniform map	142	84.82%	98.70%	95.80%	4.14	2.03	96.40%
Full map	142	92.59%	99.28%	97.33%	1.99	1.41	97.78%

.

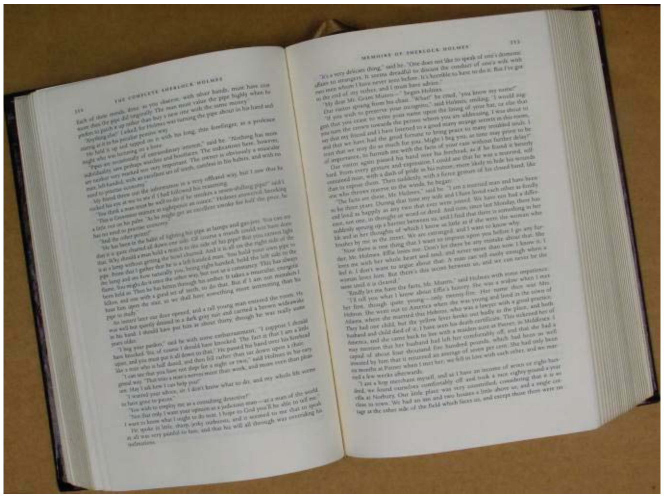
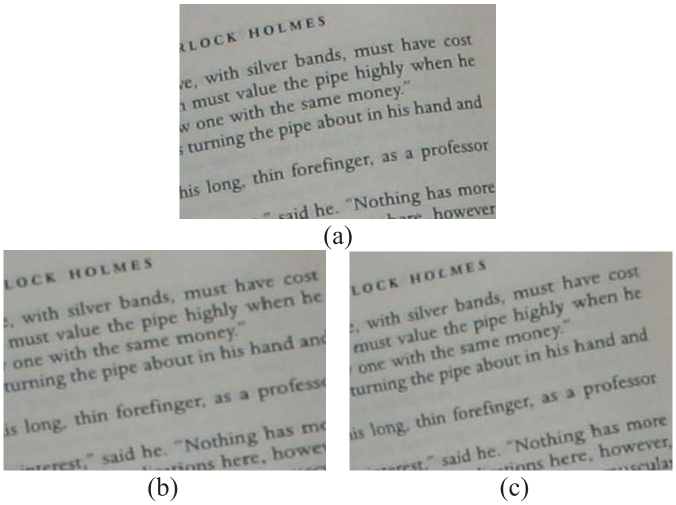


Fig. 8. Rotated book spread



RLOCK HOLMES
e, with silver bands, must have cost
n must value the pipe highly when he
w one with the same money."
s turning the pipe about in his hand and
his long, thin forefinger, as a professor
" said he. "Nothing has more

(a)

LOCK HOLMES
, with silver bands, must have cost
must value the pipe highly when he
one with the same money."
turning the pipe about in his hand and
is long, thin forefinger, as a professo
nterest," said he. "Nothing has mo

(b)

LOCK HOLMES
, with silver bands, must have cost
must value the pipe highly when he
one with the same money."
turning the pipe about in his hand and
is long, thin forefinger, as a professor
nterest," said he. "Nothing has more

(c)

Fig. 9. Examples of text: (a) original warped, (b) corrected with uniform map, and (c) corrected with full map

4.2 Rotated Placement

Many approaches rely on having a standard placement of the book spread, because of their sensitivity to misalignment in their standard placement. Therefore, to further test the resilience of the methods tested, as well as which type of height map has the best performance, the book spreads were physically rotated by 30 degrees. Then half of 142 pages were rotated 30 degrees left and the other half 30 degrees right, an example of which can be seen in Fig. 8.

The accuracy of the results for the rotated book spread images were quite high despite the rotation of the book spread. However, the overall accuracy was still lower than their standard placement counterparts. Fig. 9(a) illustrates a zoomed-in section of the warping effect where the text lines appear curbed instead of straight. Fig. 9(b) shows how the correction algorithm does straighten the lines so they are no longer warped while using uniform height map; while Fig. 9(c) depicts the correction with a full height map. Table 2 shows a statistical summary of all method with their corresponding maps. As expected, the method which had the highest overall accuracy and lowest variance was when the full height map. On the other hand, the uniform maps had a very poor performance, as they created a twisting effect in some cases that greatly decreased its performance even below the uncorrected instances, as observed in Fig. 8.

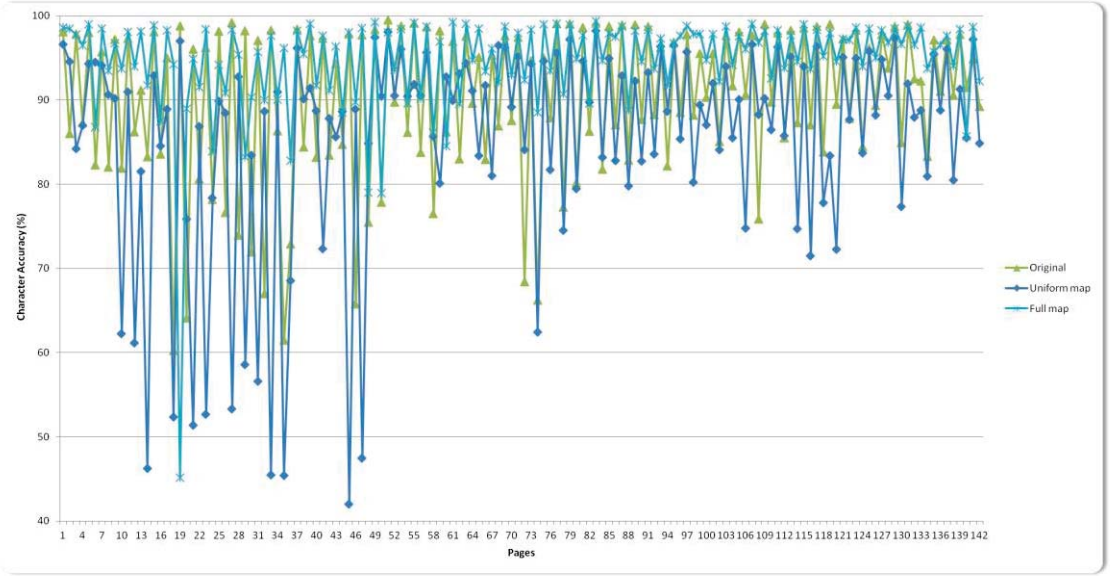


Fig. 10. Accuracy of the rotated original and corrected book spreads

Table. 2 Summary of the rotated positioned book spreads

Procedure	Number of samples	Minimum accuracy	Maximum accuracy	Mean accuracy	Variance	Standard deviation	Median accuracy (after sort)
Original – no correction	142	60.26%	99.51%	90.00%	84.82	9.21	95.74%
Uniform map	142	42.00%	98.20%	85.28%	160.95	12.69	88.90%
Full map	142	45.16%	99.38%	94.62%	36.06	6.01	96.55%

5 Conclusion

This study was able to show that performing dewarping corrections using full depth maps provides a better accuracy than just using a uniform depth map for either standard or rotated placement. Furthermore, the method proved to be more resilient to rotations of the book spread when using the full map rather than with the uniform map. Although the extension part of the algorithm introduces some noise, it seems to be providing better results in most cases. For future work, we intend to perform adjustments, such as picking better points for performing smarter expansions, perhaps by taking into account the orientation of the book spread. Other factors which could improve performance of the algorithms could be the matching of the high resolution image to the depth map, either by improving the resolution of the ToF device or the implementation of more accurate matching procedures that would reduce occlusions as well as other degrading conditions. As noted, a great deal of improvements is already present in the OCR engines such as ABBYY FineReader 12. However, there is still room for improvements, which can yield even more accurate text identification and digitization.

6 Acknowledgments

This research is made possible thanks to the support of the National Science Foundation through grants CNS-1532061, CNS-0959985, CNS-1042341, HRD-0833093, IIP 1338922. The support of the Ware Foundation is also greatly appreciated.

7 References

[1] M. Adjouadi, E. Ruiz, L. Wang, “Automated book reader for persons with blindness,” Computers Helping People with Special Needs. 4061 (2006) 1094-1101.

[2] L. Wang, M. Adjouadi, “Automated Book Reader Design for Persons with Blindness,” Lecture Notes in Computer Science, Springer Berlin Heidelberg, LNCS-5105 (2008) 318-325.

[3] P. Kakumanu, N. Bourbakis, J. Black, S. Panchanathan, "Document Image Dewarping Based on Line Estimation for Visually Impaired," 18th IEEE International Conference on Tools with Artificial Intelligence ICTAI '06. (2006) 625-631.

[4] E. Contini, B. Leporini, F. Paternò, “A Semi-automatic Support to Adapt E-Documents in an Accessible and Usable Format for Vision Impaired Users,” Computers Helping People with Special Needs. Springer Berlin Heidelberg. 5105 (2008) 242-249.

[5] A. Calabrò, E. Contini, B. Leporini, “Book4All: A tool to make an e-book more accessible to students with vision/visual-impairments,” HCI and Usability for e-Inclusion. Springer Berlin Heidelberg. 5889 (2009) 236-248.

[6] P.P. Xiu, H.S. Baird, “Whole-Book Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 34 (2012) 2467-2480.

[7] B. Almasri, I. Elkabani, R. Zantout, “An Interactive Workspace for Helping the Visually Impaired Learn Linear Algebra,” Computers Helping People with Special Needs. Springer International Publishing. 8547 (2014) 572-579.

[8] C. Simon, I.K. Park, “Correcting geometric and photometric distortion of document images on a smartphone,” Journal of Electronic Imaging. 24(1) (2015) 013038-013038.

[9] I. Ahmad, S.A. Mahmoud, G.A. Fink, “Open-vocabulary recognition of machine-printed Arabic text using hidden Markov models,” Pattern Recognition. 51 (2016) 97-111.

[10] J. Liang, D. Doermann, H. Li, “Camera-based analysis of text and documents: a survey,” International Journal of Document Analysis and Recognition (IJDAR). 7 (2-3) (2005) 84-104.

[11] A. Ulges, C.H. Lampert, T.M. Breuel, "Document image dewarping using robust estimation of curled text lines," Eighth International Conference on Document Analysis and Recognition. 2 (2005) 1001-1005.

[12] V. Kluzner, A. Tzadok, "Page Curling Correction for Scanned Books Using Local Distortion Information," International Conference on Document Analysis and Recognition (ICDAR). 18-21 (2011) 890-894.

[13] J. Liang, D. DeMenthon, D. Doermann, “Geometric rectification of camera-captured document images,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 30 (2008) 591-605.

[14] S. Lu, C. L. Tan, “Document flattening through grid modeling and regularization,” 18th International Conference on Pattern Recognition, ICPR 2006. 1, (2006) 971-974.

[15] C.L Tan, L. Zhang, Z. Zhang, T. Xia, “Restoring warped document images through 3D shape modeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 28 (2006) 195-208.

[16] J. Liang, D. DeMenthon, D. Doermann, “Geometric rectification of camera-captured document images,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 30 (2008) 591-605.

[17] L. Song, Y. Wu, B. Sun, “A Robust and Fast Dewarping Method of Document Images,” International Conference on E-Product E-Service and E-Entertainment. IEEE 2010. (2010) 1-4.

[18] N. Stamatopoulos, B. Gatos, I. Pratikakis, S.J. Perantonis, “Goal-Oriented Rectification of Camera-Based Document Images,” IEEE Transactions on Image Processing. 20 (2011) 910-920.

[19] L. Likforman-Sulem, J. Darbon, E. H. B. Smith, “Enhancement of historical printed document images by combining total variation regularization and non-local means filtering,” Image and Vision Computing. 29 (5) (2011) 351-363.

[20] K. Pal, M. Terras, T. Weyrich, “Interactive exploration and flattening of deformed historical documents,” Computer Graphics Forum. 32 (2pt3) (2013) 327-334.

[21] G. K. Myers, R. C. Bolles, Q. T. Luong, J. A. Herson, H. B. Aradhye, “Rectification and recognition of text in 3-d scenes,” International Journal of Document Analysis and Recognition (IJDAR). 7(2-3) (2005) 147-158.

[22] G.F. Meng, C.H. Pan, S.M. Xiang, J.Y. Duan, N.N. Zheng, “Metric Rectification of Curved Document Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 34 (2012) 707-722.

[23] C. Kim, P. Chiu, S. Chandra, “Dewarping Book Page Spreads Captured with a Mobile Phone Camera,” Camera-Based Document Analysis and Recognition. Springer International Publishing. 8357 (2014) 101-112.

[24] M.S. Brown, M.X. Sun, R.G. Yang, L. Yun, W.B. Seales, “Restoring 2D content from distorted documents,” IEEE Transactions on Pattern Analysis and Machine Intelligence. 29 (2007) 1904-1916.

[25] L. Galarza, Z. Wang, M. Adjouadi, "Book spread correction using a time of flight imaging sensor." In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV, WorldComp, (2014) 1.

[26] L. Galarza, Z. Wang, M. Adjouadi, "Book reader optimization using a time of flight imaging sensor." In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV, WorldComp, (2015) 324-330.

[27] Y. Zhang, C.L. Tan, “An improved physically-based method for geometric restoration of distorted document images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2008) 728-734.

[28] S. Foix, G. Alenyà, C. Torras, “Exploitation of time-of-flight (ToF) cameras,” Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Tech. Rep. IRI-TR-10-07 (2010).

[29] C. Kim, H. Yu, G. Yang, “Depth super resolution using bilateral filter,” 4th International Congress on Image and Signal Processing, CISP 2011. 2 (2011) 1067-1071.

[30] S. Schwarz, M. Sjostrom, R. Olsson, “A weighted optimization approach to time-of-flight sensor fusion,” IEEE Transactions on Image Processing. 23(1) (2014) 214-225.

[31] D. Jiménez, D. Pizarro, M. Mazo, S. Palazuelos, “Modeling and correction of multipath interference in time of flight cameras,” Image and Vision Computing. 32 (1) (2014) 1-13.

[32] L. Wang, “An Automated Book Reader Design as an Assistive Technology Tool for Persons with Blindness,” Ph.D. dissertation, Dept. Elect. and Comp. Eng., Florida International University, Miami, FL (2007).

MRI Denoising Using Randomized Version of Nonlocal Means Method

Jinrong Hu[1,2], Jia He[3], Ying Fu[3], Xi Wu[3] and Jiliu Zhou[3]
[1]School of Computer and Soft Engineering, Xihua University, Chengdu, 610039, China
[2]Key Laboratory of Pattern Recognition and Intelligent Information Processing, Chengdu University, Chengdu, 610106, China
[3]School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China

Abstract - *Non-local mean (NLM) algorithm has been implemented effectively in MRI denoising and is always limited by its computational complexity. To reduce the computational burden of NLM in 3D MRI dataset, in this paper, we used a randomized version of NLM algorithm to remove the noise in MRI data. The random NLM algorithm seeds up the classical NLM by computing a small subset of image patch distances, which are randomly select according to the uniform sampling pattern. Numerical experiments demonstrate that the random NLM can achieve a competitive denoising result at a low sampling rate (0.05) in 3D MRI dataset while reducing the runtime dramatically.*

Keywords: MRI, denoising, non-local means, uniform sampling pattern

1 Introduction

Magnetic Resonance Imaging (MRI) has been involved as one of the primary imaging tool which can provide highly detailed images of tissue and organs in the human body invasively and in vivo. However, MRI imaging is vulnerable to random noise and need effective denoising procedure for further clinical applications. Moreover, MRI suffers significantly from Rician noise which is mainly due to the echo planar imaging sequences and demands efficient denoising procedure to ensure the subsequent analysis[1].

There is a wide range of MRI denoising approaches have been proposed in literature to remedy this problem. In contrast to spatial coherence based image smoothing, Buades et al. proposed a non-local means (NLM) denoising algorithms which average pixels according to their intensity similarities[2]. NLM denoising method are effectively implemented in MRI denoising[3-6] and extended to multiple MRI modalities such as DWI [7, 8]. Despite the effective performance, the implementation of NLM denoising always limited by its high computational complexity. It is obvious to see that the computation of all the weight require operations, where are the number of pixels in image, number of patches and number of pixels in patch. For conquer the computational complexity of NLM method, many strategies have been proposed. The first strategy is to reduce the reference window size. Manjon et al. use a 17×17 searching window instead of searching the whole dataset in a heuristic manner[3], Talebi and Milanfar use uniform sampling to occupy the subset of large database and implement Nyström method to achieve approximation of whole dataset[9]. The second strategy is to reduce the dimension of patch in which SVD or PCA projection can be used to project the patches into lower dimensional space and yields less computations[10-12]. The last strategy is to optimize the patch structure. For instance, fast bilateral grid[13], fast Gaussian transform[14] and Gaussian KD tree[15] are effectively implemented in NLM denoising. However, these optimized strategies are limited by extra requirement which hinders the applications.

In this paper, for reduce the computational burden of NLM in 3D MRI dataset, a random NLM is used. For each voxel, we randomly select k reference voxels from the whole dataset with the uniform sampling pattern. Since k is considerably less than the whole dataset, the computational complexity will be significantly decreased. The experiments demonstrate that the random NLM can achieve a competitive denoising result at a low sampling rate (0.05) in 3D MRI dataset while reducing the runtime dramatically.

The rest of the paper is organized as follow. The randomized version of NLM algorithm are presented in section 2. Experimental results and computational efficiency are given in section 3, and concluding remarks are given in section 4.

2 THE RANDOMIZED VERSION OF NLM ALGORITHM

2.1 Method

The original NLM image denoising algorithm introduced by Buades et al. smoothes images according to a weighted mean defined by the intensity similarity in a pre-determined neighborhood[2]. Specifically, the filtered pixel using the NLM method is calculated as follow:

$$\hat{x}_i = \frac{\sum_{j=1}^{n} w_{i,j} y_j}{\sum_{j=1}^{n} w_{i,j}}, \quad (1)$$

where i is the filtered pixel and j represents pixels in the searching window. The weights $w_{i,j}$ are based on the similarity between the patches N_i and N_j commonly defined as a square neighborhood window around pixel i and j respectively. The similarity $w_{i,j}$ is calculated as:

$$w_{i,j} = \exp\left(-\left\|N_i - N_j\right\|_{\Lambda}^{2} \Big/ 2h_r^2\right), \quad (2)$$

where h_r is a scalar parameter determined by the noise level, and $\|\bullet\|_{\Lambda}^{2}$ is the weighted ℓ_2 -norm with a diagonal weight matrix Λ.

Despite its promising performance, NLM has always limited by its heavy computational complexity. It is easy to see that computing all weights $\left\{w_{i,j}\right\}$ requires $O(mnd)$ arithmetic operations, where m,n,d are, respectively, the number of pixels in the noisy image, the number of patches used, and the number of pixels in patch. Additionally, about $O(m,n)$ operations are needed to carry out the summations and multiplications in (1) for all pixels in the image. For MRI dataset, the computational burden increases dramatically since m and d increase hundred times for its three-dimensional structure for both dataset and patch size which prohibited the effective application of NLM denoising.

As mentioned above, computing all weights $\left\{w_{i,j}\right\}$ of the whole MRI dataset is computationally prohibitive when the size and dimension are large. To reduce the complexity, the basic idea of proposed random NLM (RNLM) is to randomly select a subset representation of $\left\{w_{i,j}\right\}$ to approximate the sums in the numerator and denominator in (1). For doing this, a sequence of independent random variables $\left\{I_j\right\}_{j=1}^{n}$ is applied to each pixel in the noisy image independently, and the probability of random variables is as follow:

$$\begin{cases} P\left[I_j = 1\right] = p_j \\ P\left[I_j = 0\right] = 1 - p_j \end{cases}. \quad (3)$$

The j-th weight $w_{i,j}$ is sampled if and only if $I_j = 1$ and refer the sampling pattern of RNLM to the vector $\mathbf{P} \overset{\text{def}}{=} \left[p_1, \cdots, p_n\right]^T$.

Given a set of random samples N_j from $\mathbf{y}$, the RNLM algorithm approximate the numerator and denominator in (1) by two random variables

$$\begin{cases} A_i(\mathbf{P}) \overset{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} \frac{y_j w_{i,j}}{p_j} I_j \\ B_i(\mathbf{P}) \overset{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} \frac{w_{i,j}}{p_j} I_j \end{cases}, \quad (4)$$

where the argument $\mathbf{P}$ emphasizes the fact that the distributions of A_i and B_i are determined by the sampling result of $\mathbf{P}$. The full NLM result $\hat{x}_i$ in (1) is then approximated by RNLM method as:

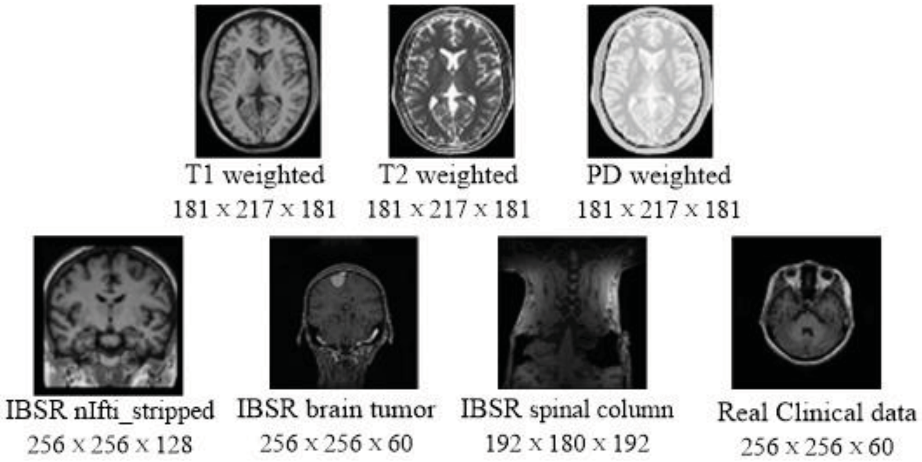
$$\hat{x}_i(\mathbf{P}) \overset{\text{def}}{=} \frac{A_i(\mathbf{P})}{B_i(\mathbf{P})} = \sum_{j=1}^{n} \frac{y_j w_{i,j}}{p_j} I_j \Big/ \sum_{j=1}^{n} \frac{w_{i,j}}{p_j} I_j. \quad (5)$$

Principally, $\hat{x}_i(\mathbf{S})$ is a biased estimate of the original value. However, as pointed out in [16], the probability of having a large deviation drops exponentially as $n \to \infty$, which ensures an accurate approximation of the original NLM when the proposed RNLM has a large n.

2.2 Performance Analysis

The primary concern of the proposed RNLM is whether the random sampled voxel could achieve good approximation of the full NLM method especially when the sampling rate is small. It is well known from the law of large numbers that the empirical mean of a large number of independent random variable stays very close to the true mean. A rigorous analysis of the approximation error using probabilistic large deviation theory[17] in scene image has been proposed in [16]. As pointed out in [16], the mean square root (MSE) achieved by random sampling with small sampling rate stays fairly close to the MSE achieved by the full NLM. The key of successful implements RNLM in 3D MRI dataset draws to the balance between the sampling rate (corresponding to the approximation accuracy) and computational complexity.

To analysis this, as demonstrated in Fig. 1, each of 10 MRI datasets (namely BrainWeb T1w, T2w, PDw, IBSR and 5 real MRI dataset scanned of different ROI: brain, spinal cord, chest, knee and ankle) are implemented using uniform sampling scheme with different sample rates.



T1 weighted
181 x 217 x 181

T2 weighted
181 x 217 x 181

PD weighted
181 x 217 x 181

IBSR nIfti_stripped
256 x 256 x 128

IBSR brain tumor
256 x 256 x 60

IBSR spinal column
192 x 180 x 192

Real Clinical data
256 x 256 x 60

Figure 1 Test 3D MRI datasets

As demonstrate in Fig. 2, the peak signal-to-noise ratio (PSNR) curves of the MRI dataset converge to limiting value as the sampling rate approach to 1. Specially, at the sampling rate of 0.005 and 0.1, the RNLM achieved more than 90% and nearly 100% results compared with full NLM result. Besides this, the proposed method achieve better and robust results in MRI dataset especially in low sample rate, this is probably because that the MRI datasets are all natural image with less singular point.

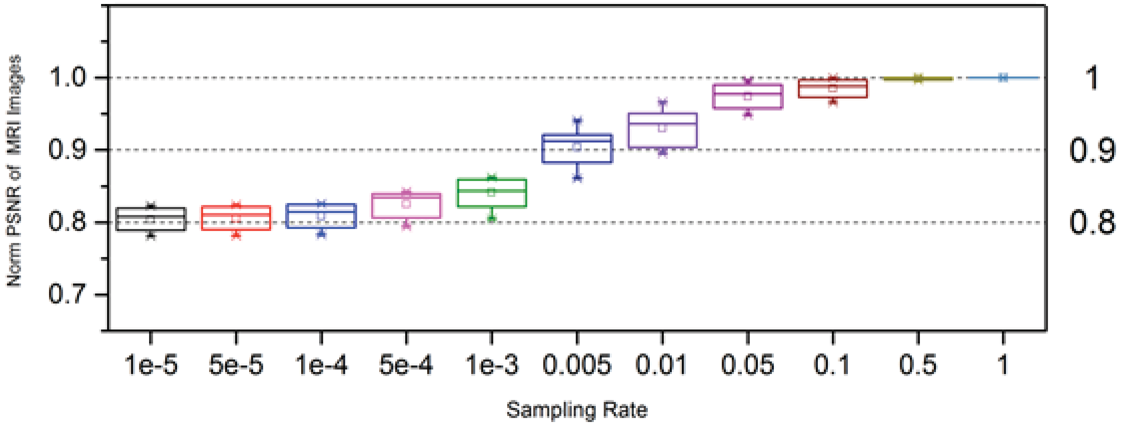


Figure 2 Normalized PSNR of MRI datasets by RNLM using uniform sampling

3 EXPERIMENTS

The random NLM (RNLM) algorithm using uniform sampling pattern with different sampling rate (0.0001, 0.005, 0.05, 0.1, 0.5, 1) and state-of-art method namely BM4D[18] were implemented for comparison. The parameters of RNLM are as follows: The search volume size is 25×25×25 (n = 15625) and the $h_s = (\lfloor \rho / 2 \rfloor)/3$, $\rho = 25$; the patch size is 3×3×3 (i.e. $d = 27$) and the weighted matrix $\Lambda = I/d$. The parameters of BM4D were set as indicated in [18].

For the BrainWeb, T1, T2 and PD weighted MRI 3D phantoms were used. All of them are 1×1×1mm pixel size and 181×217×181 pixels in size. To compare the performance of the proposed framework, the same three phantom images corrupted by Rician noise with different noise levels (3%, 6%, 9%, 12%, 15%, 18%, 21% of maximum intensity) were used, each of which was denoised with the denoising algorithms above. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity index SSIM [19] were selected for quantitative comparison and demonstrated in Figure 3 and the original records were shown in Table 1 for better discrimination. As can be seen from them, the proposed methods with high order sampling pattern can achieve more than 90% accuracy compared with full NLM. When the sampling rate increase to 0.05, the proposed RNLM can obtain almost the same result in both PSNR and SSIM.

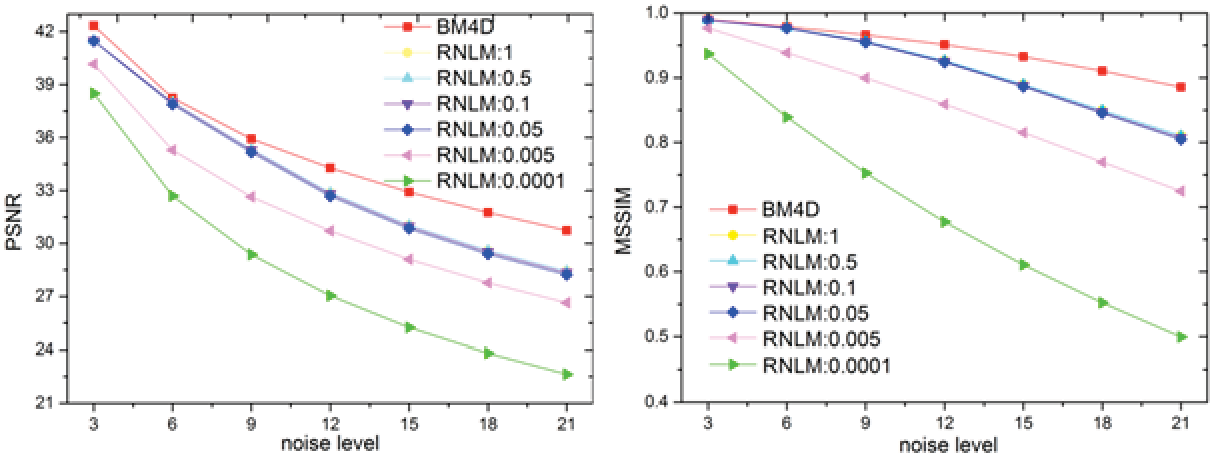


Figure 3 PSNR and MSSIM of T1 image using RNLM with different sampling rate

Table 1 PSNR and MSSIM of T1 images denoised by RNLM with different sampling rate and BM4D

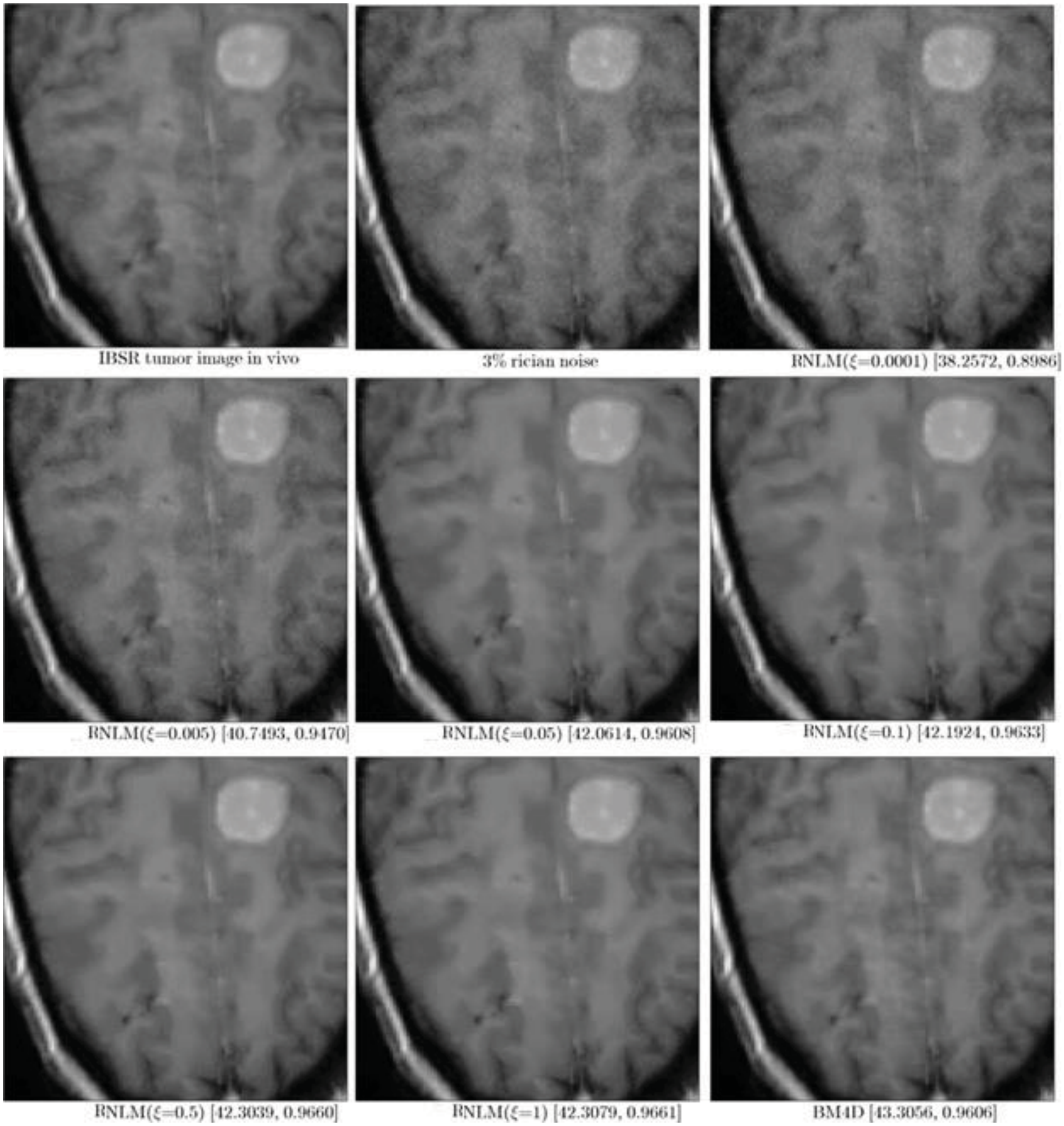
σ \ ξ	**0.0001**	**0.005**	**0.05**	**0.1**	**0.5**	**1**	**BM4D**
3	38.5033	40.1653	41.4717	41.492	41.496	41.496	42.3446
	0.9368	*0.9766*	*0.989*	*0.989*	*0.9889*	*0.9889*	*0.9897*
6	32.6959	35.2943	37.8883	37.9897	38.0125	38.0131	38.2428
	0.8391	*0.9384*	*0.9764*	*0.9772*	*0.9774*	*0.9774*	*0.9787*
9	29.3724	32.6477	35.1736	35.2961	35.3369	35.3382	35.9292
	0.7526	*0.8996*	*0.9544*	*0.9557*	*0.9562*	*0.9563*	*0.9661*
12	27.0393	30.7134	32.68	32.8153	32.8704	32.8719	34.2766
	0.6773	*0.859*	*0.9236*	*0.9252*	*0.9262*	*0.9262*	*0.9512*
15	25.2437	29.1018	30.8441	30.9681	31.0373	31.0397	32.9153
	0.6107	*0.8151*	*0.886*	*0.8878*	*0.8894*	*0.8894*	*0.9325*
18	23.8212	27.767	29.3944	29.5215	29.5932	29.595	31.7588
	0.552	*0.7694*	*0.8452*	*0.8473*	*0.8493*	*0.8493*	*0.9103*
21	22.6336	26.6429	28.2227	28.3469	28.4301	28.4328	30.7382
	0.4992	*0.7247*	*0.8046*	*0.8072*	*0.8099*	*0.81*	*0.8854*

For IBSR dataset, 3% Rician noise was added to achieve better discrimination of the performance of the above algorithms and the results can be seen in Fig. 4. In addition to visual comparisons, the comparison of computational efficiency is proposed in Table 2. It can be seen that, the results of proposed RNLM with a moderate sampling rate (0.05, 0.5, 0.1) are almost identical and become very close to the result of the full NLM solution. At low sampling rate (0.0001, 0.005), restored images via RNLM become blur and tend to lost some well-defined boundary information which is obvious in original image.

At last, Table 2 demonstrates the computational efficiency of the proposed methods with different sampling rate. It is obviously that the time of the proposed method is closely coinciding with the sampling rate. At the moderate sampling rate (0.05) which was figure out in both synthetic and in vivo datasets, the computational efficiency can be improved significantly.

Table 2 Runtime (in seconds) of RNLM with different sampling rate

Volume Size	Search Volume Size/3D Patch Size	**0.0001**	**0.005**	**0.05**	**0.1**	**0.5**	**1**
181×217×181	25×25×25/3×3×3	24.2	33.89	169.43	310.66	1460.3	3066.73


IBSR tumor image in vivo

3% rician noise

RNLM(ξ=0.0001) [38.2572, 0.8986]

RNLM(ξ=0.005) [40.7493, 0.9470]

RNLM(ξ=0.05) [42.0614, 0.9608]

RNLM(ξ=0.1) [42.1924, 0.9633]

RNLM(ξ=0.5) [42.3039, 0.9660]

RNLM(ξ=1) [42.3079, 0.9661]

BM4D [43.3056, 0.9606]

Figure 4 Denoised IBSR tumor image using RNLM with different sampling rate and BM4D

4 CONCLUSION

In this work, we implement random sampling in the NLM method to select a finite number of voxels for smoothing 3DMRI datasets. These selected voxels can reduce the computational complexity into a moderate time and ensure the generality of the NLM methods. The experiments demonstrate that the proposed method can achieve competitive results with other state-of-the art methods and can be used to improve the other NLM based algorithm.

5 Acknowledgments

This work is supported by National Basic Research Program of China (973 Program, No. 2014CB360506), National Science Foundation of China (No. 61303126), Ministry of Education, China: Chunhui Project (No. Z2015108), Sichuan Province Science and Technology Support Program (No. 2016RZ0051) and the Open Research Fund from Province Key Laboratory of Xihua University (No. szjj2013-022).

6 REFERENCES

[1] Sijbers, J. and A. Den Dekker, Maximum likelihood estimation of signal amplitude and noise variance from MR data. Magnetic Resonance in Medicine, 2004. 51(3): p. 586-594.

[2] Buades, A., B. Coll, and J.-M. Morel, A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation, 2005. 4(2): p. 490-530.

[3] Manjón, J.V., et al., MRI denoising using non-local means. Medical image analysis, 2008. 12(4): p. 514-523.

[4] Manjón, J.V., et al., New methods for MRI denoising based on sparseness and self-similarity. Medical image analysis, 2012. 16(1): p. 18-27.

[5] Manjón, J.V., P. Coupé, and A. Buades, MRI noise estimation and denoising using non-local PCA. Medical image analysis, 2015. 22(1): p. 35-47.

[6] Wu, X., et al., Nonlocal mean image denoising using anisotropic structure tensor. Advances in Optical Technologies, 2013. 2013: p. 1-6.

[7] Wiest-Daesslé, N., et al., Non-local means variants for denoising of diffusion-weighted and diffusion tensor MRI, in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007. 2007, Springer. p. 344-351.

[8] Coupé, P., et al., Collaborative patch-based super-resolution for diffusion-weighted images. NeuroImage, 2013. 83: p. 245-261.

[9] Talebi, H. and P. Milanfar, Global image denoising. Image Processing, IEEE Transactions on, 2014. 23(2): p. 755-768.

[10] Orchard, J., M. Ebrahimi, and A. Wong. Efficient nonlocal-means denoising using the SVD. in Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. 2008. IEEE.

[11] Tasdizen, T., Principal neighborhood dictionaries for nonlocal means image denoising. Image Processing, IEEE Transactions on, 2009. 18(12): p. 2649-2660.

[12] Van De Ville, D. and M. Kocher, Nonlocal means with dimensionality reduction and SURE-based parameter selection. Image Processing, IEEE Transactions on, 2011. 20(9): p. 2683-2690.

[13] Paris, S. and F. Durand, A fast approximation of the bilateral filter using a signal processing approach. International Journal of Computer Vision, 2009. 81(1): p. 24-52.

[14] Yang, C., et al. Improved fast gauss transform and efficient kernel density estimation. in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. 2003. IEEE.

[15] Adams, A., et al. Gaussian kd-trees for fast high-dimensional filtering. in ACM Transactions on Graphics (TOG). 2009. ACM.

[16] Chan, S.H., T. Zickler, and Y.M. Lu, Monte Carlo non-local means: Random sampling for large-scale image filtering. Image Processing, IEEE Transactions on, 2014. 23(8): p. 3711-3725.

[17] Dembo, A. and O. Zeitouni, Large deviations techniques and applications. 2009, Berlin: Springer.

[18] Maggioni, M., et al., Nonlocal transform-domain filter for volumetric data denoising and reconstruction. Image Processing, IEEE Transactions on, 2013. 22(1): p. 119-133.

[19] Wang, Z., et al., Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on, 2004. 13(4): p. 600-612.

Extreme-Level Eliminating Brightness Preserving Bi-Histogram Equalization Technique for Brain Ischemic Detection

V. Teh[1], K. S. Sim[1], and E. K. Wong[1]
[1]Faculty of Engineering and Technology, Multimedia University, Malacca, Malaysia

Abstract – *Stroke is a brain disease which can affect the blood supply to the brain or within the brain. Stroke is also one of the leading causes of death globally. Computed tomography (CT) scan and magnetic resonance imaging (MRI) are the main modalities used by radiologists and doctors for examination of stroke cases. Contrast and window setting plays an important role for visualization of CT brain image. However, the standard window setting and existing contrast enhancement technique may not be able to highlight the hypodense area. A new technique called extreme-level eliminating brightness preserving bi-histogram equalization (ELEBBHE) is presented in this paper to enhance the contrast on CT brain images. This new proposed technique is capable to enhance the whole image and increase the intensity of hypodense area. This new proposed technique is compared with existing techniques derived from histogram equalization (HE) and the results show that the new technique outperforms the rest of techniques.*

Keywords: contrast enhancement, CT scan, histogram equalization, extreme-level eliminating

1 Introduction

Stroke can be defined as a brain disease which happens when there is leakage within the brain arteries or blockage on the arteries to the brain [1]. Based on the American Heart Association, stroke is actually one of the leading causes of death around the world. In this modern world, computed tomography (CT) scan or computerized axial tomography (CAT) scan and magnetic resonance imaging (MRI) are among the most common imaging modalities used for brain visualization and diagnosis of stroke cases. In addition, these two imaging modalities have been widely used by radiologists and doctors around the world. However, CT scan are much frequently used primarily due to rapid process, cheaper and easily accessible [2].

The brain images obtained from CT scan are normally stored in Digital Imaging and Communication in Medicine (DICOM) format which form as 16-bits images. Out of this 16-bits DICOM images, 4-bits are used to stored textual data and the rest 12-bits are used to store the main image [3]. Examination and diagnosis of stroke cases are proceeded by radiologists and doctors through the presentation of DICOM images. In order to visualize the brain images obtained, a well-known technique as window setting which consisting of window center and window width is always used to stretch the range of CT numbers in Hounsfield Unit (HU) on 16-bits DICOM image to the standard grayscale range. The standard window setting consists of window center of 40 HU and window width of 80 HU is widely applied in most of the hospitals [4]. However, this standard window setting may not be able to provide a good contrast in CT brain images. Besides that, it may not be appropriate to highlight the hypodense area.

Contrast can be defined as the intensity differences which make the difference between the main object and the background. Thus, contrast in CT scan is important as to differentiate between the hypodense area (main object) and the normal brain soft tissue (background). Early detection of stroke is important to safe a person from suffering permanent damage or death. High contrast in CT brain image is most favorable as it eases the diagnosis process of ischemic stroke and increases the accuracy on detection of ischemic stroke. On the other hand, low contrast in CT brain image may cause the diagnosis process to be harder and leads to wrong justification by doctor. Thus, contrast enhancement technique as one of the image processing techniques is always used to enhance the CT brain image.

There have been a lots of contrast enhancement techniques being proposed and used to improve the contrast of an image such as histogram equalization (HE) technique, brightness preserving bi-histogram equalization (BBHE) technique, dualistic sub-image histogram equalization (DSIHE) technique, recursive mean separate histogram equalization (RMSHE) technique, recursive sub-image histogram equalization (RSIHE) technique and extreme-level eliminating histogram equalization (ELEHE) technique [5,6,7,8,9,10]. However, these do not perform well in enhancing CT brain images. Although ELEHE technique is specially developed and proposed for enhancing CT brain images, the intensity of normal brain soft tissue in enhanced CT brain image tends to go darker which is undesired in the process for diagnosis of ischemic stroke.

HE is considered as the simplest and rapid way of enhancing the overall contrast of an image, but it tends to affect the mean brightness of input image [5]. In this paper, a new technique is formulated and presented for enhancing the contrast of CT brain image. In the following section, some of the existing techniques mentioned earlier and the new proposed technique are briefly discussed.

2 Histogram Equalization

Histogram equalization (HE) technique is done by stretching the dynamic range of an input image [5]. In another words, HE technique reallocates the intensity values with a set of new values obtained from a non-linearity transfer function. An input grayscale image is normally made up of a set of matrix with lowest intensity value of 0 and highest intensity value of 255. Let assumed an input grayscale image with intensity level of R, and the probability density function (p.d.f) can be calculated as Eq. (1).

$$p.d.f(R_l) = \frac{n_l}{n}, \qquad (1)$$

where $l = 0, 1, \dots, L-1, n_l$ represents the number of pixels at intensity level R_l and n is the maximum number of pixels in the input image.

The cumulative density function (c.d.f) of intensity level R_l is basically equal to the summation of probability density function (p.d.f) as shown in Eq. (2).

$$c.d.f(R_l) = \sum_{l=0}^{l} p.d.f(R_l) \qquad (2)$$

After c.d.f is obtained, the transfer function of HE can be computed by Eq. (3).

$$TF = (R_{L-1} - R_0)\big(c.d.f(R_l)\big) + R_0, \qquad (3)$$

where R_0 is the lowest intensity value and R_{L-1} represents the highest intensity value of the input image.

Since the p.d.f of the input image has been normalized to the range of 0 to 1, it results the value of R_0 to be 0 and R_{L-1} to be 1. Hence, the transfer function of HE in Eq. (3) can be summarized as shown in Eq. (4).

$$TF = c.d.f(R_l) \qquad (4)$$

However, HE technique may cause over enhancement in some area, especially on the area with higher number of pixel values. Furthermore, while applying HE technique to CT brain image, it will definitely over enhance the background. This is due to the background in CT brain image occupied the most number of pixel values across the image. Therefore, various contrast enhancement techniques have been proposed to overcome the drawbacks of HE technique

3 Brightness Preserving Bi-Histogram Equalization and Dualistic Sub-Image Histogram Equalization

So as to maintain the mean brightness of the input image, Kim, (1997) proposed brightness preserving bi-histogram equalization (BBHE) technique [6]. Instead of applying the transfer function directly to the input image histogram, BBHE is done by dividing the input image histogram into two sub-image histogram based on a threshold value. In BBHE technique, the threshold is represented by the input mean intensity value, R_μ. The first sub-image histogram is within the grayscale range of 0 to R_μ, while the second sub-image histogram is within the range of $R_\mu + 1$ to 255. Transfer function of HE technique is then applied separately on each sub-image histogram. After the process of equalization, the image is then combined to form the enhanced image.

Next, Wang, (1999) proposed another type of bi-histogram equalization technique which is called as dualistic sub-image histogram equalization (DSIHE) technique [7]. This DSIHE technique has almost the same property as BBHE technique by dividing the input image histogram based on a threshold value. The only different is that DSIHE divides the input image at c.d.f of 0.50 with the purpose of maximize the output entropy. Besides that, DSIHE technique performs better while there are large regions with almost the same intensity value across an input image. Figure 1 shows the histogram of BBHE and DSIHE technique.

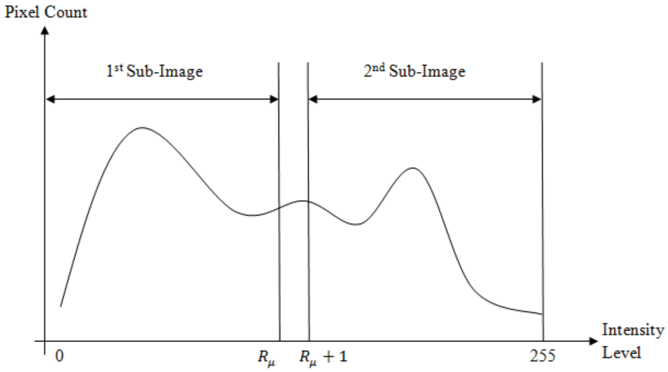


Figure 1: Sub-image histogram in the process of brightness preserving bi-histogram equalization and dualistic sub-image histogram equalization technique

Nevertheless, while applying on CT brain image, BBHE technique tends to cause the background of CT brain image to be brighter due to the threshold that based on input mean intensity value. DSIHE technique also tends to intensify the intensity of normal brain soft tissue closed to the intensity of hypodense area. This may definitely increase the chances of wrong diagnosis from radiologists or doctors who are lack of experience in diagnosis of ischemic stroke.

4 Recursive Mean Separate Histogram Equalization and Recursive Sub-Image Histogram Equalization

In year 2004, Chen and Ramli proposed recursive mean separate histogram equalization (RMSHE) technique to give a better enhancement with scalable brightness preservation [8]. In addition, RMSHE has almost the similar concept as BBHE technique. Instead of performing the separation of input image histogram once based on its input mean intensity value, RMSHE performs the separation in a recursive manner. It claims that the mean brightness of the enhanced images will converge to the mean brightness of the input image as higher recursion level is applied.

On the other hand, recursive sub-image histogram equalization (RSIHE) technique is also proposed. RSIHE is developed based on the concept of DSIHE technique by separating the input image histogram at c.d.f of 0.50, but the only different is that it is separate in a recursive manner [9]. RSIHE is able to give better energy preservation, contrast improvement and enhanced image with higher peak signal to noise ratio (PSNR) value. However, RMSHE and RSIHE involved higher complexity algorithm and longer computation time as the input image histogram need to be separated into 4 sub-image histogram before the transfer function of HE is implemented. Histogram of RMSHE and RSIHE technique are shown in Figure 2.

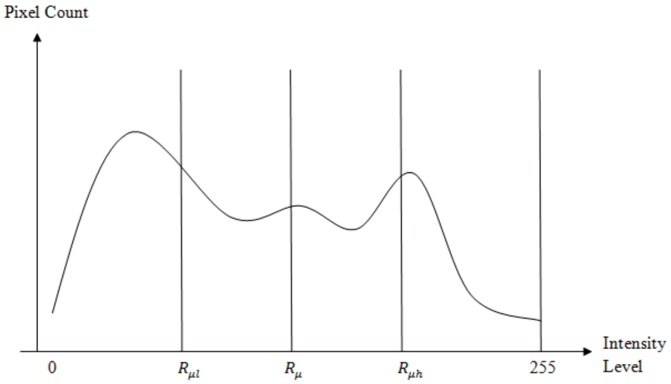


Figure 2: Sub-image histogram in the process of recursive mean separate histogram equalization and recursive sub-image histogram equalization technique

RMSHE and RSIHE technique are not suitable for enhancing CT brain image because there are huge differences of intensity value on the lowest and highest intensity value across the CT brain image. Moreover, these may cause the enhanced CT brain image to be too dark. Thus, the lesion or the hypodense area could be hardly seen.

5 Extreme-Level Eliminating Histogram Equalization

One of the global histogram equalization technique mainly proposed for enhancing CT brain image is extreme-level eliminating histogram equalization (ELEHE) technique. Tan et al. (2012) suggested that the enhancement of the two extreme-levels consist of the skull and the background in CT brain image is not needed as there are already at the boundary of grayscale range [10]. Hence, ELEHE technique is done by preserving the two extreme-levels and stretching the other grey levels as much as possible to the dynamic range.

In terms of algorithms, p.d.f of highest intensity value and lowest intensity value in the grayscale range is set to 0 before c.d.f is computed. Thus, the c.d.f can be obtained from Eq. (5).

$$c.d.f(R_l) = \sum_{l=1}^{l-1} p.d.f(R_l), \qquad (5)$$

where p.d.f of highest intensity value, R_{L-1} and the lowest intensity value, R_0 are set to 0 due to the elimination of two extreme-levels.

However, ELEHE technique may also intensify the normal brain soft tissue. This is undesired in the diagnosis process of ischemic stroke as it will be difficult to interpret between the normal brain soft tissue and the suspected hypodense area in ELEHE enhanced CT brain image.

6 New proposed technique:

Extreme-Level Eliminating Brightness Preserving Bi-Histogram Equalization

In this part, a new contrast enhancement technique for CT brain image is formulated from the existing brightness preserving bi-histogram equalization (BHE) and extreme-level eliminating histogram equalization (ELEHE) technique. The reason is that BBHE technique is able to preserve the mean brightness of input image and provides a good result in enhancing grayscale image, while ELEHE technique is proposed mainly for enhancing CT brain image. This new contrast enhancement technique, extreme-level eliminating brightness preserving bi-histogram equalization (ELEBBHE) is proposed to aid radiologists and doctors on diagnosis of ischemic stroke and reduce the error on detection. The details of ELEBBHE technique are discussed in the following sub-section.

6.1 Formulation of ELEBBHE technique

Given an input image R with l level of grayscale range and the input mean intensity value is R_μ. The input image histogram is then separated into two parts R_{low} and R_{high} as in Eq. (6).

$$R = R_{low} \cup R_{high}, \qquad (6)$$

where $R_{low} \in \{R_0, R_1, \dots, R_\mu\}$ and
$R_{high} \in \{R_{\mu+1}, R_{\mu+2}, \dots, R_{L-1}\}$.

p.d.f and c.d.f are then computed from each of the sub-image histogram.

The p.d.f can be calculated as shown in Eq. (7).

$$p.d.f(R_l) = \frac{n_l}{n}, \qquad (7)$$

where $l = 0, 1, \dots, L-1, n_l$ represents the number of pixels at intensity level R_l and n is the maximum number of pixels in the input image.

Since this new proposed technique is proposed mainly for enhancing CT brain image, the two extreme-levels (skull and the background) need to be eliminated to give a better enhancement. The main reason is that most of the existing contrast enhancement technique is not capable to give a good result while enhancing image with sudden changes among the intensity value. Thus, p.d.f of ELEBBHE on the highest intensity value R_{L-1} and the lowest intensity value R_0 are set to 0 as shown in Eq. (8).

$$p.d.f(R_0) = 0 \; and \; p.d.f(R_{L-1}) = 0 \qquad (8)$$

The c.d.f is calculated from the two sub-image histogram as shown in Eq. (9) and Eq. (10).

$$c.d.f(R_{low}) = \sum_{l=1}^{\mu} p.d.f(R_l) \qquad (9)$$

$$c.d.f\left(R_{high}\right) = \sum_{l=\mu+1}^{L-2} p.d.f(R_l) \qquad (10)$$

In this case, the calculation of c.d.f for lowest and highest grey value can be omitted since it has already been eliminated in Eq. (8).

The transfer function of lower sub-image histogram and higher sub-image histogram can be summarized in Eq. (11) and Eq. (12).

$$TF_{low} = \left[R_\mu - R_1\right] \times c.d.f(R_{low}) + R_1 \qquad (11)$$

$$TF_{high} = \left[R_{L-2} - R_{\mu+1}\right] \times c.d.f\left(R_{high}\right) + R_{\mu+1} \qquad (12)$$

Finally, the output image is given in Eq. (13).

$$\begin{aligned} E &= \{R(x,y)\} \qquad (13) \\ &= TF_{low}(R_{low}) \cup TF_{high}(R_{high}) \end{aligned}$$

Figure 3 shows a simple flowchart for the new proposed technique

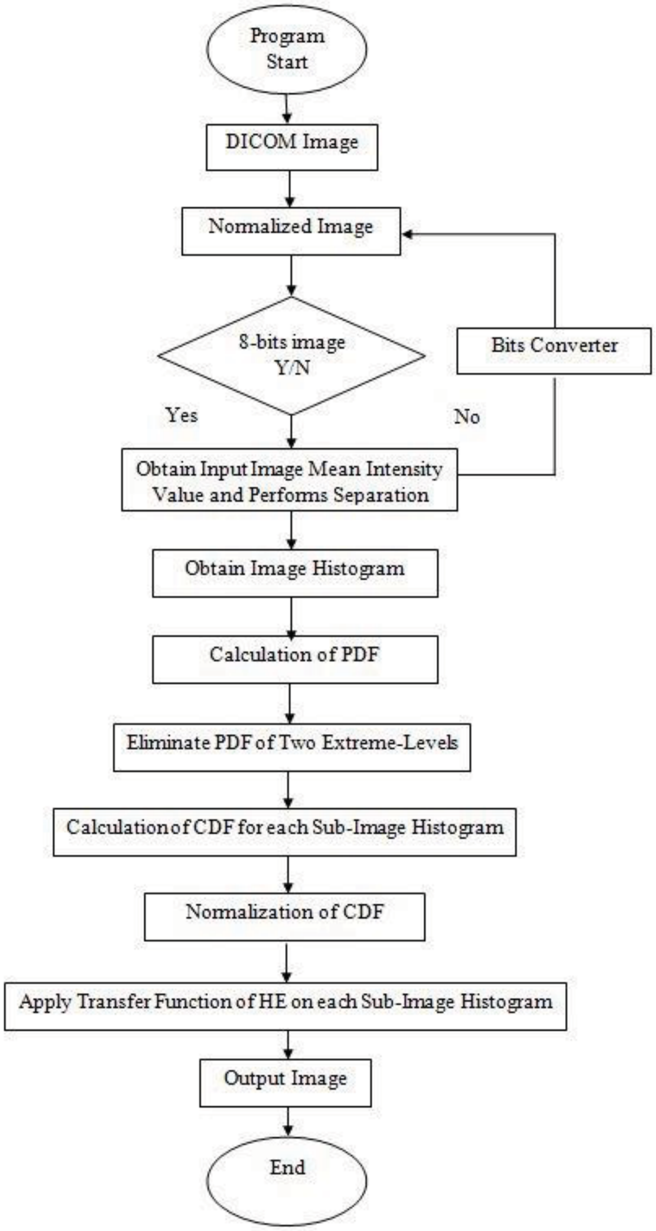
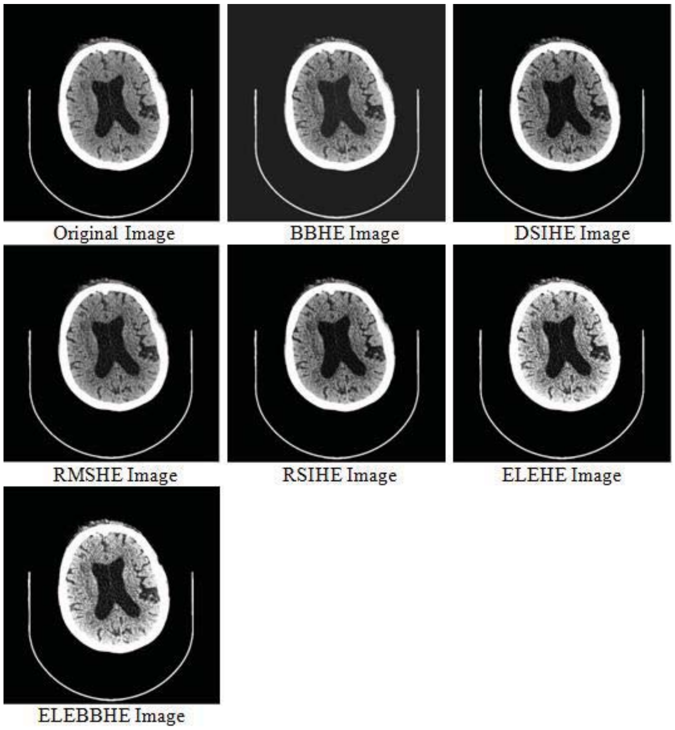

Program Start
DICOM Image
Normalized Image
8-bits image Y/N
Bits Converter
Yes
No
Obtain Input Image Mean Intensity Value and Performs Separation
Obtain Image Histogram
Calculation of PDF
Eliminate PDF of Two Extreme-Levels
Calculation of CDF for each Sub-Image Histogram
Normalization of CDF
Apply Transfer Function of HE on each Sub-Image Histogram
Output Image
End

Figure 3: Algorithm flowchart for extreme-level eliminating brightness preserving bi-histogram equalization technique

7 Results and Discussions

To check the performance of the new proposed contrast enhancement extreme-level eliminating brightness preserving bi-histogram equalization (ELEBBHE) technique, numbers of CT brain images with ischemic stroke cases are tested. All the CT brain images are converted from 16-bits DICOM images into 8-bits of grayscale images before the contrast enhancement are implemented. Sizes of the CT brain images are all initially set into the dimension of 512×512. The ELEBBHE technique, BBHE technique, DSIHE technique, RMSHE technique, RSIHE technique and ELEHE technique are used to enhance the CT brain images. From 500 CT brain images, three sets of CT brain images with various size of hypodense area are shown in Figure 4, Figure 6 and Figure 8 with respective image histograms in Figure 5, Figure 7 and Figure 9.



Original Image BBHE Image DSIHE Image

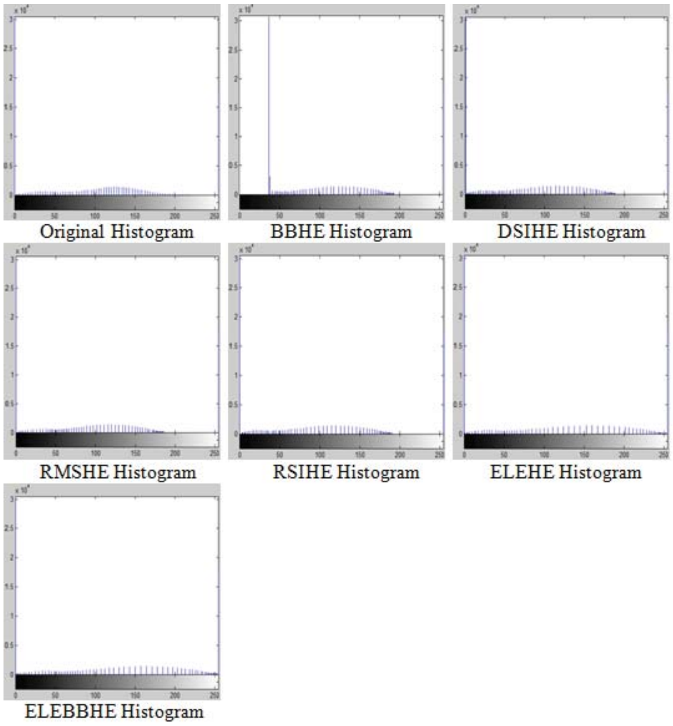
RMSHE Image RSIHE Image ELEHE Image

ELEBBHE Image

Figure 4: Images of various contrast enhancement techniques applied on CT brain image set I

It is very difficult for radiologists and doctors to diagnose on ischemic stroke case from low contrast CT brain image. Hence, contrast enhancement technique as one of the image processing tools is used to improve the contrast in CT brain images. Based on Figure 4 to Figure 9, it can be observed that the background of BBHE enhanced image tends to be brighter and the whole images are slightly washed out.

Besides that, DSIHE, RSIHE and ELEHE enhanced images also tends to intensify the normal brain soft tissue and cause the intensity of normal brain tissue closed to the intensity of hypodense area. These will definitely harden the diagnosis process of ischemic stroke.

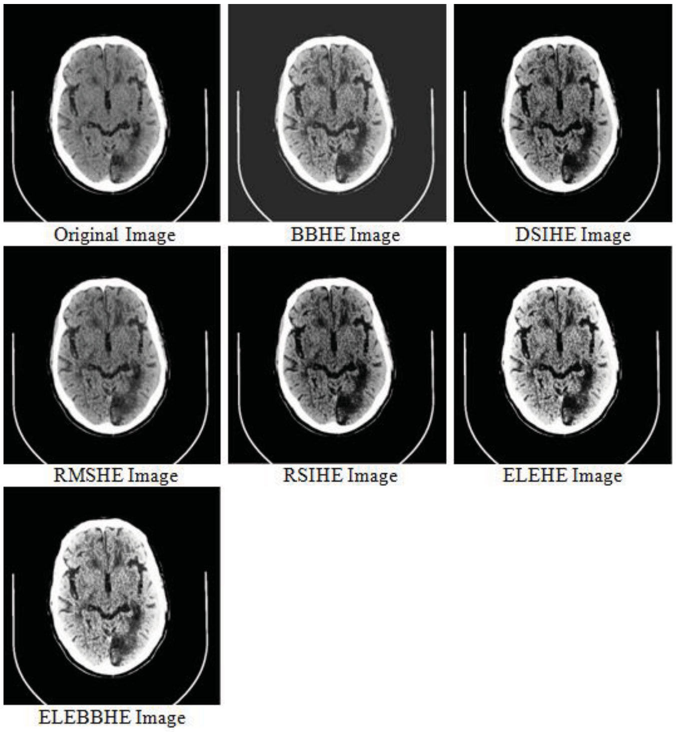


Original Histogram BBHE Histogram DSIHE Histogram

RMSHE Histogram RSIHE Histogram ELEHE Histogram

ELEBBHE Histogram

Figure 5: Image histograms of various contrast enhancement techniques applied on CT brain image set I

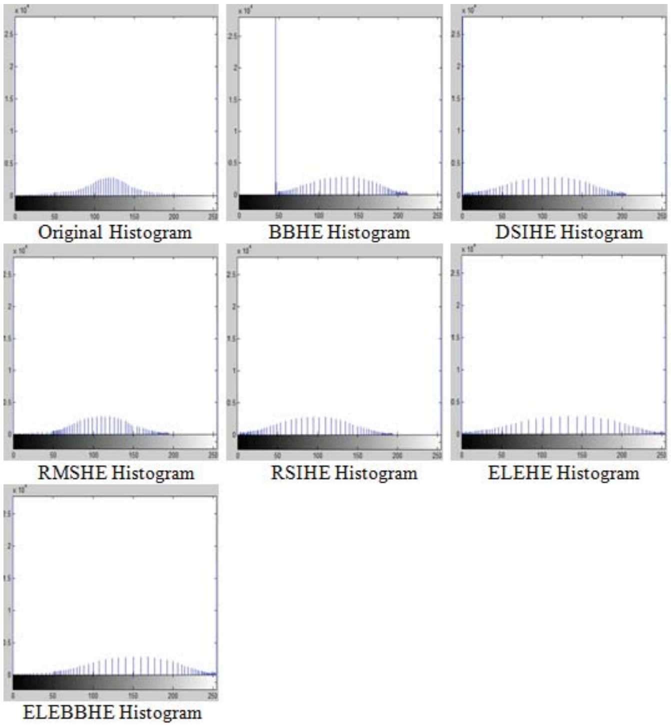


Original Image BBHE Image DSIHE Image

RMSHE Image RSIHE Image ELEHE Image

ELEBBHE Image

Figure 6: Images of various contrast enhancement techniques applied on CT brain image set II



Original Histogram

BBHE Histogram

DSIHE Histogram

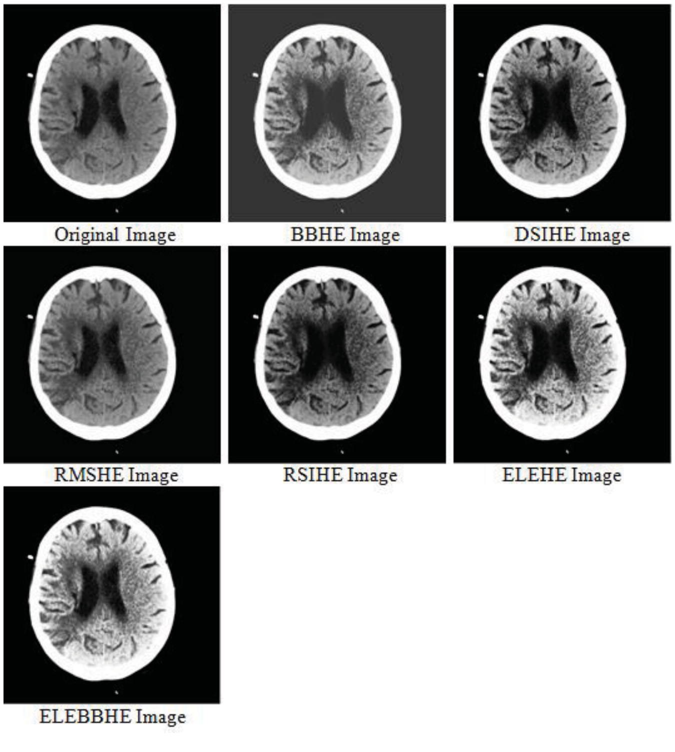
RMSHE Histogram

RSIHE Histogram

ELEHE Histogram

ELEBBHE Histogram

Figure 7: Image histograms of various contrast enhancement techniques applied on CT brain image set II



Original Image

BBHE Image

DSIHE Image

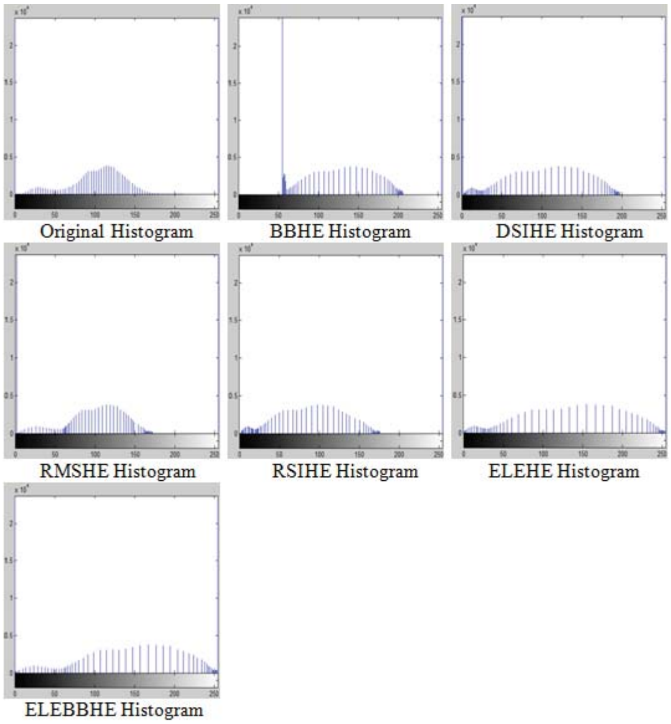
RMSHE Image

RSIHE Image

ELEHE Image

ELEBBHE Image

Figure 8: Images of various contrast enhancement techniques applied on CT brain image set III



Original Histogram

BBHE Histogram

DSIHE Histogram

RMSHE Histogram

RSIHE Histogram

ELEHE Histogram

ELEBBHE Histogram

Figure 9: Image histograms of various contrast enhancement techniques applied on CT brain image set III

Although there are some improvements on RMSHE technique from existing BBHE technique, it may not be proper to highlight the region of interest (ROI) in some cases. The new proposed ELEBBHE technique outperforms the five existing techniques. It can provide better visualization of CT brain image which make a clear different between ROI and normal brain soft tissue.

From the image histograms in Figure 5, Figure 7 and Figure 9, it can be observed that there is a high peak distribution around intensity range of 30 to 60. This indicates that the background of BBHE enhanced images is no longer in black color (intensity 0). On the other hand, it can be seen that there are much frequent distribution of pixels on the lower intensity range for DSIHE, RSIHE and ELEHE technique. These cause the whole image or the ROI to be slightly darker, while ELEBBHE technique has a much even distribution.

The comparison of the new proposed technique and the five existing techniques is also done with the image quality assessment (IQA) module such as peak signal to noise ratio (PSNR) and measure of enhancement by entropy (EMEE). The results are tabulated in Table 1.

Table 1: Comparison on peak signal to noise ratio and measure of enhancement by entropy of various contrast enhancement techniques for CT brain image set I, CT brain image set II and CT brain image set III

Technique \ CT Brain Image		Set I	Set II	Set III
Original	PSNR	-	-	-
	EMEE	3.9838	3.8511	3.2753
BBHE	PSNR	17.9742	16.2858	15.1732
	EMEE	0.1392	0.1298	0.1115
DSIHE	PSNR	31.3529	25.6934	25.1535
	EMEE	0.5874	0.6051	0.5434
RMSHE	PSNR	34.1403	32.2229	34.2331
	EMEE	3.9709	3.8443	0.4312
RSIHE	PSNR	31.7145	24.6420	24.6637
	EMEE	3.9691	3.8318	3.2871
ELEHE	PSNR	24.8081	22.0988	19.1852
	EMEE	4.0011	3.8805	3.4152
ELEBBHE	PSNR	24.3613	21.6879	18.4601
	EMEE	4.0042	3.8876	3.4254

From Table 1, RMSHE technique has the highest peak signal to noise ratio (PSNR) value for all the three sets of CT brain images, whereas the new proposed ELEBBHE technique gives the highest measure of enhancement by entropy (EMEE) value for all the CT brain images. Although, the PSNR value of the new proposed ELEBBHE technique are slightly lower than some of the existing techniques, but by comparing the enhanced image, it still provides a better visualization. The reason is that ELEBBHE can intensify the hypodense area and enhance the intensity of normal brain soft tissue. From numbers of CT brain images tested, it is also noticed that it is very difficult to diagnose on early infarct cases when the enhanced image provides too high PSNR value.

8 Conclusion

In conclusion, the diagnosis of ischemic stroke is very difficult when the CT brain image captured is low in contrast. For ischemic stroke detection, contrast is crucial to highlight the hypodense area. Time is also the key to safe a person's life. In this paper, a new proposed contrast enhancement technique called ELEBBHE technique for CT brain image is formulated. This new technique is capable to improve the overall contrast of CT brain image and intensify the region of interest (ROI). The new proposed ELEBBHE technique provides better enhancement on entropy and better visualization as compared to the existing BBHE, DSIHE, RMSHE, RSIHE and ELEHE technique.

9 References

[1] B. M. Gund, P. N. Japtap and R. Y. Patil, "Stroke: A Brain Attack", IOSR Journal of Pharmacy, Vol. 3, No. 8, pp. 1-23, 2013.

[2] R. Y. Kwong and E. K. Yucel, "Computed Tomography Scan and Magnetic Resonance Imaging", Circulation, Vol. 108, No. 15, pp. e104-e106, 2003.

[3] I. K. Indrajit, "Digital Imaging and Communications in Medicine: A Basic Review", Indian Journal of Radiology and Imaging, Vol. 17, No. 1, pp. 5-7, 2007.

[4] J. Hsieh, "Computed Tomography Principles, Design, Artifacts, and Recent Advances", Bellingham, WA: SPIE, 2009.

[5] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", New Jersey: Prentice Hall, pp. 120-144, 2008.

[6] Y. T. Kim, "Contrast Enhancement Using Brightness Preserving Bi-Histogram Equalization", IEEE Transactions on Consumer Electronics, Vol. 43, No. 1, pp. 1-8, 1997.

[7] Y. Wang, Q. Chen, B. M. Zhang, "Image Enhancement Based on Equal Area Dualistic Sub Image Histogram Equalization Method", IEEE Transactions on Consumer Electronics, Vol. 45, No. 1, pp. 68-75, 1999.

[8] S. D. Chen and R. Ramli, "Contrast Enhancement Using Recursive Mean-Separate Histogram Equalization for Scalable Brightness Preservation", IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, pp. 1301-1309, 2003.

[9] K. S. Sim, C. P. Tso, Y. Y. Tan, "Recursive Sub-Image Histogram Equalization Applied to Gray Scale Images", Pattern Recognition Letters, Vol. 28, No. 10, pp. 1209-1221, 2007.

[10] T. L. Tan, K. S. Sim, A. K. Chong, "Contrast Enhancement of Computed Tomography Images by Adaptive Histogram Equalization – Application for Improved Ischemic Stroke Detection", International Journal of Imaging Systems and Technology, Vol. 22, No. 3, pp. 153-160, 2012.

Novel Texture Pattern Based Multi-level set Segmentation in Cervical Cancer Image Analysis

Arti Taneja[1], Dr. Priya Ranjan[2], Dr.Amit Ujlayan[3]

[1]Research Scholar, Amity Institute of Information Technology, Uttar Pradesh, Noida -201303, India. Ph.: 0120-4392277
[2]Professor, Amity University, Uttar Pradesh, Noida, India. Ph.: 0120- 4392277.
[3]Professor, Gautam Buddha University, Greater Noida, India.

Abstract - *Computerized framework development is alternate to the Manual Method (MM) of cervical cancer analysis since MM suffers from the human errors, bulk quantities of Pap smear images, work loading and time complexities. Also, severity level prediction via counting of nucleus leads to incorrect prediction under geometric-based feature extraction. The cell-based segmentation evolved in research studies assures the automatic assistance with an assumption of a single cell. Practically, this assumption is not suitable since the image contains more cells. The multiple cell splitting for a number of cells is the challenging task. Also, the complex cell structure, poor contrast, and overlapping affect the cell segmentation performance. The cell contains a nucleus that describes the significant changes due to the disease. Hence, the cell boundary prediction is the complex task of geometric-based feature extraction techniques. This paper proposes the four new methods to update the cervical image processing tasks. First, the Neighborhood Concentric Filtering (NCF) is used to remove the noise present in the image and enhance the intensity level. Then, the cluster formation based on the intensity difference level estimation provides the multi-label output. Second, the Optimal Weight Updating with the Multi-Level set (OWU-ML) estimates the Region of Interest (ROI-Nucleus), extracts the nucleus texture features with an edge intensity information and form the window. Also, the intensity weight update by the OWU efficiently separates the layers that form the active contour over the image. Here, the Gray Level Co-occurrence Matrix (GLCM) extracts the texture pattern features of the nucleus portions in the form of angle variations. Finally, the Neural Network-based RVM classifier predicts the classes of (class 1 and class 2) cervical images. The optimal weight update, the GLCM features based multi-level set segmented output and the NNRVM classification improves the performance of severity level prediction in cancer treatment provision.*

Keywords: Cervical Image Analysis, Gray Level Co-occurrence Matrix, Multi-Label output, Neighborhood-Concentric Filtering, Neural Network-Relevance Vector Machine, Optimal Weight update.

1 Introduction

Global cancer statistics reports show that the death rate due to the cancer is more in the developing countries and need to be reduced. The observation of cancer incidence and the mortality patterns is the preventive action for cancer burden. Automation-assisted methods evolved in research studies are based on cervical cytology screening and this is the labor-oriented process with high-intensity. The cytological preparation is the prerequisite for the screening operation. The Integration of Manual Liquid-based Cytology (MLBC) with the hematoxylin and eosin (H&E) stain effectively prepares the cytology for cervical screening. The morbidity and the mortality prevention from the cervical images are the major objectives of cervical screening. Age-approximate and the maintenance of abnormal screening results requires an optimal strategy that has the capability to avoid the unnecessary treatment and makes the screening as a sensible one.

Research studies lie in two aspects namely, the nuclei detection in either single or overlapping cells and the nuclei detection in both single and overlapping cells. The contour enhancement approach for boundary detection employs the probability estimation of Gradient Vector Flows (GVF) for each image pixel. The integration of edge map computation method and the stack-based refinement makes the GVFs are robust. The GVF implementation and the Active Contour Model (ACM) integration requires the proper initial contour and this is not suitable for practical implementation. The Nucleus Cytoplast Contour (NCC), Adaptive Threshold Decision (ATD) and the Gray Level Gradient Difference (GLGD) extract the nucleus from the images and segment the required objects from various images

The Cytological Testing-based cervical screening has the limitations of less sensitivity, reproducibility, and specificity. Human Papillomavirus (HPV) diagnosis is an alternative to CT that has the ability of a prior indicator of cervical cancer stage and it is preferable in clinical analysis. The combination of Pap smear and the DNA images plays the major role in HPV diagnosis. The Pap smear image analysis effectively reduces the death rate. But, the evolution of stable H&E stain method provides the accurate analysis.

The less operational speed of image screening and diagnosis leads to optical imaging techniques development. Fluorescence lifetime imaging microscopy (FLIM) provides the additional information regarding the lifetime independent of excitation power fluctuations and the fluorophores concentration. The FLIM-based screening provides the better screening and diagnosis than the H & E imaging method. But, the huge time-consuming nature of H & E stain methods leads to the automatic assistance methods generation. Automated methods applicable in segmentation offers the higher accuracy than the manual processing. The attractive stage in nuclei segmentation is the Graph Cut (GC) approaches employment. The prior knowledge regarding the nuclei shape and the incorporation of manual annotation provide the more robust segmentation performance. The basic assumption of these methods is the image contains only one cell is not suitable in practical situations. Hence, the multiple touching-based cell splitting in cervical image analysis plays the major role in cell-segmentation techniques.

Research works turns to segmentation of overlapping cells that depends on the appropriate nuclei and background marker selection. The overlapping cell segmentation is the combination of nuclei boundary characteristics and the prior knowledge regarding the expected shape. The cervical cancer treatment provision requires the prior cancer detection. The increase in mean scattering coefficient due to the acidic acid agents leads to the abnormal tissue appearance within the normal regions. The evolution of optical imaging techniques in research studies assures the effective diagnosis. The coordinated and cytology screening mechanisms implementation results show that the cervical disease is the major problem compared to other. The microscopic-based liquid cytology evaluation offers the high efficiency and the sensitivity values. But, the false rates are also high during the practical implementation. The automated cytology and Human Papilloma Virus (HPV) analysis reduce the false rates. The abnormal Pap smear image utilization in HPV analysis leads to slowness and low morphologic perceptibility. The statistical results from the AAR methods convey that the part in the cell (nuclei) describes the variations due to the disease affection. Hence, an accurate nuclei boundary description is the necessary task and it is crucial in PAP smear image analysis. The normal/abnormal cell discrimination in PAP smear images includes the quantification of nuclei changes that lead to the evolution of structure element based segmentation.

The cell structure complexities for high cytoplasm variations are high for the unknown cells and location. The generic and parametric methods utilization offers the better cells delineation than the existing shape segmentation methods. But, the poor contrast and the inconsistent staining leads to the morphological feature extraction, watershed segmentation, and the nuclei separation. The super pixel algorithms evolution in research studies contributes the image boundaries prediction, speed, memory, and efficiency improvement. The simple linear iterative clustering adapts the k-means clustering algorithms for the super pixels generation. The modern super pixel generation algorithms offer the better results compared to the traditional superpixel segmentation algorithms. The structural parts segmentation based on template fitting, edge detectors, and the active contour models has the high performance compared to the other segmentation algorithms. The large cells, overlapping and the non-appropriate image artifacts cause the limitation in recognition of boundaries that contains only one cell. The energy function optimization depends upon the elliptical shape, area overlap and intensity ratio. But, the cervical images are not exact elliptical and the adaptation of elliptical shape also not offered the solution to the accurate segmentation problem. The provision of controlling the overlapping cells enhances the accuracy of segmentation. The novel technical contributions of proposed Novel Texture Pattern-based Multi-level set Segmentation (NTPMS):

- Nucleus texture pattern extraction through the optimized cell segmentation techniques is proposed.
- The multi-level set algorithm proposed in this paper extracts the exact nucleus portion in the cervical images irrespective of the geometrical features.
- Periodical estimation of intensity difference values forms the cluster that provides the multi-label output.
- The optimal weight update in proposed algorithm separates the layers which form the active contour over the image.
- Nucleus texture pattern extraction by the GLCM provided the clear image analysis and normal/abnormal classification

This paper is organized as follows: Section II describes the related works on cervical cancer image analysis. Section III discusses the implementation process of proposed Novel Texture Pattern-based Multi-level set Segmentation (NTPMS) algorithm. Section IV presents the performance analysis of NTPMS regarding the accuracy, precision, recall, sensitivity, specificity and the coefficient metrics. Finally, section V presents the conclusion.

2 Related Work

This section describes the traditional feature extraction and segmentation techniques influenced in the cervical image analysis. The reports from the different medical societies declare that the cancer is the second leading cause of death. There are several cancer treatments such as radiotherapy, surgery etc. has the limitations such as drugs selection related to diseases and toxicities possession in cancer drugs. *Saslow et al* [1] reviewed the update of the American Cancer Society (ACS) reports and provided the new screening recommendations that addressed the age-appropriate screening strategies with the high-risk human papillomavirus (HPV) basis. *Jemal et al* [2]presented the statistics about the cancer incidence and mortality rates that conveyed overall incidence rates in developing world are half of the rates in developed world. The application of cancer control knowledge and the physical activity monitoring significantly reduced the global cancer burden.

The cervical cancer diagnosis through the screening process is the labor consuming and time complexity. *Govindaraju et al* [3] investigated the green synthesis that explored the silver nitrate (Ag NO_3) influence on cervical cell analysis. With the considerations of selective toxicity and therapeutic index, the new horizon is declared for cell nuclei splitting. The Photo Acoustic diagnosis provides the solution to Cervical Cancer screening problems. *Peng et al* [4]carried out the PAI experiments for the Depth Maximum Amplitude Projection (DMAP) analysis. They employed the paired-t-test for an indication of Mean Optical Absorption (MOP) difference that exists in between normal and cervical tissue variation for high confidence value. The molecules, cells and tissues identification and characterization require a unique bio-medical fingerprint and it is the major tool in cervical cancer analysis with the ability for malignancy and premalignancy stages. *Ramos et al* [5] summarized the research areas and assured the HPV detection and monitoring response compared to diagnosis perspective. They also presented the comprehensive studies to suggest the Raman spectroscopy validation for molecular diagnostics and cancer analysis.

In cervical screening, the morphological and biochemical properties alternation for malignant transformation approach in the optical imaging techniques improve the performance. *Orfanoudaki et al* [6] utilized the contrast agents to provide the three-dimensional cell clusters that provide an effective knowledge of biological events and the multi-center randomized cells utilization offers the high standardization. The Automated Assisted Reading (AAR) techniques deployment in cervical screening process has the less error rate and the maximum productivity which depends on the accurate segmentation of abnormal cells. *Zhang et al* [7] proposed the global/local scheme with the Graph-Cut (GC) approaches that utilized the combination of normal and abnormal cells. The tumor histopathology characterization defined the nuclear regions from the H & E tissue sections. *Chang et al* [8] performed the automated analysis by using the nuclear segmentation formulation within the graph framework. They presented the Multi-Reference Graph Cut (MRGC) with the prior knowledge regarding the reference and local image features. *Zhang et al* [9] introduced the auto-focusing method that rejected the coverslip and the actual focal plan was extracted. The hybrid global and local scheme segmented the normal and the abnormal cells. The contextual and cytoplasmic information capture improved the specificity. Early detection of cervical cells is the challenging task in standardized treatment provision. The accurate detection and segmentation were limited by the cells overlapping and the less contrast. *Happy et al* [10] presented the unsupervised approach called Extended Depth of Field (EDF) for accurate cell nuclei segmentation.

The metabolism changes indication in EDF effectively detected the pre-cancer development status. *Wang et al* [11] used the Fluorescence lifetime imaging microscopy (FLIM) for metabolic changes detection. They studied the application of FLIM to the unstained tissues with the morphological features. The fluorescence lifetime analysis showed that the FLIM provided the accurate results than the H & E method. The adaptive GC method combines the intensity, texture, and shape and boundary information and offers the better abnormal cells segmentation. The accuracy enhancement in CC screening analysis involves the nuclei cell distribution and shape size analysis. *Mahanta et al* [12] provided the automated process to predict the cell abnormalities in the screening process. The interactive segmentation of normal/abnormal cells via fuzzy based histogram computation. The touching-based nuclei splitting involved the marker generation by the learning of texture, shape, and contextual information. *Talukdar et al* [13] proposed the validity measures for FCM such as Partition Coefficient (PC), Partition Entropy (PE), compactness and the separation function. *Song et al* [14] proposed the combination of multi-scale convolution network and the graph partitioning for an accurate segmentation. The refinement via coarse-to-fine segmentation reduced the complexities effectively. They also provided the enhancement to establish the cut-off values between the normal/abnormal patterns and the abnormal values classification based on cancer stage. The true nuclei determination involves the shape, text and the image intensity characterization reduces the false positive findings. *Plissiti et al* [15] presented the automated method for boundary detection and cell nuclei determination. They examined the unsupervised and supervised classification techniques with the feature selection schemes with the minimal redundancy and maximal relevance.

The Pap smear image utilization suffers from the low value of sensitivity and the specificity leads to the HPV diagnosis. *Tasoglu et al* [16] presented the need of simple, easy HPV diagnosis methods and reviewed the existing methods to provide the future directions to cervical analysis. The manual screening-based Pap smear image test required the color and shape properties to make the automated one. But, the automated process was suffered from the cell structure complexities. *Genctav et al* [17] proposed the unsupervised approach for cervical cells classification/segmentation. The sequential processes such as automatic thresholding, hierarchical segmentation, and the binary classification maximized the similarities and offered the consistent staining without any parameters adjustment. The segmentation result depends upon the pixels selection belongs to a particular class. The complexities and non-obvious nature of Pap smear images affected the general assumption such that the pixels are distributed in nature. The pixels associated with the nuclei are noisy pixels that need an isolation process. *Plissiti et al* [18] presented the automation method for cell nuclei prediction in Pap smear images. They performed morphological analysis to detect the centroid locations of nuclei and it is incorporated with the nucleus circumference. The fully automated method had the ability to handle the overlapping images. Hyperthermia is a medical therapy for enhanced cancer treatments in which the external energy is applied to raise the tumor region temperature. The cell viabilities decrease immediately to the simulation starts requires an optimal labeling process. The concentration dependent toxicity effects required the revision in magnetic labeling process. *Huang et al* [19] optimized the magnetic labeling process for the analysis of cytotoxicity's evaluation. They also obtained the condition for magnetic label optimization that leads to a reduction in cell viabilities. The obscure boundaries locating and the noise sensitivity leads to an extension of Gradient vector Flow (GVF) models to the radiating modules (RGVF). *Li et al* [20] proposed RGVF that is used for segmentation of initial contours. The edge map computation and the stack based refinement in RGVF assured the robustness against the contaminations and located the obscure boundaries effectively.

The small size nucleus compared to the cytoplasm and the background leads to a poor threshold in Adaptive Threshold Decision (ATD) for discrimination. *Pai et al* [21]presented the Maximal Gray Level Gradient Difference (MGLGD) for nuclear extraction. The contour formation in NCC dependent upon the gray level difference between the nucleus and cytoplasm that offered the superior performance than the existing GVF-ACM-ATD without requiring proper contour initialization. The compact representation of shape and the active shape model utilization are necessary for prior knowledge regarding the expected shape. *Plissiti et al* [22] detected and described the unknown nuclei boundaries in the images. By using the weight parameters that controls the force and energy of deformable models. The problem inaccurate boundary detection was resolved by the weight parameter-based overlapping nuclei prediction. The noise and cell occlusion in overlapping images degrade the performance. *Nosrati* and *Hamarneh* [23] proposed the variational segmentation using the star-shaped prior using the directional derivatives for overlapping cells segmentation in Pap smear images. They introduced the Voronoi energy term that controls the neighbor cells overlapping. The accuracy and computational time of optimized approaches better than the non-optimized approaches. The reduction in preprocessing complexities requires the fast super pixel computation and the easy utilization. The segmentation quality and the operational speed improvement depends on the extracted superpixels. *Achanta et al* [24] introduced the linear iterative algorithm called SLIC that includes the *k*-means clustering approach for an efficient pixels generation. The SLIC implementation also offered the better segmentation performance and the maximum operational speed. The cluster of distracting pixels affected the super-voxels segmentation process and hence the proper algorithm is required for computational complexity reduction. *Lucchi et al* [25] proposed the automated graph partitioning methods that incorporate the shape and distinctive shape learning for better recognition. They demonstrated the better computational efficiency and segmentation quality. The review concluded that the suitable weight estimation and the multi-level set formulation are required to provide the optimal trade-off between the accuracy improvement and clear image analysis.

3 Novel Texture Pattern-based Multi-level set Segmentation

This section illustrates the methods involved in proposed work. The flow diagram of proposed Novel Texture Pattern-based Multi-level Segmentation (NTPMS) consists of successive processes such as Neighborhood Concentric Filtering (NCF), Optimal Weight Update-Multi-level set segmentation, Gray Level Co-occurrence Matrix (GLCM) feature extraction, and the Neural Network- Relevance Vector Machine (NN-RVM) classification as in Fig. 1.

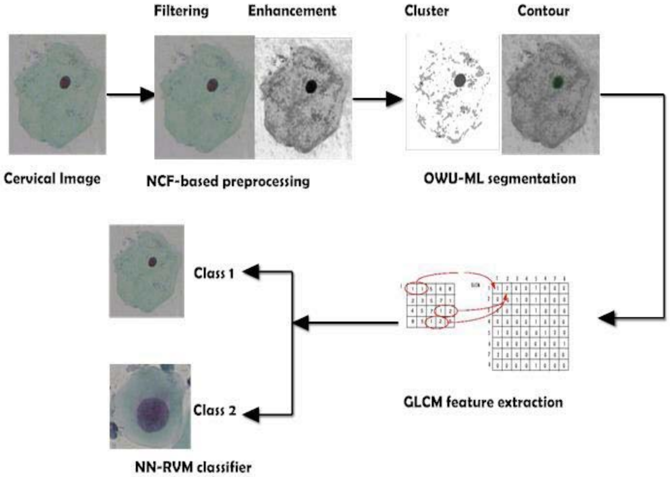


Fig. 1 Flow diagram of NTPMS.

Initially, Neighborhood Concentric Filtering (NCF) removes the noise through the connected component analysis. The interpretation of information in the images is the necessary prior task in the segmentation process. Hence, the Gaussian model is used to enhance the image quality for better human perception. Then, the energy, direction and weight update in diverse directions from the active contour. The unique form of image representation depends on the performance of feature extraction. Here, the Gray Level Co-occurrence Matrix (GLCM) is used to extract the various features for unique representation. Finally, the Neural Network-Relevance Vector Machine (NN-RVM) classifier provides the necessary labeling.

3.1 Neighborhood Concentric Filtering

The good quality images are the major requirement for feature extraction and the segmentation. The noise removal and the image enhancement are the initial stages of proposed work. The window formation by using the connected components extraction removes the noise present in the images. The algorithm for Neighborhood Concentric Filtering (NCF) is as follows:

Initially, the matrix window with the size 3×3 is initialized for the input cervical image. Then, the Connected Components (CC) are extracted from the matrix form and maximum CC is computed and erased. Finally, the noise-free image *Y (i, j)* is formed by comparing the differences center pixel/boundary with the center pixel$temp(5)$. If the difference is greater than the center pixel value, then the matrix cell is replaced with the average value otherwise they are replaced with the median value. The interpretation of the information in the images requires the enhanced image form. Hence, the Gaussian model with the standard deviation (σ) and the image gradient (I_g) values are used to find the enhanced image (I_e) as follows:

$$\sigma = \sqrt{\frac{1}{N*M}\sum_{i=1}^{M*N}\left(I_g(i) - \frac{\sum I_g}{N}\right)^2} \tag{1}$$

$$I_e = \frac{I_g}{\max\left(I_g\left(\frac{\sum I_g}{N}*\sigma\right)\right)} \tag{2}$$

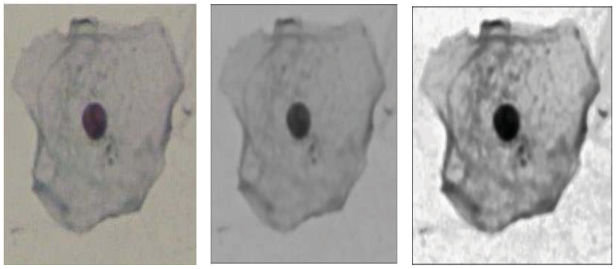
Where, M, N –Row and column size of the image. Fig. 2 (a), (b) and (c) shows the input image, NCF filtered image and the enhanced image. The clear shape boundary description with the minimum energy consumption is the major requirement to track the dynamic objects.

Neighborhood Concentric Filtering

Input: *Cervical image 'I'*

Output: *Preprocessed Image, 'Y'*

```
Step 1: Initialize window size (3×3).
Step 2: Extract Connected Components, CC.
Step 3: $Idx = max(CC)$.
Step 5: $I = I(Idx)$ //Update Image with maximum connected
                      Component of the pixel.
Step 6: for (i = 2 to $Row\_Size\ (I) - 1$) //'i' Row size of
                                              image
Step 7:    for (j = 2 to $Column\_Size\ (I) - 1$)//'j' Column
                                              size of image
Step 8:        temp= $I_{i-1\ to\ i+1, j-1\ to\ j+1}$ //Project window
                                   over image matrix
Step 9:        if ((temp (5) ~ temp (Boundary)) > temp (5))
                          //Check neighboring Pixel variation.
Step 10:           Y (i, j) = Avg. (temp);
Step 11:       else
Step 12:           Y (i, j) = median (temp);
Step 13:       end if
Step 9:    end loop 'j'
Step 10: end loop 'i'
```



(a) (b) (c)

Fig. 2 (a) Input image, (b) Filtered image and (c) Enhanced image

The model that is helpful for clear boundary analysis is called active contour model. The segmentation accuracy depends on the convergence state of energy minimization that requires the optimal weight update.

3.2 Optimal Weight Update-Multi-level Set Segmentation

The important stage in proposed NTPMS is the multi-level segmentation process. The energy, weight, and direction updates are forms the active contour over the region. The algorithmic steps to perform the multi-level segmentation based on optimal weight as follows:

Multi-level set segmentation

Input – Enhanced image, 'I_e', ROI Mask, 'R_{out}', Texture Pattern, 'ATP'

Output – Clustered Output, 'I_C'

```
Step 1: Initial Masking
$M = \sqrt{\left(R_{out}(x)\right)^2 - \left(R_{out}(y)\right)^2} - \sqrt{\left(1 - R_{out}(x)\right)^2 - \left(1 - R_{out}(y)\right)^2} + R_{out} - \frac{1}{2}$
Step 2: For iteration = 1 to N
Step 3: $Idx = Index(M)$
Step 4: $Curvature, \frac{\partial u}{\partial t} = \nabla uM(Idx) + k$
Step 5: Energy Update,
$I_p = \begin{Bmatrix} M & if\ (M \le 0) \\ 0 & else \end{Bmatrix}$, Internal Energy
$E_p = \begin{Bmatrix} M & if\ (M > 0) \\ 0 & else \end{Bmatrix}$, External Energy.
    $Energy, I_E = \frac{\left(I_g(Idx) + \sum I_p\right)}{max\left(\left(I_g(Idx) + \sum I_p\right)\right)} + \alpha * \frac{\partial u}{\partial t}$
Step 6: Difference in energy update,
    $dt = \frac{t}{max(I_E)}$
Step 7: Direction Update
    $D_{Pos} = \sqrt{max(ap^2, bn^2) + max(cp^2, dn^2)}$
    $D_{Neg} = \sqrt{max(an^2, bp^2) + max(cn^2, dp^2)}$
Step 8: Contour Weight Update
    $\emptyset_{i+1} = \emptyset_i - dt * \frac{\emptyset_i}{20 * \sqrt{\emptyset_i^2 + 1}} * (M + dt * I_E)$
Step 9: End for
```

Initially, the Region of Interest (ROI) masking is the prior computation in the segmentation process. Then, the following processes are performed for each iteration.

- The index corresponds to the masked form is computed and based on the difference of pixel variations in eight directions with the spacing of *0°, 30°, 45°, 60°, 90°, 120°, 135°, 180°* and the opposite values, the curvature ($\frac{\partial u}{\partial t}$) is computed for the images in step 4.

- The energy of contour formation comprises the internal and external energy values. Based on the masked value (either > 0 or <=0), the internal and the external energy values are updated with the mask value in step 5:

$$I_p = \begin{Bmatrix} M & if\ (M \leq 0) \\ 0 & else \end{Bmatrix} \tag{3}$$

$$E_p = \begin{Bmatrix} M & if\ (M > 0) \\ 0 & else \end{Bmatrix} \tag{4}$$

- The optimized energy is computed by using the following formula in step 6.

$$I_E = \frac{\left(I_g(Idx)+\sum I_p\right)}{max\left(\left(I_g(Idx)+\sum I_p\right)\right)} + \alpha * \frac{\partial u}{\partial t} \tag{5}$$

- The existing energy level is updated with the computed maximum energy and the optimized new energy level has the great impact of weight update.
- The diverse directional coefficients (positive, negative, forward, backward, left and right- $ap, an, bp, bn, cp, cn, dp, dn$)-based directional update supports the clear image analysis in step 7.
- Finally, the contour weight is updated with the computed mask, energy and the updated energy level in step 8.

Fig. 3 (a) and (b) shows the binary mask image and the multi-level segmented output.

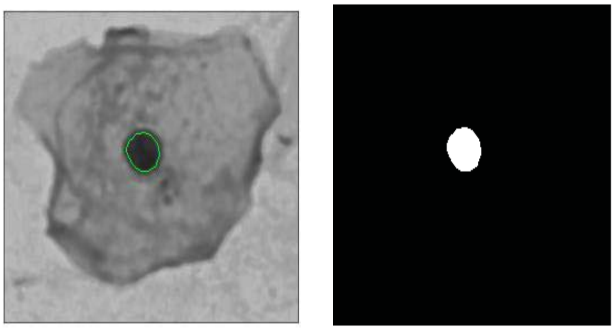


Fig. 3 (a) Binary Mask image, (b) Multi-level set segmented output

From the segmented output, the features extraction is the next stage to provide the unique form of the image. The proposed NTPMS utilizes the Gray Level Co-occurrence Matrix (GLCM) for the feature extraction process.

3.3 GLCM feature extraction

The matrix representation in which the number of rows and columns represent the gray levels of the clustered output image (I_C) is referred as Gray Level Co-occurrence Matrix (GLCM). The relative frequency variations of the pixels separated by the distance, intensity and the angle in the matrix element includes the statistical probability values$P(i,j|d,\theta)$. The increase in the number of pixel levels will increase the dimensionality of the GLCM features storage. Hence, the gray level reduction is preferred for storage minimization. Table 1 shows the features extracted from the GLCM for unique representation. For the number of neighboring gray levels (N), the statistical probability (p) values of two gray levels (a, b) and their mean (μ_x, μ_y) standard deviations (σ_x, σ_y) are used to estimate the texture features.

Table 1 GLCM Features

S.no	Feature	Equation
1	Contrast (C_t)	$C_t = \sum_a^N \sum_b^N (a-b)^2 p(a,b)$
2	Correlation (C_r)	$C_r = \sum_a^N \sum_b^N \frac{p(a,b)-\mu_x\mu_y}{\sigma_x * \sigma_y}$
3	Cluster Prominence (C_P)	$C_P = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} \{i + j - \mu_x - \mu_y\}^4 * p(a,b)$
4	Cluster shade (C_s)	$C_s = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} (a + b - u_x - u_y)^3 p(a,b)$
5	Dissimilarity (D)	$D = \sum_{a,b=1}^{N} C_{a,b} \lvert a-b \rvert$
6	Energy (E)	$E = \sum_{a=0}^{N} p^2(a,b)$
7	Entropy (Ent)	$Ent = \sum_{a=0}^{N} \sum_{b=0}^{N} p(a,b)\log(p(i,j))$
8	Cluster homogeneity (C_h)	$C_h = \sum_a \sum_b \frac{P_d[a,b]}{1+\lvert a-b \rvert}$
9	Overall homogeneity (Homop)	$Homop = \sum_a \sum_b \frac{1}{1+abs(a-b)} X_{ij}$
10	Maximum Probability (P_{max})	$P_{max} = \max_{a,b} p_d[a,b]$
11	Variance (Var)	$Var = \sum_{a=o}^{N-1} \sum_{b=0}^{N-1} (i-u_x)^2 . p(a,b) + \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} (j-u_y)^2 . p(a,b)$
12	Auto Correlation (AC)	$AC = \frac{XY}{(X-a)(Y-b)} * \frac{\sum_{a=1}^{X-a} \sum_{b=1}^{Y-b} f(a,b)f(a+m,b+n)}{\sum_{a=1}^{X-a} \sum_{b=1}^{Y-b} f^2(a,b)}$
13	Average Kurtosis (AK)	$AK = \frac{\sum\left(\frac{1}{\sigma^4} \sum_a \sum_b ((a*b)-\mu)^4 (p_{ab})^{-3}\right)}{N}$
14	Average Skewness (AS)	$AS = \frac{\sum\left(\frac{1}{\sigma^3} \sum_a \sum_b ((a*b)-\mu)^3 (p_{ab})\right)}{N}$

The extracted features from GLCM computation have the great impact on the classification and the pattern recognition like either normal or abnormal.

3.4 Neural-Network Relevance Vector Machine

The dependency investigation of targets on the input requires the supervised learning approach. The supervised model includes the input vectors $\{x_n\}_{n=1}^{N}$and associated target values$\{t_n\}_{n=1}^{N}$. The target values represents the class labels obtained from the classification process. The overfitting in training set due to the real data requires the suitable overlap analysis. The probabilistic sparse kernel that adopts Bayesian approach for learning the overfitting samples refers Relevance Vector Machine (RVM). The number of predictions from the RVM is based on the function described as

$$y(x) = \sum_{n=1}^{N} \omega_n K(x, x_n) + \omega_0 \tag{6}$$

The function defined in (6) represents the relationship between the model weights (ω_n) and kernel function $K(.,.)$ in terms of input samples. The non-associated input samples with the non-zero weights are close to the decision boundary refers “relevance” vectors. The prediction of posterior membership for the given class and the optimal solution are the objectives of the RVM. The logistic sigmoidal function generalizes the linear model and the corresponding likelihood is computed for the class instances (c) as

$$P(c/w) = \prod_{i=1}^{n} \sigma\{(y(x_i)\}^{c_i}[1-\sigma\{(y(x_i)\}]^{1-c_i}$$
(7)

Where, $\sigma(y)$ is the logistic sigmoid function

$$\sigma\big(y(x)\big) = \frac{1}{1+\exp(-y(x))}$$
(8)

The most probable weights computation, iterative reweighted least square algorithm utilization and Gaussian approximation are repeated until the convergence criteria is satisfied. The iterative processes are modelled as the neural network training to provide the effective class labels. The features extracted from the GLCM are responsible for training of Neural Network (NN). The contour to be segmented is defined by the sequence of control points equal to the number of neurons in the network. The weights associated with each neuron denoted by the gain value. The membership function and repetitive weight update to classify the samples as either normal or abnormal.

```
NN-RVM
Input: Feature Set, 'T' and Testing feature vector, 'V'
Output: Y – Labeled Output
Initialize map radius size, radius sample updating rate, radius decay rate of Neurons for cluster extraction.
i, j – Row and Column iteration of Feature Matrix 'T' respectively.
Step 1: $Input_{Neuron} = T$; Particles = V;
Step 2: For t=1 to number of iteration
Step 3: Compute the summation
    $Sum_{Particle} = Sum_{Particle} + \sqrt{(map - Input_{Neuron})^2}$;
Step 4: $Dis = \sqrt{\Delta x^2 + \Delta y^2}$;
// Difference from original cluster size to extracted window size and calculate distance of Δx and Δy
Step 5: Update gain parameter for each sample $gain = update_{rate} * exp\left(-\frac{dis}{2 * update_{radius}}\right)$;
Step 6: Update map matrix according to gain. Where $wx, wy$ - window size
    $mw(x,y) = mw(x,y,a) + gain * \big(mw(wx,wy) - mw(x,y)\big)$;
Step 7: Update radius sample rate, radius of Neuron cluster area
    $update_{radius} = 1.0 + (update_{radius} - 1.0) * radius_{decay}$;
End loop
Step 8: $dis(j,k) = \left(\left(\big(sp(j) - sp(k)\big)^2\right)^{0.5}\right)$;
Step 9: $sp_{update} = (dis_{min}(sp) + sp_{previous})/2$;
// Update minimum distance level of particle and update particle
Step 10: $dif(i) = \big(x - cent(i)\big)^2$;
// Find the magnitude of the difference between particles and center of the window.
Step 11: $new_{Dist} = munimum(difference)$;
// Extract minimum difference and update that article as new fitness value in a cluster
Step 12: N=0; $sum_v = 0$;
For length of particles,
    $sum_v = sum_v + new_{dist}$;
    Step 13: If $new_{Dist}$ > 0,
        N=N+1;
End loop
Step 14: Update center as, $cent_{update} = \frac{sum_v}{N}$;
End loop
Step 15: If $cent_{update} = cent_{old}$,
        Y= Return Label;
    End if
```

The feature set from the GLCM and the testing vectors are extracted from the prior steps. The map reduces the size, sample update rate, and decaying rate are initialized as the first stage. The mean window size and the distance update from the step 3 to 11 offers the center update to the logistic function. The average center update value is equal to the old center, then the classified output is extracted from step 12 to 15. The output (Normal(class 1), Abnormal (class 2)) from multi-level set segmented form, GLCM extracted features offers the better severity level prediction in cervical images.

4 Performance Analysis

This section illustrates the performance validation of proposed Novel Texture Pattern-based Multi-level set Segmentation (NTPMS) regarding the sensitivity, specificity, accuracy, precision, and recall. Besides the comparative analysis of proposed NTPMS with the existing SVM and Neuro-fuzzy formulation on the coefficient metrics states the effectiveness in NTPMS in cervical image analysis.

Performance Metrics

The performance validation of the proposed NTPMS on the basic parameters and the comparative analysis of the existing SVM and the Neuro-Fuzzy formulation assures that the optimal weight update improves the segmentation accuracy. Table 2 presents the comparison of performance metrics for proposed NTPMS and the existing methods of SVM/Neuro-Fuzzy formulation.

Table 2 Performance Analysis

Parameters	NTPMS	SVM	Neuro Fuzzy
TP	105	108	52
TN	44	25	38
FP	4	23	86
FN	3	0	56
Sensitivity (%)	97.2222	100	48.1481

Specificity (%)	98.0392	52.0833	57.8431
Precision (%)	96.3303	82.4427	37.6812
Recall (%)	97.2222	100	48.1481
Jaccard Coeff	0.9776	0.8526	0.5449
Dice Overlap	0.9887	0.9204	0.7054
Kappa Coeff.	0.9904	0.6008	0.7065
Accuracy (%)	96.7949	85.2564	37.8205

Fig. 4 shows the comparative analysis of sensitivity, specificity measures for NTPMS, SVM, Neuro-Fuzzy formulation. The optimal weight update by the multi-level set formulation and the novel texture patterns (NTP) improves both the sensitivity and specificity values. With Neuro-Fuzzy formulation, the NTPMS offers 49.07 and 69.41 % better in sensitivity and specificity values. Even though the SVM offers the highest sensitivity (100), the substantial reduction in specificity occurs in SVM. But, the NTPMS offers 88.24 % better specificity due to the clear image analysis.

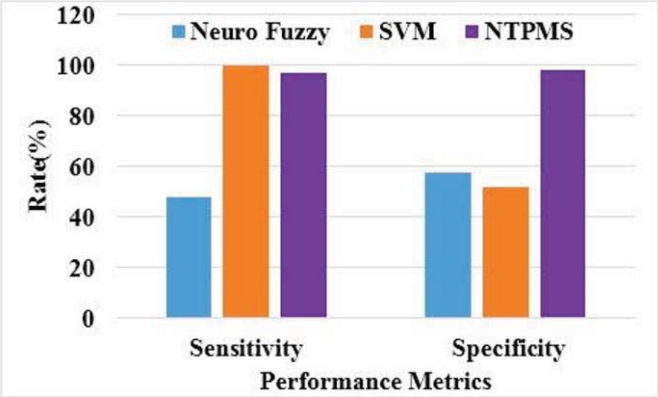


Fig. 4 Sensitivity, specificity analysis for NTPMS, SVM and Neuro-fuzzy models

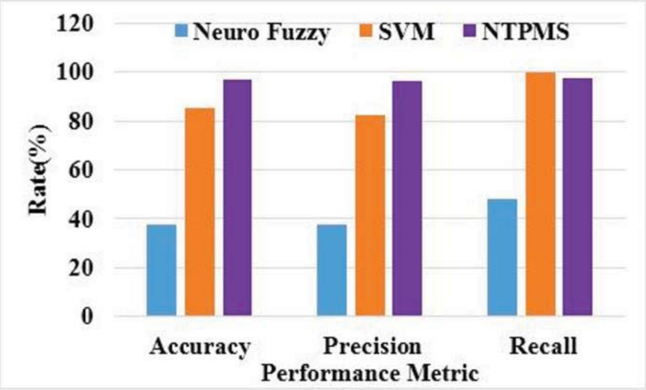


Fig. 5 Accuracy, Precision and Recall analysis for NTPMS, SVM and Neuro-fuzzy models

Fig. 5 shows the comparative analysis of accuracy, precision and recall values for NTPMS, SVM, and neuro-fuzzy formulation. The SVM offers 85.2564, 82.4427 and 100 % and the NTPMS offers 96.7949, 96.3303 and 97.2222 %. The optimal weight update unit by the ML formulation improves the accuracy and precision by 13.53 and 16.85 %.

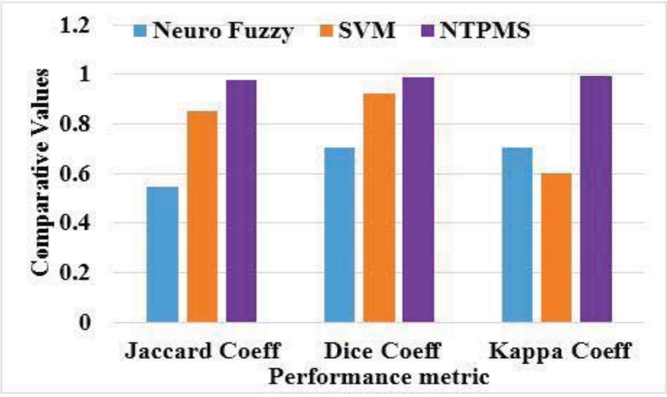


Fig. 6 Coefficient metrics analysis

Fig.6 shows the comparative analysis of coefficient metrics (Jaccard, Dice, and Kappa) for proposed NTPMS and the existing SVM and neuro-fuzzy formulation. The SVM offers 0.8526, 0.9204 and 0.6008 and the NTPMS offers 0.9776, 0.9887 and 0.9904. The optimal weight update unit by the ML formulation improves the coefficient metrics by 14.66, 7.42 and 64.85 %.

5 Conclusion

This paper discussed the limitations in severity level prediction in cervical cancer images and the Manual Methods (MM) and their solution via computerized framework. Here, the nucleus texture pattern is extracted through the optimized cell segmentation techniques. The employment of multi-level set algorithm extracted the exact nucleus portion in the cervical images irrespective of the geometrical features. The periodical estimation of intensity difference values forms the cluster that provides the multi-label output. The optimal weight update based intensity weight estimation in proposed algorithm separated the layers which form the active contour over the image. The nucleus texture pattern extraction by the GLCM provided the clear image analysis and normal/abnormal classification. An accurate severity level prediction is achieved by using the proposed GLCM-OWU-ML combination compared to the geometrical feature extraction algorithms.

References

[1] D. Saslow, D. Solomon, H. W. Lawson, M. Killackey, S. L. Kulasingam, J. Cain*, et al.*, "American Cancer Society, American Society for Colposcopy and Cervical Pathology, and American Society for Clinical Pathology screening guidelines for the prevention and early detection of cervical cancer," *CA: a cancer journal for clinicians,* vol. 62, pp. 147-172, 2012.

[2] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman, "Global cancer statistics," *CA: a cancer journal for clinicians,* vol. 61, pp. 69-90, 2011.

[3] K. Govindaraju, K. Krishnamoorthy, S. A. Alsagaby, G. Singaravelu, and M. Premanathan, "Green synthesis of silver nanoparticles for selective toxicity towards cancer cells," *Nanobiotechnology, IET,* vol. 9, pp. 325-330, 2015.

[4] K. Peng, L. He, B. Wang, and J. Xiao, "Detection of cervical cancer based on photoacoustic imaging—the in-vitro results," *Biomedical optics express,* vol. 6, pp. 135-143, 2015.

[5] I. R. M. Ramos, A. Malkin, and F. M. Lyng, "Current Advances in the Application of Raman Spectroscopy for Molecular Diagnosis of Cervical Cancer," *BioMed research international,* vol. 2015, 2015.

[6] I. M. Orfanoudaki, D. Kappa, and S. Sifakis, "Recent advances in optical imaging for cervical cancer detection," *Archives of gynecology and obstetrics,* vol. 284, pp. 1197-1208, 2011.

[7] L. Zhang, H. Kong, C. T. Chin, S. Liu, Z. Chen, T. Wang*, et al.*, "Segmentation of cytoplasm and nuclei of abnormal cells in cervical cytology using global and local graph cuts," *Computerized Medical Imaging and Graphics,* vol. 38, pp. 369-380, 2014.

[8] H. Chang, J. Han, A. Borowsky, L. Loss, J. W. Gray, P. T. Spellman*, et al.*, "Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association," *IEEE Transactions on Medical Imaging,* vol. 32, pp. 670-682, 2013.

[9] L. Zhang, H. Kong, C. Ting Chin, S. Liu, X. Fan, T. Wang*, et al.*, "Automation-assisted cervical cancer screening in manual liquid-based cytology with hematoxylin and eosin staining," *Cytometry Part A,* vol. 85, pp. 214-230, 2014.

[10] S. Happy, S. Chatterjee, and D. Sheet, "Unsupervised Segmentation of Overlapping Cervical Cell Cytoplasm," *arXiv preprint arXiv:1505.05601,* 2015.

[11] Y. Wang, C. Song, M. Wang, Y. Xie, and L. Mi, "Rapid, Label-free and Highly Sensitive Detection of Cervical Cancer with Fluorescence Lifetime Imaging Microscopy," *IEEE Journal of Selected Topics in Quantum Electronics* vol. 22, 2016.

[12] L. B. Mahanta, D. C. Nath, and C. K. Nath, "Cervix cancer diagnosis from pap smear images using structure based segmentation and shape analysis," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 3, pp. 245-249, 2012.

[13] J. Talukdar, C. K. Nath, and P. Talukdar, "Fuzzy Clustering Based Image Segmentation of Pap smear Images of Cervical Cancer Cell Using FCM Algorithm," *markers,* vol. 3, 2013.

[14] Y. Song, L. Zhang, S. Chen, D. Ni, B. Lei, and T. Wang, "Accurate Segmentation of Cervical Cytoplasm and Nuclei Based on Multiscale Convolutional Network and Graph Partitioning," *IEEE Transactions onBiomedical Engineering,* vol. 62, pp. 2421-2433, 2015.

[15] M. E. Plissiti, C. Nikou, and A. Charchanti, "Automated detection of cell nuclei in Pap smear images using morphological reconstruction and clustering," *IEEE Transactions on Information Technology in Biomedicine,* vol. 15, pp. 233-241, 2011.

[16] S. Tasoglu, H. Cumhur Tekin, F. Inci, S. Knowlton, S. Q. Wang, F. Wang-Johanning*, et al.*, "Advances in Nanotechnology and Microfluidics for Human Papillomavirus Diagnostics," *Proceedings of the IEEE,* vol. 103, pp. 161-178, 2015.

[17] A. Gençtav, S. Aksoy, and S. Önder, "Unsupervised segmentation and classification of cervical cell images," *Pattern Recognition,* vol. 45, pp. 4151-4168, 2012.

[18] M. E. Plissiti and C. Nikou, "Overlapping cell nuclei segmentation using a spatially adaptive active physical model," *IEEE Transactions on Image Processing,* vol. 21, pp. 4568-4580, 2012.

[19] C.-Y. Huang, P.-J. Chen, T.-R. Ger, K.-H. Hu, Y.-H. Peng, P.-W. Fu*, et al.*, "Optimization of Magnetic Labeling Process for Intracellular Hyperthermia in Cervical Cancer Cells," *IEEE Transactions on Magnetics,* vol. 50, pp. 1-4, 2014.

[20] K. Li, Z. Lu, W. Liu, and J. Yin, "Cytoplasm and nucleus segmentation in cervical smear images using Radiating GVF Snake," *Pattern Recognition,* vol. 45, pp. 1255-1264, 2012.

[21] P.-Y. Pai, C.-C. Chang, and Y.-K. Chan, "Nucleus and cytoplast contour detector from a cervical smear image," *Expert Systems with Applications,* vol. 39, pp. 154-161, 2012.

[22] M. E. Plissiti, C. Nikou, and A. Charchanti, "Combining shape, texture and intensity features for cell nuclei extraction in Pap smear images," *Pattern Recognition Letters,* vol. 32, pp. 838-853, 2011.

[23] M. S. Nosrati and G. Hamarneh, "Segmentation of overlapping cervical cells: a variational method with star-shape prior," in *IEEE 12th International Symposium on Biomedical Imaging (ISBI), 2015* 2015, pp. 186-189.

[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, pp. 2274-2282, 2012.

[25] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, "Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features," *IEEE Transactions on Medical Imaging,* vol. 31, pp. 474-486, 2012.

Fuzzy Segmentation of MR Brain Real Images Using Modalities Fusion

ASSAS Ouarda *
Department of Computer Science,
Laboratory Pure and Applied Mathematics (LPAM)
University of M'sila
M'sila, Algeria
e-mail: assas_warda@yahoo.fr

***Abstract*— With the development of acquisition image techniques, more data coming from different sources of image become available. Multi-modality image fusion seeks to combine information from different images to obtain more inferences than can be derived from a single modality. The main aim of this work is to improve cerebral IRM real images segmentation by fusion of modalities (T1, T2 and DP) using Fuzzy C-Means approach (FCM). The evaluation of adopted approaches was compared using four criteria which are: the standard deviation (STD), entropy of information (IE), the coefficient of correlation (CC) and the space frequency (SF). The experimental results on MR brain real images prove that the adopted scenarios of fusion approaches are more accurate and robust than the standard FCM approach**

Keywords-component; Data fusion, Segmentation, Fuzzy C-Means, MR images.

I. INTRODUCTION

In last decades, biomedical and medical image processing have become one of the most challenging fields of image processing and pattern recognition. Brain segmentation consists of separating the different tissues: gray matter (GM), white matter (WM), cerebrospinal fluid (CSF) and probably abnormal (tumor) tissue.

The aim of segmentation of MR Brain images is to: Study anatomical structure, Identify region of interest: locate tumor, …, others abnormalities, measure tissue size (to follow the evolution of tumor) and help in treatment planning prior to radiation therapy(radiation dose calculation).

However, the segmentation of MR Brain images has remained a challenge in image segmentation. And this is due to partial volume effects, motion (patient movement, blood circulation and respiration), the existence of image noise, the presence of smoothly varying intensity in-homogeneity, the fact that different anatomical structures may share the same tissue contrast and large amounts of data to be processed. For these and others many approaches have been studied, including Methods based edge [1][2][3], methods based region [4][5], Methods based on thresholding [6][7], methods based artificial neural networks [8], data fusion methods [9], Markov random field methods [10] and hybrid Methods [11][12][13].

In fuzzy segmentation, the image pixel values can belong to more than one segment, and associated with each of the points are membership grades that indicate the degree to which the data points belong to different segments.

Segmentation process also helps to find region of interest in a particular image. The main goal is to make image more simple and meaningful. Fuzzy C-Means (FCM) is a unsupervised fuzzy classification algorithm. Resulting from the C-means algorithm (C-means), it introduces the concept of fuzzy set in the class definition: each point in the data set for each cluster with a certain degree, and all clusters are characterized by their center of gravity.

The data fusion, in imaging, is used mainly on radar images, satellite images, and aerial images. Recently, it is also applied in medical image. The increasing diversity of the medical image acquisition techniques motivated recent years much research aimed at developing models increasingly effective data fusion. Indeed, medical imaging, it may happen that no images available alone does not contain sufficient information. On the other hand the medical community entrusts each image type to an expert who has a partial diagnosis of the modality of his specialty and specialists exchange experiences and this confrontation comes the final diagnosis.

In this paper, our contribution is mainly to propose an architecture of a information fusion system guided by the prior knowledge and based on Fuzzy C-Means approach to segment human MR real Brain images.

The organisation of the paper is as follows. In section 2 the Fuzzy C-Means approach of segmentation is reviewed and in section 3 describes briefly data fusion. Section 4 present a complete description of proposed segmenting approach using data fusion, where each step of the algorithm is developed in detail. Section 5 illustrates the obtained experimental results and discussions and section 6 concludes this paper.

II. FUZZY C-MEANS TECHNIQUE [14]

Modeling inaccuracy is done by considering gradual boundaries instead of clear borders between classes. The uncertainty is expressed by the fact that a pixel has attributes that assign a class than another. So, Fuzzy clustering assigns not a pixel a label on a single class, but its degree of membership in each class. These values indicate the uncertainty of a pixel belonging to a region and are called membership degrees. The membership degree s in the interval [0, 1] and the obtained classes are not necessarily disjoint. In this case, the data Xj are not assigned to a single class, but many through degrees of membership Uij of the vector Xj to class i. The purpose of classification algorithms is not only calculating cluster centers *bi* but all degrees of membership vectors to classes. If Uij is the membership degree of Xj to class i, the matrix U(CxN, C number of cluster and N is the data size) is called fuzzy C-partitions matrix if and only if it satisfies the conditions (1) and (2):

$$\forall i \in [1, C], \forall j \in [1, N] \begin{cases} u_{ij} \in [0,1] \\ 0 \prec \sum_{j=1}^{N} u_{ij} \prec N \end{cases} \quad (1)$$

$$\forall i \in [1, C] \sum_{i=1}^{C} u_{ij} = 1 \quad (2)$$

The objective function to minimize J and the solutions bi, Uij, of the problem of the FCM are described by the following formulas:

$$J(B,U,X) = \sum_{i=1}^{C}\sum_{j=1}^{N}(U_{ij})^{m} d^{2}(x_{j}, b_{i}) \quad (3)$$

$$bi = \frac{\sum_{j=1}^{N}(u_{ij})^{m}.X_{j}}{\sum_{j=1}^{N}(u_{ij})^{m}} \quad (4)$$

$$u_{ij} = \left[\sum_{k=1}^{C}\left(\frac{d^{2}(X_{j}, b_{i})}{d^{2}(X_{j}, b_{k})}\right)^{\frac{2}{(m-1)}}\right]^{-1} \quad (5)$$

with the variable m is the fuzzification coefficient which takes values in the interval [0, + ∞ [. The FCM algorithm stops when the partition becomes stable.

Like other unsupervised classification algorithms, it uses a criterion minimization of intra-class distances and maximizing inter-class distances, but gives a degree of membership of each class for each pixel. This algorithm requires prior knowledge of the number of clusters and generates classes through an iterative process by minimizing an objective function. Thus, it allows to obtaining a fuzzy partition of the image by providing each pixel with a membership degree (between 0 and 1) to a given class. The cluster which is associated with a pixel is one whose degree of membership is the highest.

The main steps of the Fuzzy C-means algorithm are:

1. Input the image Xj: j=1..N, N: size of image.
2. Set the parameters of the algorithm: C: number of cluster, m: fuzzy coefficient, ε: convergence error.
3. Initialize the membership matrix U with random values in the range [0,1].
4. Update the centers *bi* using the equation (4) and evaluation of the objective function J*old* using the formula (3).
5. Update the membership matrix U using the equation (5) and evaluation of the objective function J*new* using the formula (3).
6. Repeat steps 4 and 5 until satisfaction of the stopping criterion which is written: || J*old*-J*new*:||≤ε.
7. The outputs are the membership matrix U and the centers *bi.*

III. DATA FUSION

Information fusion is to combine information (often imperfect and heterogeneous) from multiple sources to obtain better complete global information, to improve decision making and make better act. The terms "information" (numeric or symbolic) and "sources" cover many possibilities. In the same way, the notion of improvement depends wholly on the application.

Information fusion has evolved considerably in recent years in various fields, especially in vision and robotics, information sources have increased (sensors, a priori information, generic knowledge ... etc.). In general, each source of information is imperfect, it is important to combine several to get a better understanding of the all of the system. MRI is a powerful tool to improve clinical diagnosis because it can provide various information in the form of image intensities related to the anatomy through a variety of excitation sequences (for example: T1, T2, and PD).

The proposed fusion involves the aggregation MR images from different acquisition techniques. Data to be combined are so homogeneous, and depending on the type of image acquisition will provide more or less pronounced contrast between tissues or between parenchyma and pathology. One of the main interests of the fusion will be to exploit in particular the complementarity between the different images. Many applications can benefit from this technique. These include:

The detection of tumour regions:

MRI provides easy assess tumour extension, especially when contrast media are used. With certain acquisition techniques, the specificity is also greater in some cases to distinguish between tumor and oedema. The whole point is going to reside in a combination of these techniques with a more anatomical acquisition (weighted T1 type) to measure the tumor extension.

Quantification of brain tissue volumes

Because of its anatomical accuracy and variety of acquisition techniques, MRI is used to assess the distribution of different brain tissues following several contrasts. The volume quantification of these tissues is clinically fundamental to the study of many pathologies that affect the white matter, gray matter or cerebrospinal fluid, or simply for the measurement of volumes in healthy subjects.

Information fusion can be doing at three conceptual levels corresponding to three types of information:

•Data fusion: it is essentially to marry low-level information such as primitives, in order to make information less noisy than that obtained with a single source of information.

•Decision fusion: it performs the combination of sophisticated information (numeric or symbolic) that can be considered as proposals for a decision.

•Models fusion: in this case, different approaches are set apart to fill imperfections affecting each of them independently.

A general information fusion problem can be stated in the following terms : given L sources S1, S2,…SL representing heterogeneous data on the observed phenomenon, take a decision di on an element x, where x is higher level object extracted from information, and Di belongs to a decision space D={d1, d2, d3,…,dn} (or set of hypotheses). In numerical fusion methods, the information relating x to each possible decision di according to each source Sj is represented as a number Mij having different properties and different meanings depending on the mathematical fusion framework. In the centralized scheme, the measures related to each possible decision i and provided by all sources are combined in a global evaluation of this decision, taking the form, for each i : Mi = F(Mi1, Mi2, Mi3, …, Min), where F is a fusion operator. Then a decision is taken from the set of Mi, $1\leq i\leq n$. in this scheme, no intermediate decision is taken and the final decision is issued at the end of the processing chain. In decentralized scheme decisions at intermediate steps are taken with partial information only, which usually require a difficult control or arbitration step to diminish contradictions and conflicts [15][16].

The three-steps fusion can be therefore described as:

Modeling of information: in a common theoretical frame to manage vague, ambiguous knowledge and information imperfection. In addition, in this step the Mij values are estimated according to the chosen mathematical framework.

Combination: the information is then aggregated with a fusion operator F. This operator must affirm redundancy and manage the complementarities and conflicts.

Decision: it is the ultimate step of the fusion, which makes it possible to pass from information provided by the sources to the choice of a decision di.[17]

I. Bloch [18] classified these operators in three classes defined as:

- Context independent and constant behaviour operators (CICB);
- Context independent and variable behaviour operators (CIVB);
- Context dependent operators (CD).

IV. Methodology

Segmentation of brain images can separate different brain structures and detect possible pathologies, namely brain tumors. A good segmentation helps the doctor for making a final decision before his surgery. The main applications of the segmentation are morphometry, functional mapping and surface or volume visualization. Morphometry is the quantitative measurement of the positions, shapes and sizes of brain structures. It requires prior segmentation of these structures, and can identify, understand and follow the progression of diseases such as Alzheimer's or different tumors

The figure 1 shows the implementation of the proposed approach with its various stages:

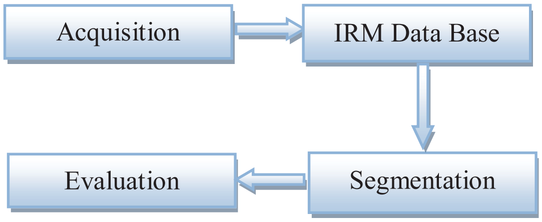


Fig. 1. Diagram of the various steps of the analysis system MRI images.

A. *Acquisition:*

MR Brain images are obtained by a Magnetic resonance imaging (MRI). Examination performed on a machine of high field 1.5 T according to the sequences:

- Axial and Sagittal Tl
- Axial T2 * Flair and diffusion.
- Coronal T2.
- Examination with and without gadolinium injection.

These MRI images are of different sections (axial, sagittal, and coronal) of healthy and pathological subjects. They are grouped into several sections.

B. *IRM DataBase:*

The format of images is the format DICOM (Digital Imaging and Communication in Medicine). This latter is a file used by most of the manufacturers of medical imaging; this standard was issued by the ACR (American College of Radiology) in association with the NEMA (National Electrical Manufacturers Association). The DICOM format is a file containing the image and patient data compressed (patient name, exam type, hospital, examination date, type of acquisition…etc.). To validate our segmentation algorithms, we use a real database. These images are encoded in the DICOM format size 256x256 pixels. These images are grouped into several sections. Each image DICOM used has the following details:

- Format : ‘DICOM’
- Color Type : ‘grayscale’
- Modality : ‘MR’
- Manufacturer: ‘GE MEDICAL SYSTEMS’
- Institution Name: ‘Medical Imaging Center of M'sila (DR S. F. Ghadbane)’
- Study Description: ‘CEREBRAL’
- Series Description: ‘FL:A/3-pl T2* FGRE S’
- Slice Thickness: 5
- Repetition time: 55.500
- Magnetic Field Strength: 15000
- Echo Time: 2.1000
- Spacing Between Slices: 10
- Spatial Resolution: 1.8750
- Flip Angle: 0

- Pixel pacing:[2x1 double]

The modality T1 is used to distinguish the different tissues such as : gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF). However, The T2 modality do not allow to distinguish the GM from WM but highlight lesions and CSF.

C. Segmentation

Normally MR Brain image can be classified in three classes: gray matter (GM), white matter (WM), cerebrospinal fluid (CSF). Each region has a certain gray level, for the T1 modality, the WM region has the gray level which tend to white one, the CSF region has the gray level which tend to black one and the grey level of GM region is between the both. The process of segmentation is done with FCM using modalities fusion to separating the different tissues of MR Brain images.

For our MR images fusion, context-based conjunctive operators are chosen because in the medical context, both images were supposed to be almost everywhere concordant, except near boundaries between tissues and in pathologic areas. In addition, the context-based behaviour allowed to taking into account these ambiguous but diagnosis–relevant areas. Then we retained four operators of this class, three of them are introduced in [18][19][20]:

$$\text{OP1: } \pi_T(\text{v}) = \min\left(\pi_T{}^{T1}(\text{v}), \pi_T{}^{T2}(\text{v})\right) + 1\text{-h} \quad (4)$$

$$\text{OP2: } \pi_T(\text{v}) = \max\left(\frac{\min\ (\pi_T{}^{T1}(\text{v}),\pi_T{}^{T2}(\text{v}))}{h}, 1\text{-h}\right) \quad (5)$$

$$\text{OP3: } \pi_T(\text{v}) = \min\left(1, \frac{\min\ (\pi_T{}^{T1}(\text{v}),\pi_T{}^{T2}(\text{v}))}{h} + 1\text{-h}\right) \quad (6)$$

with :

$$\text{h}=1\text{-}\sum_{v \epsilon Image} |\pi_T{}^{T1}(\text{v}) - \pi_T{}^{T2}(\text{v})|/|\text{Image}| \quad (7)$$

$$\text{OP4: } \pi_T(\text{v}) = \frac{\pi_T{}^{T1}(\text{v}),\pi_T{}^{T2}(\text{v}))}{2} \quad (8)$$

The general algorithm for the FCM using modalities fusion approach can be formulated as follows:

1. Set the parameters of the algorithm: C: number of cluster, ε: convergence error.
2. For each image of section j Xj: j=1...SC, which SC is the section number.
3. Segmentation of each image section of each modality (T1, T2, PD) using FCM provide posteriori-probabilities membership Rjt, t=(T1, T2, PD), which t is the modality.
4. Use one operator fusion OPi(t1,t2) then the output is membership matrix Rj,.
5. Assign all pixels to clusters by using the maximum membership value of every pixel.

D. Evaluation :

In addition to visual analysis, a quantitative evaluation is used on the above experimental results by different fusion algorithms. The selected quantitative criterions are standard deviation (SD), entropy (EN), spatial frequency (SF) and coefficient correlations (CC).

Standard deviation (SD): standard deviation is the square root of the variance, the variance of an image reflects the degree of dispersion among the grayscale values and the average value of gray levels. The larger the value is, the better fusion results are obtained.

$$SD = \sqrt{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} F(i,j)}{M}} \quad (9)$$

Entropy (EN): Entropy can effectively reflect the amount of information in certain image. The larger the value is, the better fusion results are obtained [21]:

$$EN = \sum_{i=0}^{L-1} P_f(i) log_2 P_f(i) \quad (10)$$

where P_{ui} is the normalized histogram of the fused image to be evaluated, L is the maximum gray level for a pixel in the image.

Spatial frequency (SF): Spatial frequency can be used to measure the overall activity and clarity level of an image. Larger SF value denotes better fusion result [21]:

$$SF = \sqrt{RF^2 + CF^2} \quad (11)$$

With

$$RF = \sqrt{\left(\frac{1}{M(N-1)}\right)\sum_{i=0}^{M-1}\sum_{j=0}^{N-2}(F(i,j+1) - F(i,j))^2} \quad (12)$$

And

$$CF = \sqrt{\left(\frac{1}{N\ (M-1)}\right)\sum_{i=0}^{M-2}\sum_{j=0}^{N-1}(F(i+1,j) - F(i,j))^2} \quad (13)$$

Coefficient correlation (CC): Coefficient correlation can show similarity in the small structures between the original and reconstructed images. Higher value of correlation means that more information is preserved [21]:

$$CC = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}(x_{i,j} - \mu(A))(x'_{i,j} - \mu(B))}{\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}(x_{i,j} - \mu(A))^2(x'_{i,j} - \mu(B))^2}} \quad (14)$$

where μ(A) and μ(B) are the mean value of the corresponding dataset.

V. Experimental Results

The proposed approach Fuzzy c-means using modalities fusion have been tested on real MR brain images to certify their efficiency. It was acquired in medical imaging center of M'sila (DR S. F. Ghadbane). These MRI images are of different sections (axial, sagittal, and coronal) of healthy and pathological subjects of size (256×256). These images are grouped into 26 sections. The format of images is the format DICOM (Digital Imaging and size Communications in Medicine).

Examples of real images MRI present in this paper are extracted on axial sections of three modalities (T1, T2 and PD). The healthy images are segmented in four parts (c=4): background, white matter WM, gray matter GM and cerebrospinal fluid CSF. For tumoral subject, the images are segmented in five classes: background, WM, GM, CSF and the tumor. Standard deviation (SD), entropy (EN), spatial frequency (SF) and coefficient correlations (CC) are used to compare the performance of the adopted techniques for segmentation of MR brain images.

To evaluate the performance of the proposed approach, two slices are presented for each adopted fusion operator and each fusion system. And obtained results are compared to those of FCM. The two slices are: slice 16 from the healthy subject (Figure 2) and slice 22 from the tumor one (Figure 4) with three modalities (T1, T2 and PD). Figure 2, 3, 4 and 5 present for each example the segmented image using the FCM and FCM using system fusion ((T1, T2) (T1, PD), (T2, PD) and (T1, T2, PD)) with each operator(OP1, OP2, OP3 and OP4). Tables I, II, III, IV and V present for each example and for FCM and FCM using system fusion with each operator the detailed quantitative evaluation: (standard deviation (SD), entropy (EN), spatial frequency (SF) and coefficient correlations (CC)).

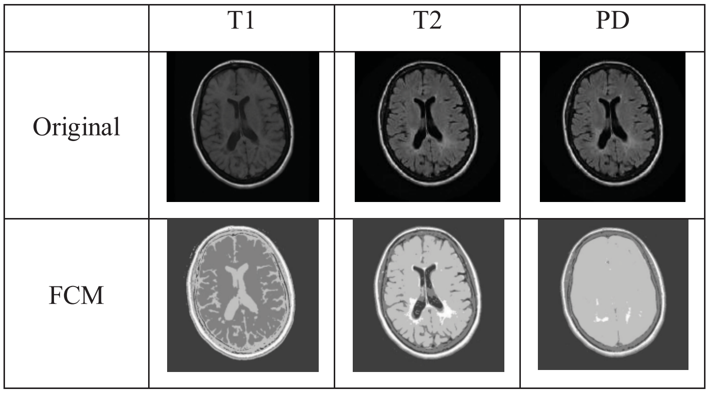
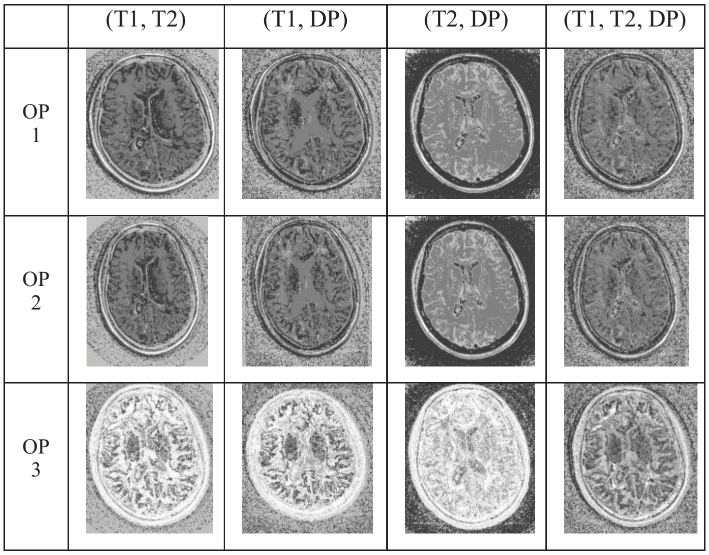


Fig. 2. Slice 16 from healthy MR brain images and corresponding segmented image using FCM algorithm for c=4 .



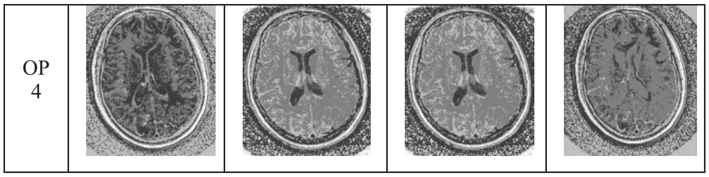


Fig. 3. The segmented results of the Slice 16 from healthy MR brain images using the fuzzy c-means algorithm using the four fusion operateur and the four fusion sytem (c=4).

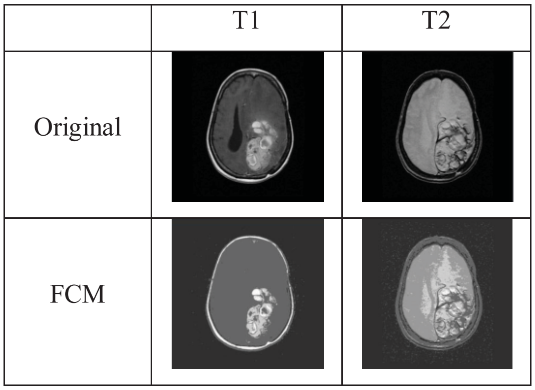


Fig. 4. Slice 22 from tumor MR brain images and corresponding segmented image using FCM algorithm for c=5.

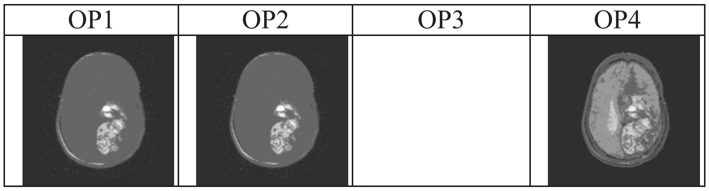


Fig. 5. The segmented results of the Slice 22 from tumor MR brain images using the Fuzzy c-means algorithm using the four fusion operateur and the fusion sytem (T1, T2) with (c=5).

TABLE I. EXPERIMENTAL RESULTS USING FCM FOR THE SLICE 16 FROM HEALTHY MR BRAIN

	CC	EN	SF	STD
T1	-0.718	1.360	7.329	6.868
T2	-0.672	1.291	6.420	7.090
DP	0.259	1.494	10.703	4.889

TABLE II. EXPERIMENTAL RESULTS USING FCM WITH THE FOUR FUSION OPERATOR AND THE FUSION SYSTEM (T1, T2) FOR THE SLICE 16 FROM HEALTHY MR BRAIN

		CC	STD	SF	EN
Op1	T1	-0.329	12.121	58.603	1.560
	T2	-0.297			
Op2	T1	-0.329	12.121	58.603	1.560
	T2	-0.297			
Op3	T1	0.448	13.656	71.660	1.567
	T2	0.410			
Op4	T1	0.040	11.125	61.253	1.592
	T2	0.019			

TABLE III. EXPERIMENTAL RESULTS USING FCM WITH THE FOUR FUSION OPERATOR AND THE FUSION SYSTEM (T1, T2, PD) FOR THE SLICE 16 FROM HEALTHY MR BRAIN

		CC	STD	SF	EN
Op1	T1	0.137	10.948	78.135	1.640
	T2	0.189			
	DP	0.180			
Op2	T1	0.137	10.948	78.135	1.640
	T2	0.189			
	DP	0.180			
Op3	T1	0.398	11.659	82.598	1.700
	T2	0.388			
	DP	0.371			
Op4	T1	0.242	11.00	59.727	1.650
	T2	0.240			
	DP	0.187			

TABLE IV. EXPERIMENTAL RESULTS USING FCM FOR THE SLICE 22 FROM TUMOR MR BRAIN

	CC	EN	SF	STD
T1	0.999	1.368	6.187	3.573
T2	-0.616	1.433	8.575	4.913

Table I and IV show that modality T2 provide the best segmentation. The use of modalities fusion has improved segmentation in terms of evaluation criteria (Table II, III and V). Segmentation by modalities fusion depends on modalities themselves and the used fusion operator: for example for the min operator (OP1) the best combination is T1 with T2.

TABLE V. EXPERIMENTAL RESULTS USING FCM WITH THE FOUR FUSION OPERATOR AND THE FUSION SYSTEM (T1, T2) FOR THE SLICE 22 FROM TUMOR MR BRAIN

		CC	STD	SF	EN
Op1	T1	0.975	30.520	23.047	1.174
	T2	0.997			
Op2	T1	0.975	52.520	17.047	1.174
	T2	0.927			
Op3	T1	0	30.968	0	0
	T2	0			
Op4	T1	0.949	35.796	25.166	1.346
	T2	0.981			

The fusion using the three modalities with the operator min (OP1) offer the best segmentation with a rate of correlation CC= 0.137, 0.189 and 0.180 standard deviation STD= 10.948 spatial frequency SF= 78.135 and information entropy EN= 1.640. The segmentation using the third operator provide white image for the tumour subject and for we fusion T1 with DP and T2 with DP. Fusion operator that has the best performance is the fourth one which confirms the qualitative improvement. As it can be seen, the adopted approach is very good and allows a good segmentation (Figure 3 and 5).

We can see that the main tissues (GM, WM and CSF) of the brain images are well separated for healthy subject and tumor region is well extracted for pathological subject. Fusion modalities and FCM algorithm perform equally well in terms of the quality of image segmentation and leads to a good visual result.

VI. CONCLUSION

Segmentation of medical images is still a vast field of research. The aim of our work is devoted to brain tissue segmentation from magnetic resonance images, in order to extract the tumor and also all other tissues (white matter, gray matter and CSF) by using of Fuzzy c-means with modalities fusion approach. This aggregation was performed by fusion operators that model doctor daily analysis confronted heterogeneous clinical data. The proposed approach Fuzzy c-means using data fusion has been tested on real MR brain images (healthy and pathological) to certify their efficiency. Experimental results show that: modalities fusion improves the segmentation of brain images. The fusion operators min and mean are the best for the segmentation of brain images and can deliver satisfactory performance to separating the different parts of an MR brain real image. Further research is needed to improve the proposed approach. At level of modeling we would like to integrate other numeric or symbolic information to increase the mass of available knowledge and at the combination one to design adaptive fusion operators for the combination of data in the medical field.

ACKNOWLEDGMENT*(Heading 5)*

The authors would like to thank the medical imaging center of M'sila, Algeria, and specially Dr S. F. Ghadbane, for providing the MR Brain images.

REFERENCES

[1] S. He, X. Shen, Y. Yang, R. He and W. Yan, “Research on MRI rain Segmentation Algorithm with the Application in Model –Based EEG/MEG”, IEEE Transactions on Magnetics, 37(5) 3741-3744. 2001.

[2] A. Yezzi, S.Kichenassamy, A. Kumar, P. Olver and A. Tannenbaum, “A geometric snake model for segmentation of medical imagery, Medical Imaging“, IEEE Transactions on, vol.16, no.2, pp.199-209, April 1997. Doi: 10.1109/42.563665.

[3] G. B. Aboutanos, J. Nikanne, N. Watkins and B. Dawant, “Model Creation and Deformation for the Automatic Segmentation of the Brain in MR Images“. IEEE Transactions on Biomedical Engineering, 46(11).1999.

[4] A. Kouhi, H. Seydarabi, A. Aghagolzadeh, “A Modified FCM Algorithm for MRI Brain Image Segmentation”. Machine Vision and Image Processing (MVIP), 2011 (7th Iranian Digital Object Identifier: 10.1109/IranianMVIP.2011.6121551, pp 1–5. 2011

[5] I. Soesanti, A. Susanto, T.S. Widodo, M. Tjokronagoro. “optimized fuzzy logic application for mri brain images segmentation”. International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 5, Oct 2011. pp 137-146. DOI : 10.5121/ijcsit.2011.3512.

[6] P. Kalavathi. “Brain Tissue Segmentation in MR Brain Images using Multiple Otsu's Thresholding Technique”. The 8th International Conference on Computer Science & Education (ICCSE 2013) April 26-28, 2013. Colombo, Sri Lanka. 978-1-4673-4463-0/13/$31.00 ©2013 IEEE 639

[7] C. Cheng; Y. Chen, T. Lin. “FCM Based Automatic Thresholding Algorithm to Segment the Brain MR Image” Machine Learning Machine Learning and Cybernetics, 2007 International Conference on Volume: 3 Digital Object Identifier: 10.1109/ICMLC.2007.4370358, pp: 1371 – 1376. 2007

[8] D.A Karras and B.G. Mertzios. “On Edge Detection in Mri Using the Wavelet Transform and Unsupervised Neural Networks”, EC-VIP-MC 2003. 4th EURASIP Conference focused on Video I Image Processing and Multimedia Communications, 2-5 July 2003, Zagreb, Croatia

[9] L. Gui, R. Lisowski, T. Faundez, P.S. Huppi, F. Lazeyras and M. Kocher. “Automatic Segmentation of Newborn Brain Mri Using

Mathematical morphology”. 978-1-4244-4128-0/11/$25.00 ©2011 IEEE.

[10] J. Tohka, I. D. Dinov, D.W. Shattuck, A.W. Toga. “Brain MRI tissue classification based on local Markov random fields” Magnetic Resonance Imaging, Volume 28, Issue 4, May 2010, pp 557-573

[11] N. Sharma, A. Ray, S. Sharma, K. Shukla, S. Pradhan and L. Aggarwal, "Segmentation and Classification of Medical Images using Texture-Primitive Features: Application of BAM-type Artificial Neural Network", Medical Physicists, vol. 33, pp. 119-126, 2008.

[12] M. Stella and B. Mackiewich, "Fully Automated Hybrid Segmentation of Brain, " Handbook of Medical Imaging: Processing and Analysis Management, I. Bankman, Ed, 2009.

[13] B.S. Anami, P.H. Unki. “A combined fuzzy and level sets based approach for brain MRI image segmentation” Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on Digital Object Identifier: 10.1109/NCVPRIPG.2013.6776216. pp 1 – 4. 2013

[14] O. Assas “mages segmentation based on interval type-2 Fuzzy C-Means” SAI Intelligent Systems Conference (IntelliSys), 2015 10-11 Nov. 2015. IEEE communication pp 773 – 781. DOI: 10.1109/IntelliSys.2015.7361228

[15] Bloch, I., and Maitre, H. 1997. Data fusion in 2D and 3D image processing: an overview. In proceedings of the X Brazilian symposium on computer graphics and image processing, Brazil, 127-134.

[16] Barra, V. and Boire, J. Y. 2001. A general framework for the fusion of anatomical and functional medical images. NeuroImage 13, 410-424.

[17] Lamiche C. Moussaoui A. Segmentation of MR Brain Images using a Data Fusion Approach International Journal of Computer Applications (0975 – 8887) Volume 36– No.12, December 2011

[18] Bloch, I. 1996. Information combination operators for data fusion : a comparative review with classification. IEEE Transactions in systems, Man. and Cybernitics 1, 52-67.

[19] Dubois, D. and Prade, H. 1992. Combination of information in the framework of possibility theory. In Data Fusion in Robotics and Machine Intelligence, M. AL Abidi et al.

[20] Barra, V. 2000. Fusion d'Images 3D du Cerveau : Etude de Modèles et Applications. Thèse de doctorat, Université d’Auvergne.

[21] L. Yang, B.L. Guo, W. Ni, 2008, Multimodality medical image fusion based on multiscale geometric analysis of contourlet transform, Neurocomputing 72, P 203-211

[22]

Window Width Value Estimation Technique for CT Brain Images Using Average of Median of Statistical Central Moments

C. S. Ee[1], **K. S. Sim**[1], **N. Koh**[1]

[1] Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450,
Melaka, Malaysia

Abstract – *For stroke detection, computed tomography (CT) scan is always the initial choice for imaging the damages or infraction on the brain. However, CT is commonly poor in infarction diagnosis due possible problems of the proper window settings. There is similar default window setting for every CT brain images but the default setting is unable to fully enhance the contrast of infarction of the brain images. Thus, performance of infarction diagnosis in CT brain images is poorer than other medical image modalities. Therefore, this paper introduces a novel estimation method with fixed value of window center (WC) to estimate the window width value (WW) for selected CT brain images, by calculating average of median of statistical central moments. This method requires only 101 WW values to produce estimated value with sensitivity of 0.05HU. The focus of the proposed approach is to improve the efficiency brain infarction diagnosis for radiologists.*

Keywords: Window Width, Estimation Technique, CT Brain Image, Statistical Central Moments, Median, Average.

1 Introduction

As stated by the World Health Organization (WHO), 15 million people suffered stroke globally every year. From this statistic, 5 million people die while another 5 million people suffered permanent disability. Stroke is a cerebrovascular accident when the blood supply to an area of the brain is cut off [1]. In order to support the stroke diagnosis, two common image modalities have been used namely the computed thermography (CT) scan and magnetic resonance imaging (MRI). CT scan is preferable compared to MRI due to its wider availability, inexpensive and ease of access [2]. Examination of brain images have been a vital task and have received much attention in the literature. Past or present, analysis of brain stroke lesions is difficult especially for inexperienced radiologists or doctors. In CT images, the stroke lesions appear with darker or hypodense region.

In CT imaging, the images are in Digital Imaging and Communications in Medicine (DICOM) format. The DICOM image is in a 16 bit format where 12-bits are used for storing the image without any contrast enhancement or image pre-processing and 4-bits are used to store the textual data [3]. During the examination of the CT brain images, the first thing that the radiologist does is to set the correct window settings of the CT images as different window settings produce different tissue information including the brain lesions. The window settings consist of the window width and window center level plays an important role for stroke lesion detection and diagnosis accuracy. Window width is defined as the display range while window center is defined as the mid value of an image. Although, there are common window setting which consists of window width of 80 HU and window center of 40 HU proposed and used, the output brain images may still have low contrast and the lesion area might not appear more obviously. In image processing and computing system, HU values for each pixels of selected CT brain image is converted as pixel value before processed, and respective equation is shown in equation (1) [4,5].

$$PX = \frac{HU - RI}{RS} \tag{1}$$

where *PX* is the pixel value; *HU* is the Hounsfield unit, HU; *RI* is the rescale intercept; and *RS* is rescale slope. Both *RI* and *RS* can be found in the textual information of CT brain image.

There were many window settings proposes in the past namely window width of 40 HU and window center of 30 HU; window width of 3 HU and window center of 25 HU [6]. Gadda (2002), proposed with window width of 50 HU and window center of 45 HU to 50 HU for good contrast [7]. In 2014, researchers proposed window setting with window center of 40 HU and window width of 50 HU to 60 HU [8]. Although these parameters can show some improvement on the contrast for diagnosis of stroke cases, it is still image dependent and need to be manually tuned. Thus, in this paper a new estimation on window width is proposed. This paper aims to improve prior method in [8], and proposes a new estimation technique to estimate window width value (*WW*) automatically based on the statistical central moments.

2 Problem Statements

The default setting of window setting stored in textual information of CT brain images for visualization is set with *WC*=40 HU and *WW*=80 HU. However, this setting is poor in evaluation of infarction, and not suitable for every CT brain images due to the dynamic differences in the terms of size and volume of each brain.

Another problem is that brain is scanned with CT device into many slices, the estimated window setting for each slice should be different. Using default window setting to these slices may cause misinterpretation in infarction evaluation. Besides that, it is not realistic and time consuming to suggest that the expert radiologists to tune the value of window settings manually, based on their experience and experiments.

Furthermore, the prior methods do not provide any estimation method for window setting for CT brain images. They are required to be set up manually and the values of window setting are within a narrow range. These methods perform well only in certain infarction cases and brain slices.

Therefore the focus of this paper is to find estimation value for window width (*WW*), while value of *WC* is fixed as 40HU, similar with default *WC*. 40HU determines the central region of brain soft tissue, which is also the region of interest (*ROI*) in the paper.

3 Solutions

In this section, estimation method for window width (WW) using average of median of statistical moments is proposed and discussed.

3.1 Flow Chart of Proposed Method

In this subsection, a flow chart of the proposed method is shown in Figure 1. The proposed method contains three main steps, starting with calculating values of statistical central moments of input image (I_{in}) and plotting the graph of statistical central moments versus WW. Next step is to find the average median value of statistical central moments from generated graphs, and then estimate value of *WW*. The last step is to implement the estimated WW to input image (I_{in}) to generate output image (I_{out}).

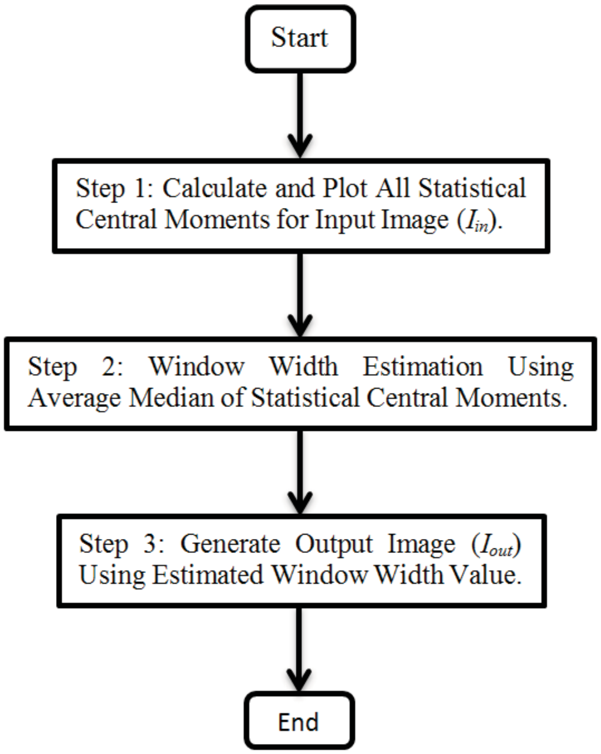

Start
Step 1: Calculate and Plot All Statistical Central Moments for Input Image (I_{in}).
Step 2: Window Width Estimation Using Average Median of Statistical Central Moments.
Step 3: Generate Output Image (I_{out}) Using Estimated Window Width Value.
End

Figure 1: Flow chart of proposed method

3.2 Step 1: Calculate and Plot All Statistical Central Moments for Input Image (I_{in}).

The first step shows the process of generating 4 graphs of statistical central moments versus window width (WW). The process starts to determine the setting of the range of samples for WW. In this paper, WW is set to have p range, from 0 HU to 100 HU, with increment rate of 1 HU. Therefore, there are 101 samples as the x-axis for all 4 graphs and respective formula is shown in equation (2).

$$WW(p) = \begin{cases} 0HU & ,p = 0 \\ WW(p-1) + 1HU, & 0 < p \leq 100 \end{cases} \quad (2)$$

Given that selected CT brain image is the input image (I_{in}), and it is converted into 101 greyscale images, with vector of WW, using equation (3).

$$g(p)_{x,y} = \begin{cases} 0 & ,I_{in}(x,y) < gmin(p) \\ \frac{I_{in}(x,y) - gmin(p)}{WW(p)} \times 255 & ,gmin(p) \leq I_{in}(x,y) \leq gmax(p) \\ 255 & ,I_{in}(x,y) > gmax(p) \end{cases} \quad (3)$$

where $I_{in}(x,y)$ is pixel value of input image (I_{in}) at location (*x*,*y*); $WW(p)$ is the *p*th WW value in vector WW; $g(p)$ is the greyscale image after windowing with $WW(p)$; $gmax(p)$ is maximum window value in HU unit of $WW(p)$ and illustrated in equation (4); $gmin(p)$ is the minimum window value in HU unit of $WW(p)$ and equation (5) illustrates respective formula.

$$gmax(p) = WC + \frac{ww(p)}{2} \tag{4}$$

$$gmin(p) = WC - \frac{ww(p)}{2} \tag{5}$$

The following step is to calculate the values of statistical central moments of all 101 generated greyscale images ($g(p)$). There are 4 elements of statistical central moments. These elements include mean, variance, skewness and kurtosis. Table 1 shows the basic description of these elements, and respective basic equations can be found in [8] for more information.

Table 1: Summary of Statistical Central Moments

Statistical Central Moments	**Other Definition**	**Description**
1st Moment	Mean	Measures Expected Value
2nd Moment	Variance	Measures Spread Dispersion
3rd Moment	Skewness	Measures Asymmetry
4th Moment	Kurtosis	Measures Peakedness

In order to implement these statistical moments with image $g(p)$, equations formulated in [8] are edited and shown in equations (6-9).

$$\mu_g(p) = \frac{1}{X \times Y} \sum_{\substack{0 \le x \le X-1 \\ 0 \le y \le Y-1}} g(p)_{x,y} \tag{6}$$

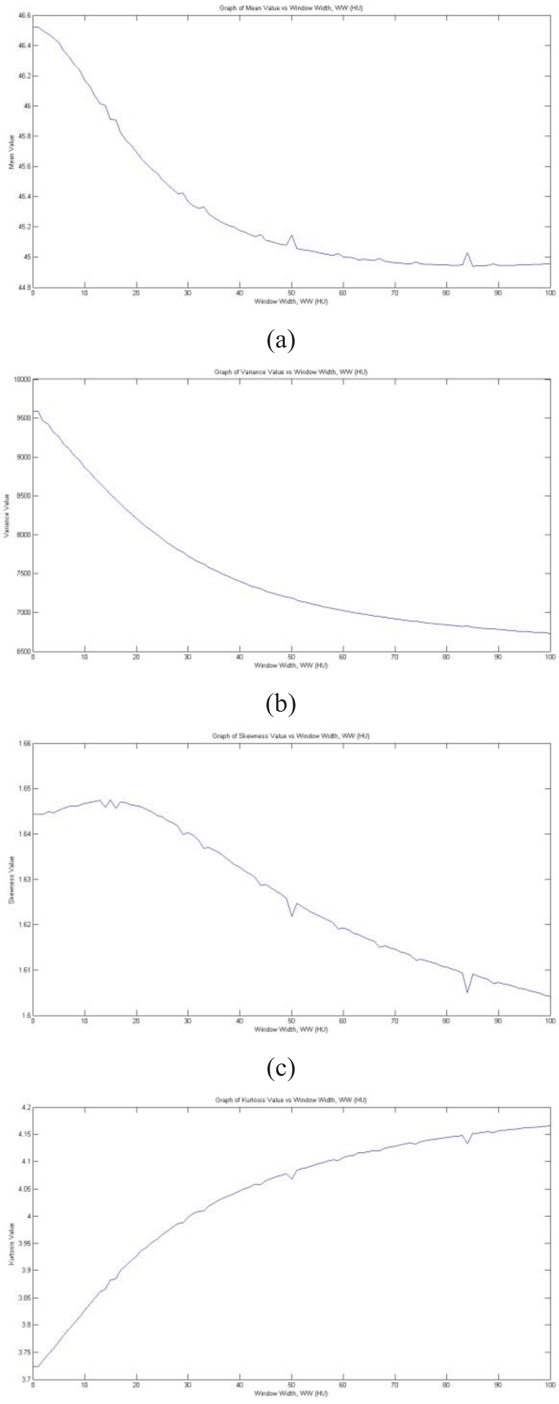
$$\sigma_g{}^2(p) = \frac{1}{X \times Y} \sum_{\substack{0 \le x \le X-1 \\ 0 \le y \le Y-1}} \left[g(p)_{x,y} - \mu_g(p) \right]^2 \tag{7}$$

$$\gamma_g(p) = \frac{1}{X \times Y} \sum_{\substack{0 \le x \le X-1 \\ 0 \le y \le Y-1}} \left[g(p)_{x,y} - \mu_g(p) \right]^3 \tag{8}$$

$$Kurt_g(p) = \frac{1}{X \times Y} \sum_{\substack{0 \le x \le X-1 \\ 0 \le y \le Y-1}} \left[g(p)_{x,y} - \mu_g(p) \right]^4 \tag{9}$$

Where $g(p)_{x,y}$ is the pixel value of image $g(p)$ at position of (*x*,*y*); X is total number of rows of image $g(p)$; Y is the total number of columns of image $g(p)$; $\mu_g(p)$ is the value of mean of image $g(p)$; $\sigma_g{}^2(p)$ is the variance value of image $g(p)$; $\gamma_g(p)$ is the skewness value of image $g(p)$; $Kurt_g(p)$ is the kurtosis value of image $g(p)$.

Then, equations (6-9) are implemented to image $g(p)$ to calculate values of vector of mean (μ_g), variance ($\sigma_g{}^2$), skewness (γ_g), and kurtosis ($Kurt_g$). After that, graph of these vectors versus WW are plotted, and shown in Figure 2. In total, each statistical central moment produces a vector with the range of p, which has 101 values.



(a)

(b)

(c)

(d)

Figure 2: Graphs of Statistical Central Moments vs Window Width, WW (HU): (a) Mean (μ_g), (b) Variance ($\sigma_g{}^2$), (c) Skewness (γ_g), (d) Kurtosis ($Kurt_g$).

3.3 Step 2: Window Width Estimation Using Average Median of Statistical Central Moments.

Following step is to estimate window width value, by calculating the average median of vector of statistical central moments (mean (μ_g), variance (${\sigma_g}^2$), skewness (γ_g), and kurtosis ($Kurt_g$)), based on the graphs in Figure 2. Measurement of median value for each vector is similar. With mean (μ_g) vector as sample, median value for window width can be determined by following mathematical sequences:

i. Sort all the values of mean (μ_g) vector in ascending order to form new mean vector (μ_s).

ii. Calculate the sequence of median of sorted mean vector (μ_s) with equation (10).

$$s_{median} = \frac{S+1}{2} \tag{10}$$

where $S = 101$ is the length of sorted mean vector (μ_s); s_{median} is the sequence of median of sorted mean vector (μ_s).

iii. Then, the median value of sorted mean vector (μ_s) is determined with equation (11).

$$MedianM = \mu_s(s_{median}) \tag{11}$$

iv. After that, using equations (12, 13) are used to find the median sequence of $MedianM$ value from mean (μ_g) vector (p_{median}). Equation (12) determines the vector of differences of $MedianM$ and $\mu_g(p)$, $M(p)$, and equation (13) calculates $Mmin$, the minimum value of vector M. The sequence p of vector M that obtains the value of $Mmin$ is the median sequence, p_{median} for window width (WW) vector.

$$M(p) = \begin{cases} \sum_{p=0}^{P-1}\left[MedianM - \mu_g(p)\right], & MedianM \geq \mu_g(p) \\ \sum_{p=0}^{P-1}\left[\mu_g(p) - MedianM\right], & \mu_g(p) < MedianM \end{cases} \tag{12}$$

$$Mmin = \min\{\textstyle\sum_{p=0}^{P-1} M(p)\} \tag{13}$$

v. Therefore, the median window width value based on mean (μ_g) vector ($WWmedM$), is determined with equation (14).

$$WWmedM = WW(p_{median}) \tag{14}$$

vi. Step i to Step v are repeated for respective vectors of variance (${\sigma_g}^2$), skewness (γ_g), and kurtosis ($Kurt_g$), in order to obtain respective median WW values of $WWmedV$, $WWmedS$, and $WWmedK$. These values are shown in Table 2.

Table 2: Median WW values for vector of mean (μ_g), variance (${\sigma_g}^2$), skewness (γ_g), and kurtosis ($Kurt_g$).

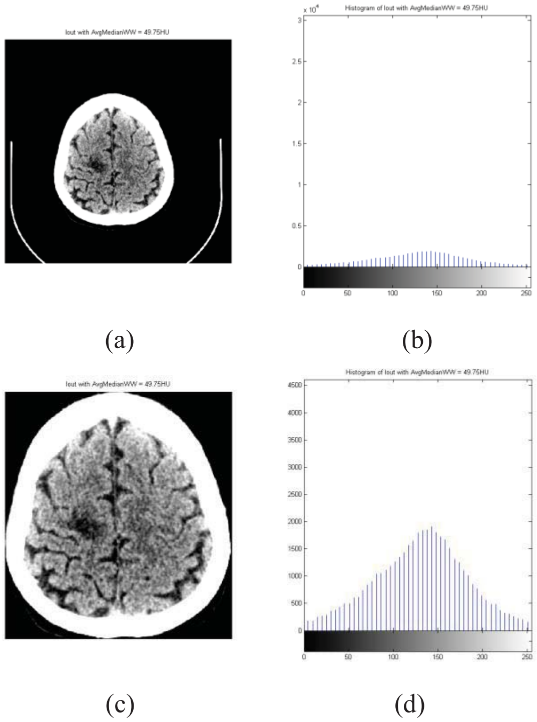
Vector	**WW Values**	**Values (HU)**
Mean (μ_g)	$WWmedM$	49HU
Variance (${\sigma_g}^2$)	$WWmedV$	50HU
Skewness (γ_g)	$WWmedS$	51HU
Kurtosis ($Kurt_g$)	$WWmedK$	49HU

The final part in this subsection is implementing equation (15) to determine the average value of $WWmedM$, $WWmedV$, $WWmedS$, and $WWmedK$. The average value, WWo is the estimated value for window width (WW). Based on Table 2 and equation (15), the output value of WWo is 49.75HU with sensitivity of 0.05 HU.

$$WWo = \frac{(WWmedM + WWmedV + WWmedS + WWmedK)}{4} \tag{15}$$

3.4 Step 3: Generate Output Image (I_{out}) Using Estimated Window Width Value.

The final step for proposed method is generating the output image (I_{out}) by repeating step 1 and step 2 and replacing $WW(p)$ with WWo. However, for better study on histograms, only brain structure and little background pixels of image I_{out} are manually cropped out. Figure 3 shows the differences of visualization and histograms of image I_{out} and cropped image I_{out} (I_{outc}).


Iout with AvgMedianWW = 49.75HU
x 10^4
Histogram of Iout with AvgMedianWW = 49.75HU
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
Iout with AvgMedianWW = 49.75HU
Histogram of Iout with AvgMedianWW = 49.75HU
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

(a) (b)

(c) (d)

Figure 3: (a) Desired Output Image (I_{out}) and (b) respective Histogram, (c) Cropped Output Image (I_{outc}) and (d) respective Histogram.

4 Results and Discussions

500 different CT brain images with infarctions are chosen to evaluate the performance of proposed method. There are 3 evaluation tests for proposed method. First test is

to compare proposed method with 2 evaluations of expert radiologists, E1 and E2. Figure 4 shows comparison between image I_{outc} and expert diagnoses of E1 and E2.

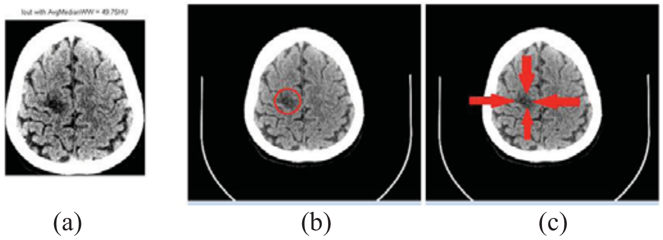


Figure 4: (a) Image Enhanced with Proposed Method (I_{outc}), (b) Evaluation of E1, (c) Evaluation of E2.

The next test is to compare image I_{outc} with grayscale image generated with default WW (I_d) and prior methods. These prior arts are mentioned in introduction section. However, some of window parameters are set within a range; therefore average value of the range will be selected as the estimated value. Table 3 shows the selected windowing parameters for prior techniques and proposed method. The resulting image is shown in Figure 5 and respective histogram is shown in Figure 6.

Table 3: Windowing parameters of proposed method and prior methods.

Approaches	Windowing Parameters	
	WC (HU)	WW (HU)
Default	40 HU	80 HU
Przelaskowski (2005) [6]	30 HU	40 HU
	25 HU	3 HU
Gadda, (2002) [7]	47.5 HU	50 HU
Sim (2014) [8]	40 HU	55 HU
Proposed method	40 HU	WWo

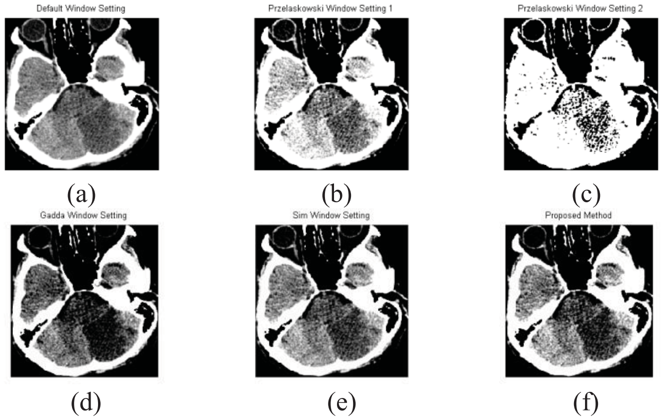


Figure 5: Image Produced by Window Settings of (a) Default, (b) and (c) Przelaskowski (2005) [6], (d) Gadda, (2002) [7], (e) Sim (2014) [8], and (f) Proposed Method.

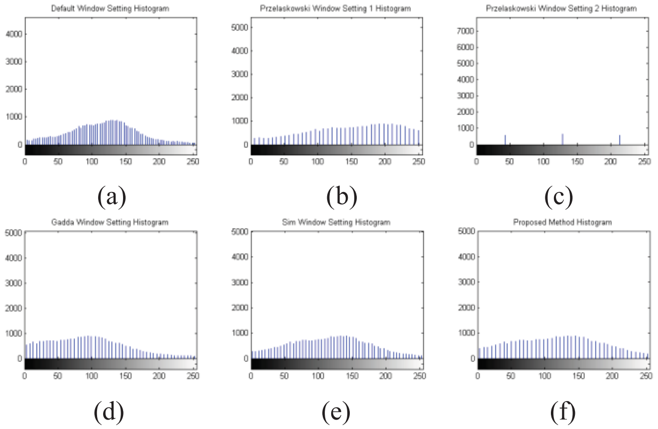


Figure 6: Histogram Produced by Window Settings of (a) Default, (b) and (c) Przelaskowski et al.(2005) [6], (d) Gadda, (2002) [7], (e) Sim et al.(2014) [8], and (f) Proposed Method.

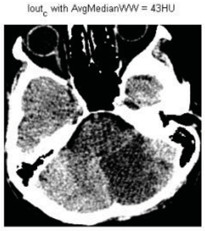
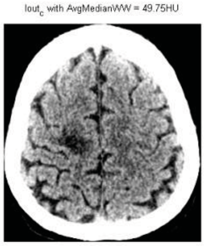
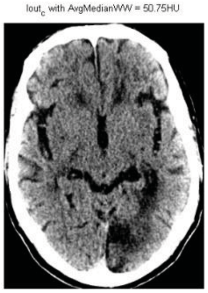
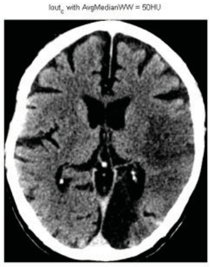
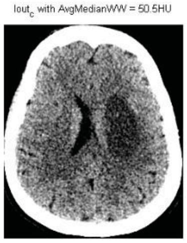
Figure 5 and Figure 6 show the output images and respective histograms produced by all the approaches in Table 3. Przelaskowski's window setting [6] with WC = 25 HU and WW = 3HU, produces output image with over enhancement problem. The healthy brain tissue and parts of the infarctions is washed out. Thus, this method is unadvisable to be implemented for infarction diagnoses.

While for Gadda's window setting [7], it produces output image with under enhancement problem and introduces unwanted artifacts. So, this method will cause misinterpretation by assuming some healthy tissue as brain infarction. Next, the output image produced by Przelaskowski window setting [6] with WC = 30 HU and WW = 40HU, is slightly washed out. This may causes misinterpretation by assuming some brain infarction as healthy brain tissue. Therefore, this window setting is also not recommended for infarction diagnoses.

Sim's window setting [8] and proposed method window setting produce better output image when compared with default window setting. This method is highly enhance infarction region and slightly brighter the normal brain tissue. Thus, the visibility of infarction is better than other methods. However, proposed method has two advantages over Sim's window setting [8]. The first advantage is that it improves the visibility of infarctions better than Sim's method [8]. Another advantage is the value of WWo that is image dependent the range of WWo is not fixed.

The last measurement of visualization is to determine the changes of WWo value in different slices of CT brain images. Table 4 shows five CT brain images at different parts of brain soft tissue area. This table proves the robustness of our proposed method which is able to estimate window width value (WWo), based on any CT brain images.

Table 4: WWo value in HU unit for five different CT brain images with infarctions.

No.	CT Brain Image	WWo
A		43 HU
B		49.75 HU
C		50.75 HU
D		50 HU
E		50.5 HU

5 Conclusions

Based on the results of experiments and observations on 500 CT brain images, our proposed method has proven that it provides better window setting for infarction evaluation than other existing methods. Existing methods are required to be identified manually, but our proposed method is able to estimate the value for window width (WWo) by itself. Furthermore, the sensitivity of WWo of proposed method is 0.05 HU and only requires a vector of 101 values from 0HU to 100 HU to estimate $WWo. WWo$ produced by proposed method is able to provide a good reference for those fresh radiologists who have low experience in infarction diagnosis. Finally, our technique helps to speeds up the duration of infarction diagnosis by expert radiologists.

6 References

[1] R. S. Jeena and S. Kumar. “A comparative analysis of MRI and CT brain images for stroke diagnosis”; Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy (AICERA/ICMiCR), 2013 Annual International Conference on (IEEE), 1—5, 2013.

[2] L. Contin, C. Beer, M. Bynevelt, H. Wittsack, G. Garrido. “Semi-automatic segmentation of core and penumbra regions in acute ischemic stroke: preliminary results”; IWSSIP International Conference, 2010.

[3] M. H. Lev, J. Farkas, J. J. Gemmete, S. T. Hossain, G. J. Hunter, W. J. Koroshetz, and R. G. Gonzalez. “Acute stroke: improved nonenhanced CT detection-benefits of soft-copy interpretation by using variable window width and center level settings”; Radiology, vol. 213 no. 1, 150—155, 1999.

[4] B. Liu, M. Zhu, Z. Zhang, C. Yin, Z. Liu, and J. Gu. “Medical image conversion with DICOM”; Canadian Conference on Electrical and Computer Engineering (IEEE), 36 — 39, April 2007.

[5] National Electrical Manufacturers Association (NEMA). “Digital imaging and communications in medicine (DICOM), Part 3: information object definitions”; (PS 3.3-2011). Virginia: NEMA, 2011b.

[6] A. Przelaskowski, J. Walecki, K. Szerewicz and P. Bargiel. “Acute Stroke Detection in Unenhanced CT Exams: Perception Enhancement by Multi-Scale Approach”; National Conference on Physics and Engineering in The Present Medicine and Health Care the Challenges to Poland as a New European Union Member, 94 — 95, 2005.

[7] D. Gadda, L. Vannucchi, F. Niccolai, A. T. Neri, L. Carmigani, and P. Pacini. “CT in Acute Stroke: Improved Detection of Dense Intracranial Arteries by Varying Window Parameters and Performing a Thin-

Slice Helical Scan”; Neuroradiology, vol. 44, no. 11, 900 — 906, 2002.

[8] K. S. Sim, M. E. Nia, C. S. Ta, C. P. Tso, T. K. Kho and C. S. Ee. "Chapter 32 - Evaluation of Window Parameters of CT Brain Images with Statistical Central Moments"; Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and System Biology, 493 — 503, 2016.

SESSION

COMPRESSION METHODS

Chair(s)

TBA

Lossless Image Compression using Zipper Transformation

Babajide O. Ayinde[1] **and Ahmed H. Desoky**[2]
[1]ECE Department, [2]CECS Department, University of Louisville, Louisville, KY 40218.
{babajide.ayinde, ahmed.desoky}@louisville.edu

Abstract— *This paper proposes a lossless compression scheme for greyscale images using Zipper Transformation (ZT) and Inverse Zipper Transformation (iZT). The proposed transformation exploits the conjugate symmetry property of DFT. We benchmark the proposed ZT with both Discrete Cosine Transformation (DCT) and Fast Walsh Hadamard Transformation (FWHT) in order to quantify the efficacy of the proposed transformation. Numerical simulations show that ZT-based compression algorithm is lossless and gives a faster implementation than its counterparts. The experiments we performed for different block sizes also reveal that the ZT-based algorithm outperforms the FWHT-based algorithm in terms of how much they losslessly compress the original image.*

1. Introduction

Lossless image compression has been found useful in many real world applications such as biomedical image analysis, art images, security and defense, remote sensing, just to mention a few [1]. In most medical applications, images are acquired and stored digitally. This is mostly true in radiology application where the images are grayscale. The images are always very large in size and number, and any means to losslessly compress them can lead to the reduction of the storage cost and improvement in the speed of transmission. Even though the cost of storage and transmission of digital signal has plummeted, however, the demand for lossless compression of medical images is exponentially increasing [2].

In a general sense, image compression can be divided into two main categories: lossy and lossless compression. Lossy compression deals with compression schemes that tolerate some certain amount of error, that is, the compressed and the decompressed images are not perfectly identical. In contrast, lossless compression encodes all the information from the original image and therefore, the decompressed image is exactly the same with the original image.

Lately, the sophistication and complexity of compression paradigms are also increasing with increasing speed of computing. It must be remarked that the cost of compression must be accounted for and considered accordingly. The cost of implementing and deploying a compression scheme is proportional to its complexity. One way to mitigate the complexity is by deploying proprietary methods. However, these methods are detrimental and come with a cost, and one of the common problem is inter-operability with existing equipments [3–5]. For the purpose of striking a balance between complexity and cost, a good number of recent proposed methods focused on lossy compression. In lossy compression, information that is not of significant importance is deliberately discarded. In medical imaging, lossy compression can sometimes achieve a minute compression before a good percentage of information is dropped. Better compression can be achieved if some visible losses can be tolerated for the clinical task purposes. In a sense, there are still a lot of controversies as to what the real life applications of lossy compressions are, especially in the medical field. One other approach to lossy compression is the machine learning approach where images are encoded with a sparse feature representations using autoencoders [6–9].

Generally speaking, lossless compression can be categorized into three broad categories, namely: predictive scheme with statistical modeling, transform based coding and finally, dictionary based coding. The predictive deals with using statistical method to evaluate the differences between pixels and their neighbors, and performed context model before coding. Whereas in transform based compression, pixel are transformed using frequency or wavelet transformation before modeling and coding. Dictionary based compression is the third category and it deals with replacing strings of symbols with shorter codes. It must be noted that dictionary based schemes are widely used for text compression [10]. An example of a dictionary based algorithm is the well known ZIP package. Other dictionary based compression algorithms for image data are the Lempel-Ziv-Welsh (LZW), Portable Network Graphics (PNG), Graphics Intercahnge Format, and so on.

Many decades ago, the JPEG image coding standard is two-fold: The first involves lossy image compression using Discrete Cosine Transform (DCT) and entropy; the second fold deals with lossless reconstruction of the original image using both the predictive scheme and entropy coding. Nowadays, lossless compression typically deals with both predictive and statistical modeling. One of the new lossless compression paradigms is the JPEG 2000 scheme which utilizes wavelet transformation. In this paper, we propose a new way to losslessly compress grayscale images and evaluate its performance in comparison with two existing lossless compression paradigms for grayscale images.

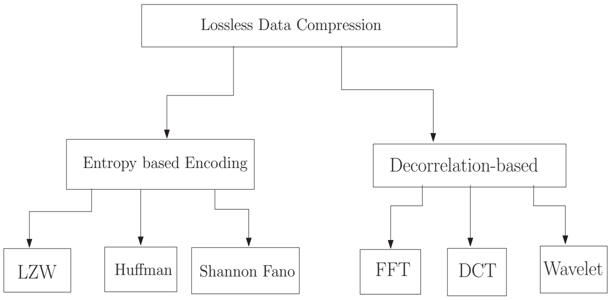


Fig. 1: Types of Lossless Compression

The rest of the paper is organized as follows. Section II gives the state of the art on lossless compression methods. Section III describes our implementation of the zipper, inverse zipper transformation, zipper-based Huffman coding and section IV discusses the experimental designs and presents the results. Finally, conclusions are drawn in section V.

2. Related work

A good number of image data compression paradigms have been investigated in the recent past. The least squares adaptive prediction scheme was proposed in [11] for lossless compression of natural images and the authors show that their novel scheme improves the computational complexity with negligible performance trade-off. Lossless compression that is based on adaptive spectral band re-ordering and adaptive backward previous closest neighbors algorithms (PCN) was also proposed in [12], [13] for hyperspectral images, and it was shown that the compression performance was greatly enhanced with the implementation of both the re-ordering of spectral band. The problem of compression in medical applications has also be dealt with. For instance in [14], different types of compression standards on grayscale medical images were compared by the authors, and the pros and cons of each of the methods were highlighted.

In [1], problems of video compression is addressed taking cognisance of the temporary spectral information. Also in [15], the possibility of using 3-D versions of the lossless JPEG spatial predictors was considered and the likelihood of using best predictor, determined on the basis of the previous frame, to encode the present frame was investigated. The spectral redundancy was also exploited by implementing the best predictor from one spectral component to another spectral component. It can be inferred from the paper that pixels in a given neighborhood are concurrent in adjoining color bands. To improve on this, a different predictor for interband correlation was proposed in [16]. The authors in [17] also proposed a simple context predictive scheme where both intraframe or interframe coding is selected on the basis of temporal and spatial variations, and they then computed the prediction of the current pixel. A good number of predictors are considered taking cognizance of the spatial redundancy.

The Huffman coding technique is a commonly used scheme for data compression because it is very simple and effective. It requires the statistical information about the distribution of the data to be encoded. Besides, an identical coding table is used in both the encoder and the decoder. In [18], a new image lossless compression scheme based on Huffman coding was presented. The scheme is implemented in two stages - the linear predictor stage and the Huffman coding stage. It was shown by the authors that the propose scheme has the capability of reducing the Huffman coding table while improving the compression ratio. Improving the lossless compression of images with sparse histogram is the main concern in [19], and it was shown that the proposed scheme is also robust on other images. Both lossy and lossless compression was investigated in a unified framework and a new cost effective coding strategy was proposed to enhance the coding efficiency. A new motion-JPEG-LS based lossless scheme was also proposed in [2] and the authors only explored high enough correlation between adjacent image frames in order to avoid the possible coding loss and abrupt high computational cost. Also in [20], a wavelet-based lossy to lossless ECG compression was proposed. Lossless digital audio compression scheme was proposed and many other audio compression technologies were also mentioned in [21]. The main contribution of this paper is to propose a new lossless compression for grayscale images and benchmark its performance with DCT and FWHT compression paradigms for grayscale images in term of computational complexity, run time, and compression ratio.

3. Software Description

3.1 Zipper Transformation

In this transformation, we exploited the conjugate symmetry property of DFT. The main idea in this transformation is to first carry out a Discrete fast fourier transform (DFT) on the input array. We then extracted

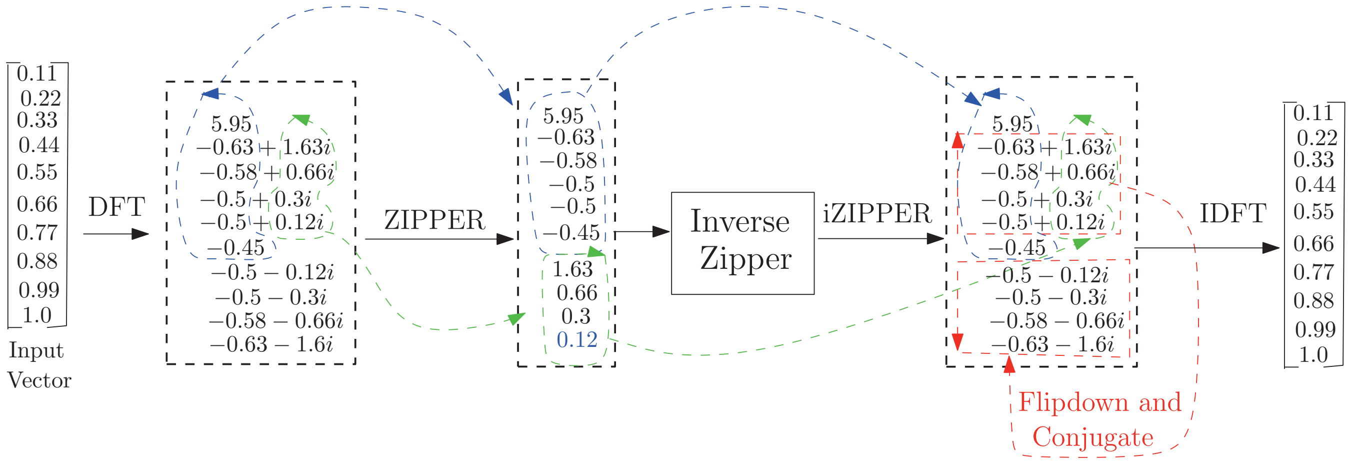


Fig. 2: Illustration of Zipper and Inverse Zipper Transformation

the complex elements of the vector in the upper half of the symmetry. The imaginary part of the complex numbers in the upper half of the symmetry are stripped off and concatenated with their corresponding real counterpart. This procedure is reversed in the inverse zipper transformation. Fig 5 illustrates how the zipper and inverse zipper transformations are implemented using a 10 × 1 vector. It must be remarked that the number of elements in the vector before ZT stays the same after the transformation, that is, ZT preserves the dimensionality of the input data. It must also be noted that this transformation is absolutely lossless. Also, the two-dimensional zipper transformation is implemented by performing ZT on all the columns of the input matrix and then on all the rows.

3.2 Huffman Coding

This method was proposed in 1952 by David Huffman to compress data by reducing the amount of bits necessary to represent a string of symbols. Huffman coding use codewords with variable length, and with shortest codewords for most frequently used characters. The flowchart of the Huffman coding is given below:

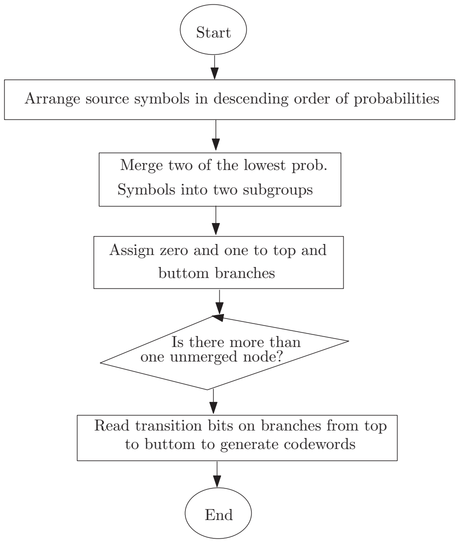

Start
Arrange source symbols in descending order of probabilities
Merge two of the lowest prob. Symbols into two subgroups
Assign zero and one to top and buttom branches
Is there more than one unmerged node?
Read transition bits on branches from top to buttom to generate codewords
End

Fig. 3: Huffman Coding Flowchart

3.3 Zipper-based Huffman Coding

In this section, the pixel intensities of a grayscale image is transformed using the ZT and the output is encoded using the Huffman coding scheme. By virtue of the transformation, the transformed image has a lower entropy, and hence, Huffman scheme is able to encode this stream of data with codewords of variable length. At the receiving end, Huffman decoder can then use the encoded information and the lookup table (or dictionary) to retrieve the information back. It is worthy to know that these Huffman encoding and decoding operations are also lossless. The inverse zipper transformation can then be utilized to recover the original greyscale image with no loss incurred. The schematic of the whole process is as shown in Fig 4.

4. Experimental Setup and Results

4.1 Dataset

In the experiments, we used five grayscale images that are available online namely: *lenna.jpg- size 512×512*, *elaine.gif- size 512×512*, *cameraman.png- size 256×256*, *man.tif- size 512×512*, and *couple.png- size 512×512*. These are visualized in Fig 5.

4.2 Experimental Design

We carried out the experiments in MATLAB environment and we report the standard metrics: the compression ratio (CR) and the running time in (seconds) evaluated on a machine with Intel(r) Core(TM) i7-6700

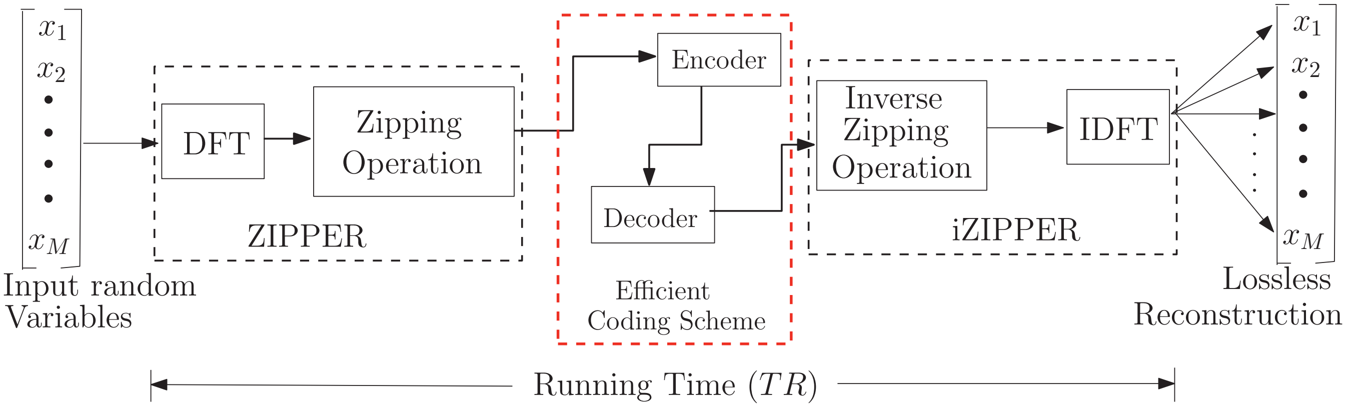


Fig. 4: Lossless image compression pipeline using Zipper Transformation

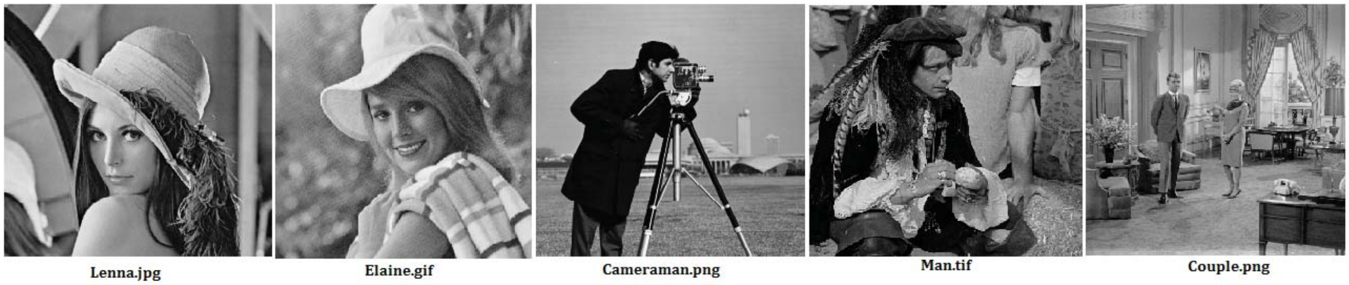


Fig. 5: Image data used in the experiments

CPU @ 3.40Ghz and a 64GB of RAM running a 64-bit Windows 10 Enterprise edition. The compression ratio (CR) is an important criterion in choosing a compression scheme for lossless image compression. The criterion is used to compare different compression paradigms, and is defined as:

$$CR = \frac{\text{Original file size}}{\text{Compressed file size}} \quad (1)$$

In order to benchmark the proposed scheme with other methods, we also implemented DCT and FWHT. We also utilized the running time to compare the proposed method with DCT and FWHT based compression algorithm. In this work, we define the running time as the time elapsed between the zipper transformation and the inverse transformation as shown in Fig. 4. The MATLAB implementation of these algorithms can be downloaded from https://github.com/babajide07/Zipper-Transformation.

4.3 Performance Evaluation

As shown in Fig 6-10, the ZT-based compression algorithm outperforms both the DCT and FWHT counterparts for different block sizes. This performance is more pronounced when the block size is 64. In addition to the poor compressing capability, implementing DCT and FWHT are more expensive in terms of running time

Table 1: Comparison table for DCT, FWHT and ZT using Lenna Image

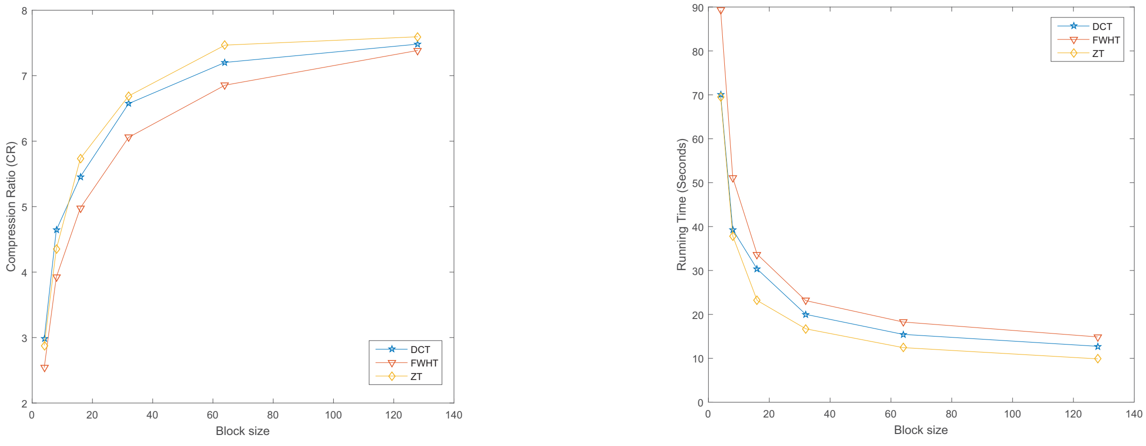
	DCT		FWHT		Zipper	
Block Size	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*
4	2.746	2.678	2.957	3.148	2.822	2.778
8	1.427	1.724	2.072	2.039	1.617	1.839
16	0.983	1.465	1.716	1.606	1.025	1.394
32	0.792	1.217	0.846	1.320	0.516	1.195
64	0.415	1.110	1.006	1.167	0.873	1.071
128	0.386	1.069	0.648	1.083	0.217	1.053

Table 2: Comparison table for DCT, FWHT and ZT using Elaine.GIF Image

	DCT		FWHT		Zipper	
Block Size	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*
4	3.3716	3.3266	3.4303	3.4169	3.1783	3.1689
8	2.0114	2.1092	2.1811	2.2309	1.8831	1.9490
16	1.0158	1.4790	1.4543	1.4775	1.1149	1.3087
32	0.5835	1.1591	1.2995	1.5012	1.1911	1.1691
64	0.2771	1.1365	0.6823	1.1042	0.4188	1.0507
128	0.1540	1.0378	0.2200	1.0772	0.1008	1.0216

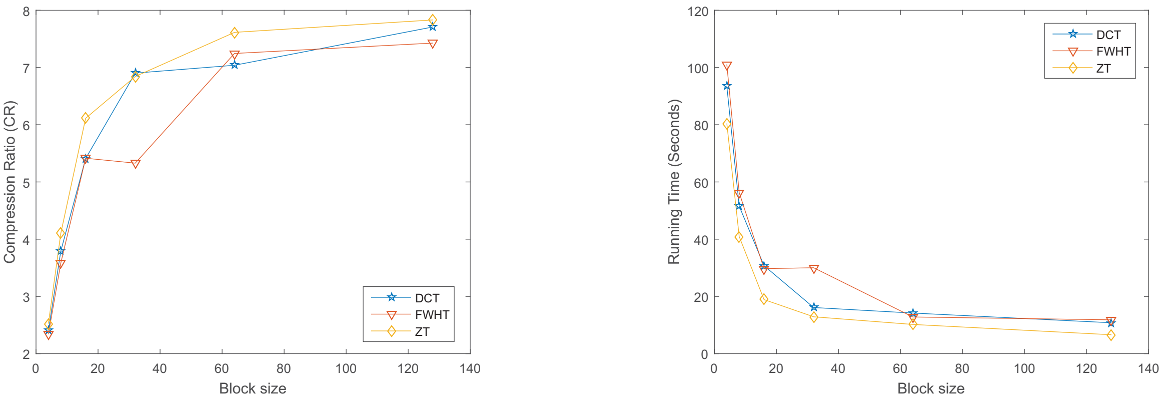
Table 3: Comparison table for DCT, FWHT and ZT using Cameraman.PNG Image

	DCT		FWHT		Zipper	
Block Size	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*
4	3.3260	3.1843	3.3128	3.3539	2.9919	3.0291
8	2.5294	2.3920	2.6009	2.4979	2.0671	2.1729
16	1.7815	1.8506	2.0891	2.0168	1.4019	1.9002
32	1.3217	1.4974	1.5239	1.7676	1.1557	1.4082
64	0.8406	1.2657	1.0793	1.5098	0.4568	1.1487
128	0.5813	1.1440	0.5756	1.1645	0.5211	1.0637



(a) (b)

Fig. 6: A plot of (a) compression ratio (b) running time of DCT, FWHT and Zipper against the block size using Lenna.jpg



(a) (b)

Fig. 7: A plot of (a) compression ratio (b) running time of DCT, FWHT and Zipper against the block size using Elaine.gif

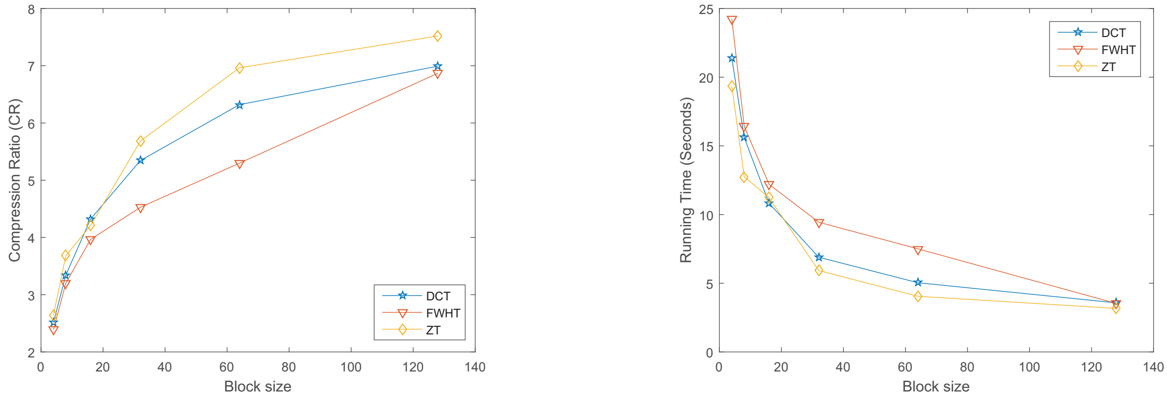
Table 4: Comparison table for DCT, FWHT and ZT using man.tif Image

	DCT		FWHT		Zipper	
Block Size	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*
4	3.8014	3.8186	3.8643	3.9015	3.6904	3.6949
8	2.7311	2.7637	2.9267	2.9952	2.3195	2.3512
16	2.0005	2.0510	2.0936	2.111	1.3979	1.6129
32	1.1594	1.4047	1.4526	1.6301	1.1506	1.2702
64	0.6654	1.2030	0.8032	1.2617	0.4205	1.1165
128	0.4224	1.1177	0.8214	1.1963	0.5513	1.1934

Table 5: Comparison table for DCT, FWHT and ZT using Couple.PNG Image

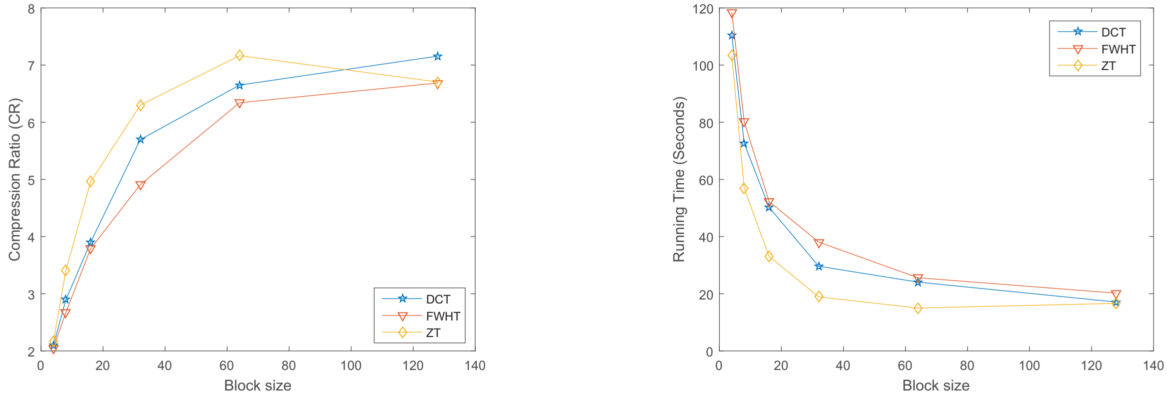
	DCT		FWHT		Zipper	
Block Size	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*	*Entropy*	*Av Lenght*
4	3.4584	3.4986	3.5477	3.5808	3.2588	3.3115
8	2.2741	2.3140	2.5116	2.4978	2.0270	2.2018
16	1.7169	1.9409	1.8903	1.8561	1.5641	1.5132
32	0.8346	1.3051	1.3810	1.4426	0.5976	1.1823
64	0.7715	1.3704	1.2156	1.3494	0.9582	1.1026
128	0.3324	1.0792	0.4184	1.1902	0.1738	1.0392

than the ZT-based lossleess compression. For instance in Fig 6, with block sizes of 4 and 8, DCT and zipper transform have similar running time, however as the block size increases, zipper transform performs better than DCT and FWHT in terms of both the compression and running time. Tables *I*-*V* give the average length of the codewords and the entropy for all the three methods that were compared in this study. It can be observed that the average length of the codeword decreases as the entropy decreases for all the three methods, and also the value of entropy is very close to the average codeword length.



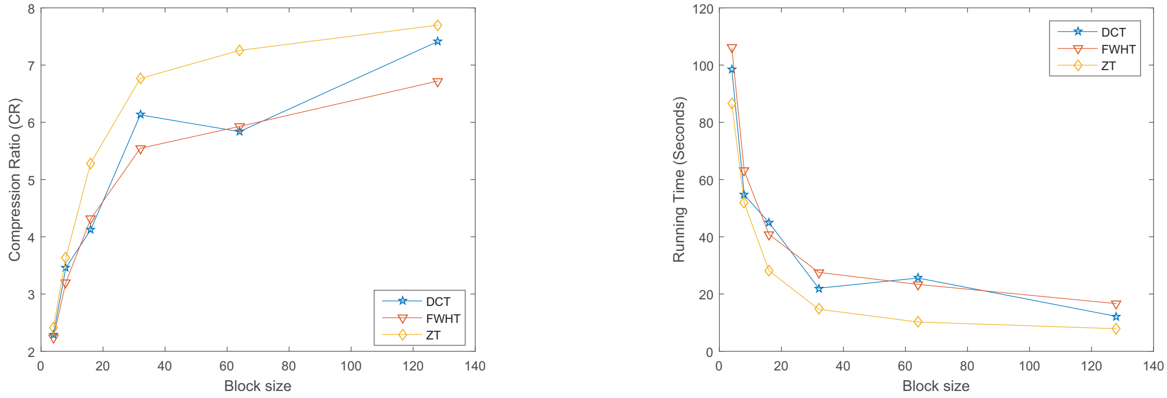
(a) (b)

Fig. 8: A plot of (a) compression ratio (b) running time of DCT, FWHT and Zipper against the block size using cameraman.png



(a) (b)

Fig. 9: A plot of (a) compression ratio (b) running time of DCT, FWHT and Zipper against the block size using man.tif



(a) (b)

Fig. 10: A plot of (a) compression ratio (b) running time of DCT, FWHT and Zipper against the block size using couple.png

5. Conclusion

We show in this study that using zipper transform based algorithm, a better compression can be achieved. The proposed algorithm also performed better than both DCT and FWHT based algorithms in terms of how much they are able to compress the image, and also, the implementation time of zipper based compression is superior to the other two methods we considered.

References

[1] D. Brunello, G. Calvagno, G. A. Mian, and R. Rinaldo, "Lossless compression of video using temporal information," *Image Processing, IEEE Transactions on*, vol. 12, no. 2, pp. 132–139, 2003.

[2] S.-G. Miaou, F.-S. Ke, and S.-C. Chen, "A lossless compression method for medical image sequences using jpeg-ls and interframe coding," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 5, pp. 818–821, 2009.

[3] H. Patel, U. Itwala, R. Rana, and K. Dangarwala, "Survey of lossless data compression algorithms," in *International Journal of Engineering Research and Technology*, vol. 4, no. 04 (April-2015). ESRSA Publications, 2015.

[4] S. Alzahir and A. Borici, "An innovative lossless compression method for discrete-color images," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 44–56, 2015.

[5] G. Campobello, O. Giordano, A. Segreto, and S. Serrano, "Comparison of local lossless compression algorithms for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 47, pp. 23–31, 2015.

[6] B. O. Ayinde, E. Hosseini-Asl, and J. M. Zurada, "Visualizing and understanding nonnegativity constrained sparse autoencoder in deep architecture," in *International Conference on Soft Computing and Artificial Intelligence*. Zakopane: Springer, June 2016.

[7] B. O. Ayinde and J. M. Zurada, "Clustering of receptive fields in autoencoder," in *International Joint Conference on Neural Networks*. Vancouver: IEEE, July 2016.

[8] B. O. Ayinde and A. Y. Barnawi, "Differential evolution based deployment of wireless sensor networks," in *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*. IEEE, 2014, pp. 131–137.

[9] H. A. Hashim, B. Ayinde, and M. Abido, "Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm," *Journal of Network and Computer Applications*, vol. 64, pp. 239–248, 2016.

[10] N. Egorov, D. Novikov, and M. Gilmutdinov, "Performance analysis of prediction methods for lossless image compression," in *Intelligent Interactive Multimedia Systems and Services*. Springer, 2015, pp. 169–178.

[11] X. Li and M. T. Orchard, "Edge-directed prediction for lossless compression of natural images," *Image Processing, IEEE Transactions on*, vol. 10, no. 6, pp. 813–817, 2001.

[12] J. Zhang and G. Liu, "An efficient reordering prediction-based lossless compression algorithm for hyperspectral images," *Geoscience and Remote Sensing Letters, IEEE*, vol. 4, no. 2, pp. 283–287, 2007.

[13] H. Wang, S. D. Babacan, and K. Sayood, "Lossless hyperspectral image compression using context-based conditional averages," in *null*. IEEE, 2005, pp. 418–426.

[14] D. A. Clunie, "Lossless compression of grayscale medical images: effectiveness of traditional and state-of-the-art approaches," in *Medical Imaging 2000*. International Society for Optics and Photonics, 2000, pp. 74–84.

[15] N. D. Memon and K. Sayood, "Lossless compression of video sequences," *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1340–1345, 1996.

[16] N. D. Memon, X. Wu, V. Sippy, and G. Miller, "Interband coding extension of the new lossless jpeg standard," in *Electronic Imaging'97*. International Society for Optics and Photonics, 1997, pp. 47–58.

[17] K. H. Yang *et al.*, "A contex-based predictive coder for lossless and near-lossless compression of video," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 1. IEEE, 2000, pp. 144–147.

[18] Y.-C. Hu and C.-C. Chang, "A new lossless compression scheme based on huffman coding scheme for image compression," *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 367–372, 2000.

[19] A. J. Pinho, "An online preprocessing technique for improving the lossless compression of images with sparse histograms," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 5–7, 2002.

[20] S.-G. Miaou and S.-N. Chao, "Wavelet-based lossy-to-lossless ecg compression in a unified vector quantization framework," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 3, pp. 539–543, 2005.

[21] M. Hans and R. W. Schafer, "Lossless compression of digital audio," *Signal Processing Magazine, IEEE*, vol. 18, no. 4, pp. 21–32, 2001.

Image Compression and Denoising Algorithm based on Multi-resolution Discrete Cosine Transform

Yanjun Zhao[1], **Saeid Belkasim**[2]
[1]Computer Science Department, Troy University, Troy, AL, USA
[2]Computer Science Department, Georgia State University, Atlanta, GA, USA

Abstract - *Discrete cosine transform (DCT) and wavelet transform coding system are the most popular image compression methods. Although DCT has outstanding energy compaction properties, blocking artifacts impact its performance. Wavelet avoids blocking artifacts; it is also the most popular approach to doing image compression and denoising simultaneously. However wavelet has higher computational complexity. Exploring an image in different resolutions reveals its dominant information in comparison to redundant one. We propose a novel multi-resolution DCT; based on our multi-resolution DCT, we propose a novel algorithm to do image compression and denoising simultaneously. Our algorithm achieves multi-resolution analysis, avoids blocking artifacts, has excellent energy compaction property and is ideal for parallel computing. Compared to wavelet, our algorithm has good computation accuracy and efficiency.*

Keywords: Multi-resolution Discrete Cosine Transform (Multi-resolution DCT); Multi-resolution Analysis; Blocking Artifacts; Compression and Denoising.

1 Introduction

The goal of image compression is to reduce redundancy of image data to efficiently store or transmit data while preserving quality required for a given application [1]. Many compression schemes based on lossless or lossy criteria are proposed [1-8]. Although lossy compression is irreversible; it maintains visually lossless data [9]. To achieve higher compression ratios, lossy compression is applied. Various coding schemes are proposed for lossy compression, including predictive coding [10], subband coding [11], transform coding [12], vector quantization [13, 14]. The most popular one is transform coding; discrete cosine transform (DCT) [15, 16] and wavelet transform [3, 16-18], are the most popular transform coding methods.

DCT has excellent energy compaction property and high computation efficiency; however the performance of its traditional coder generally degrades at high compression ratios mainly due to the underlying block-based strategy [19]. Each block of DCT coefficients only represents the local information of an image. Separately compressing each block breaks correlation between the pixels at the borders of blocks and causes blocking artifacts [19].

Wavelet avoids blocking artifacts and maintains high image quality at high compression ratios; it is more robust under transmission and decoding errors [16, 18]. However wavelet has high computation complexity [16].

Image compression removes redundant information; if the removed information is noise, image compression and denoising can be done simultaneously. Currently the most popular method for doing image compression and denoising simultaneously is also wavelet.

Exploring an image in different resolutions reveals its dominant information in comparison to redundant one. We propose a novel multi-resolution discrete cosine transform (multi-resolution DCT); based on our multi-resolution DCT, we propose a novel algorithm to do image compressing and denoising simultaneously. Our algorithm achieves multi-resolution analysis, avoids blocking artifacts, maintains outstanding energy compaction property of traditional DCT, and is ideal for parallel processing. Compared with wavelet, our algorithm has good computation accuracy and efficiency.

2 Multi-resolution DCT

2.1 An odd-even image tree

A $K_1\ by\ K_2$ image can be represented as a $K_1\ by\ K_2$ matrix $A(k_1, k_2)$ where $k_1 = 0, 1, 2, \dots, K_1 - 1$ and $k_2 = 0, 1, 2, \dots, K_2 - 1$. An odd-even image tree is constructed through dyadically dividing an image A into four sets: a set of odd-odd pixels that contains pixels from odd rows and odd columns; a set of odd-even pixels that contains pixels from odd rows and even columns, a set of even-odd pixels that contains pixels from even rows and odd columns, and a set of even-even pixels that contains pixels from even rows and even columns. The multi-resolution image at node m of tree level r where $m = 1, 2, \dots, 4^r$ and $r = 0,1,2,\dots$ is represented as $A_r^m((k_1)_r^m, (k_2)_r^m)$ where $(k_1)_r^m = 0, 1, 2, \dots, (K_1)_r - 1$; $(k_2)_r^m = 0, 1, 2, \dots, (K_2)_r - 1$; $(K_1)_r = \frac{K_1}{2^r}$ and $(K_2)_r = \frac{K_2}{2^r}$. Each node A_r^m represents the original image in a global view.

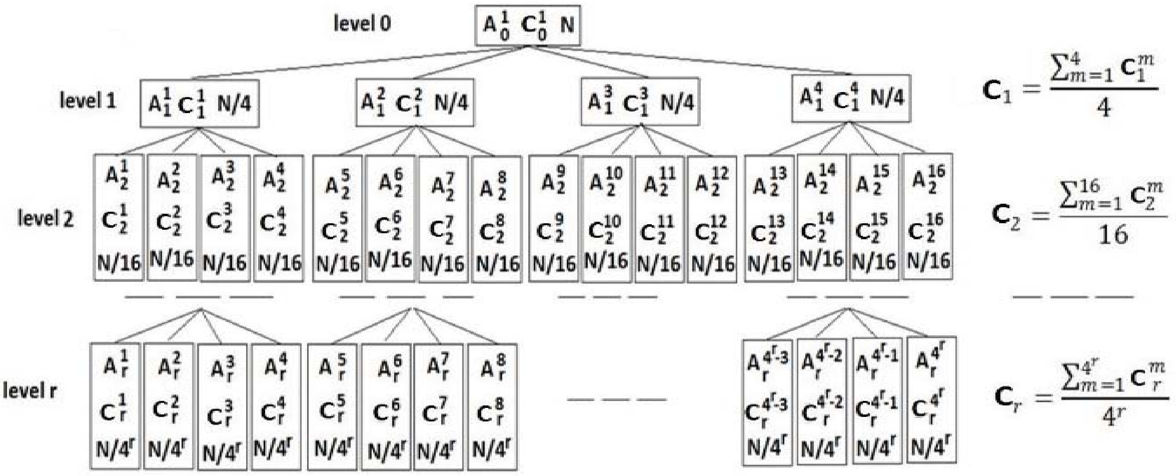

level 0
A₀¹ C₀¹ N
level 1
A₁¹ C₁¹ N/4
A₁² C₁² N/4
A₁³ C₁³ N/4
A₁⁴ C₁⁴ N/4
level 2
A₂¹ … A₂¹⁶
C₂¹ … C₂¹⁶
N/16
level r
A_r¹ … A_r⁸, A_r^{4^r-3}, A_r^{4^r-2}, A_r^{4^r-1}, A_r^{4^r}
C_r¹ … C_r⁸, C_r^{4^r-3}, C_r^{4^r-2}, C_r^{4^r-1}, C_r^{4^r}
N/4^r
C₁ = Σ⁴ₘ₌₁ C₁ᵐ / 4
C₂ = Σ¹⁶ₘ₌₁ C₂ᵐ / 16
C_r = Σ^{4^r}ₘ₌₁ C_rᵐ / 4^r

Fig. 1 A DCT tree (Multi-resolution DCT)
A: image; C: DCT coefficients set; N: The size of the original image $N = K_1 \times K_2$.

2.2 Multi-resolution DCT

Through applying traditional DCT to the odd-even image tree, we generate our multi-resolution DCT:

$$\begin{aligned} &C_r^m(u_r^m, v_r^m) \\ &= w_1(u_r^m)\times w_2(v_r^m) \\ &\times \sum_{(k_1)_r^m=0}^{(K_1)_r-1}\sum_{(k_2)_r^m=0}^{(K_2)_r-1} A_r^m((k_1)_r^m,(k_2)_r^m) \\ &\times cos\frac{(2\times(k_1)_r^m+1)\times u_r^m\times\pi}{2\times(K_1)_r} \\ &\times cos\frac{(2\times(k_2)_r^m+1)\times v_r^m\times\pi}{2\times(K_2)_r} \qquad (1) \end{aligned}$$

Where $w_1(u_r^m) = \begin{cases} \frac{1}{\sqrt{(K_1)_r}} & u_r^m = 0 \\ \sqrt{\frac{2}{(K_1)_r}} & u_r^m = 1,2,\dots,(K_1)_r - 1 \end{cases}$;

$w_2(v_r^m) = \begin{cases} \frac{1}{\sqrt{(K_2)_r}} & v_r^m = 0 \\ \sqrt{\frac{2}{(K_2)_r}} & v_r^m = 1,2,\dots,(K_2)_r - 1 \end{cases}$;

$u_r^m = 0, 1, \dots, (K_1)_r - 1$; $v_r^m = 0, 1, \dots, (K_2)_r - 1$.

The multi-resolution DCT can be represented by a tree of DCT sets, shown in Fig. 1. The DCT sets from different tree levels represent the original image in different resolutions and the resolution is inversely proportional to the height of the tree. The highest resolution occurs at root whereas the lowest ones occur at leaves; each other node has four children of lower resolution in terms of odd-odd, odd-even, even-odd and even-even DCT set and a parent node of higher resolution. At tree level r, each of 4^r DCT sets independently represents the original image with $\frac{1}{4^r}$ of its original size in a global view; all 4^r DCT sets together lossless represent the original image.

At level r, by computing mean of 4^r DCT sets, we use a DCT set $C_r$to represent all 4^r DCT sets together:

$$C_r(u_r, v_r) = \frac{1}{4^r}\sum_{m=1}^{4^r} C_r^m(u_r^m, v_r^m) \qquad (2)$$

Where $u_r = 0, 1, \dots, (K_1)_r - 1$; $v_r = 0, 1, \dots, (K_2)_r - 1$.

The original image can be reconstructed:

$$\begin{aligned} I_r((k_1)_r,(k_2)_r) &= \sum_{u_r=0}^{(K_1)_r-1}\sum_{v_r=0}^{(K_2)_r-1} w_1(u_r)\times w_2(v_r) \\ &\times C_r(u_r, v_r) \times cos\frac{(2\times(k_1)_r+1)\times u_r\times\pi}{2\times(K_1)_r} \\ &\times cos\frac{(2\times(k_2)_r+1)\times v_r\times\pi}{2\times(K_2)_r} \qquad (3) \end{aligned}$$

Where $w_1(u_r) = \begin{cases} \frac{1}{\sqrt{(K_1)_r}} & u_r = 0 \\ \sqrt{\frac{2}{(K_1)_r}} & u_r = 1,2,\dots,(K_1)_r - 1 \end{cases}$;

$w_2(v_r) = \begin{cases} \frac{1}{\sqrt{(K_2)_r}} & v_r = 0 \\ \sqrt{\frac{2}{(K_2)_r}} & v_r = 1,2,\dots,(K_2)_r - 1 \end{cases}$;

$(k_1)_r = 0, 1, \dots, (K_1)_r - 1$; $(k_2)_r = 0, 1, \dots, (K_2)_r - 1$.

The size of reconstructed image $I_r((k_1)_r,(k_2)_r)$ is $\frac{1}{4^r}$ of the original image size.

In traditional DCT coding system, each DCT set only represents local information of the original image, which leads to blocking artifacts. In our multi-resolution DCT, each DCT set individually represents the global information of the original image, which avoids blocking artifacts. By computing the mean of the DCT sets at the same resolution level, we further enhance the correlations among pixels.

Image noise is random variation of intensity information in an image. At the same resolution level, each DCT set, globally representing the noised image, can be considered as a sample of the noised image; computing mean of the DCT sets at the same resolution level is a good way to smooth this image which in turn reduces its embedded noises.

3 Compression & denoising algorithm

Our algorithm has two parts: encoding and decoding.

A. Encoding

Mapper: map an image into a DCT tree to represent this image in multi-resolution levels.

Quantizer: at tree level r, compute the mean of 4^r DCT sets; Since DCT has strong energy compaction property; based on a threshold T , threshold mean C_r to C_r' for further compression. C_r' globally represents the original image; its compression ratio $a_r > 4^r$.

Output C_r' as compressed and denoised image.

B. Decoding

Inverse Mapper: compute reconstructed image I_r' by applying invert DCT to C_r'; resize I_r' to A_r' with the same size of the original image, by applying bicubic interpolation to I_r'. Bicubic interpolation is good for image smoothing which in turn further reduces noises.

Output A_r' as decompressed and denoised image.

The odd-even image tree and DCT tree are also ideal for parallel computing.

Our Image Compression and Denoising Algorithm

Input: Image A
Output: Compressed and denoised image C_r'
Decompressed and denoised image A_r'

Part One: Encoding

1.1 Transform the image A into an odd-even image tree $A_r^m((k_1)_r^m,(k_2)_r^m)$ where $(k_1)_r^m = 0,1,\ldots,(K_1)_r - 1$; $(k_2)_r^m = 0,1,\ldots,(K_2)_r - 1$; $(K_1)_r = \frac{K_1}{2^r}$; $(K_2)_r = \frac{K_2}{2^r}.;m = 1,2,3,\ldots,4^r;r = 0,1,2,\ldots$.

1.2 Through applying traditional DCT to the odd-even image tree, generate DCT tree $C_r^m(u_r^m,v_r^m) = w_1(u_r^m)\times w_2(v_r^m)\times \sum_{(k_1)_r^m=0}^{(K_1)_r-1}\sum_{(k_2)_r^m=0}^{(K_2)_r-1} A_r^m((k_1)_r^m,(k_2)_r^m)\times cos\frac{(2\times(k_1)_r^m+1)\times u_r^m\times\pi}{2\times(K_1)_r}\times cos\frac{(2\times(k_2)_r^m+1)\times v_r^m\times\pi}{2\times(K_2)_r}$

1.3 Represent all of 4^r DCT sets at tree level r, by computing their mean $C_r\,(u_r,v_r) = \frac{1}{4^r}\sum_{m=1}^{4^r} C_r^m(u_r^m,v_r^m)$

1.4 Threshold C_r to C_r' based on a threshold T.

Output C_r' as the compressed and denoised image.

Part Two: Decoding

2.1 Through applying invert DCT to C_r' , generate reconstructed image $I_r'((k_1)_r,(k_2)_r) = \sum_{u_r=0}^{(K_1)_r-1}\sum_{v_r=0}^{(K_2)_r-1} w_1(u_r)\times w_2(v_r)\times C_r'(u_r,v_r)\times cos\frac{(2\times(k_1)_r+1)\times u_r\times\pi}{2\times(K_1)_r}\times cos\frac{(2\times(k_2)_r+1)\times v_r\times\pi}{2\times(K_2)_r}$

2.2 Generate decompressed and denoised image A_r' with the same size of the original image A, by applying bicubic interpolation to image I_r'.

Output A_r' as the decompressed and denoised image.

4 Simulation

Since wavelet is the most widely used method for image compression and denoising, this simulation is to compare our algorithm with wavelet in terms of computation accuracy and efficiency.

4.1 Simulation data

We use one standard image “Lena”, and two image databases “MITForest” and “PCA” as simulation data. “MITForest” has 328 images of different forests. It is downloaded from http://www-cvr.ai.uiuc.edu/ponce_grp/data/ (under “Fifteen Scene Categories” of the webpage). PCA has 91 images of different backgrounds. It is downloaded from http://pics.psych.stir.ac.uk/Other_image_types.htm.

We use Matlab function *im2double* to convert the intensity image to double precision; then we respectively add Gaussian noise with *mean =0 & variance =0.005* and “Salt &Pepper” noise with *noise density=0.02* into image.

4.2 Simulation process

We respectively apply our algorithm and wavelets to the images corrupted by noise for compression and denoising.

We construct our multi-resolution DCT C_r^m where $m = 1,2,\ldots,4^r; r = 0,1,2$. Using threshold $T = 0.01$, we get compressed & denoised image C_1' with compression ratio $a_1 > 4$ and compressed & denoised image C_2' with compression ratio $a_2 > 16$. Based on C_1' and C_2' , we compute decompressed & denoised image A_1' and A_2' respectively.

Haar and Biorthogonal 1.5 [19] wavelet are widely used in image compression and denoising; we respectively apply these wavelets in two levels.

Applying wavelet transform to image A , we get approximation coefficient CA_1 with details CH_1, CV_1, CD_1.We considered CA_1 with threshold details th_CH_1 , th_CV_1 , th_CD_1 as compressed & denoised image in level one; its compression ratio $a_1 < 4$. Applying wavelet transform to CA_1, we get approximation coefficient CA_2 with details CH_2, CV_2, CD_2 .We considered CA_2 with threshold details th_CH_2 , th_CV_2 , th_CD_2 , th_CH_1 , th_CV_1 , th_CD_1 as compressed & denoised image in level two; its compression ratio $a_2 < 16$. Denoising threshold is generated by Matlab function *ddencmp*.

Applying invert wavelet transform to CA_1 with th_CH_1, th_CV_1, th_CD_1 , we get decompressed & denoised image for level one. Applying invert wavelet transform to CA_2 with th_CH_2 , th_CV_2 , th_CD_2 , we get $\mathrm{CA}_1{}'$; applying invert wavelet transform to $CA_1{}'$ with th_CH_1, th_CV_1 , th_CD_1, we get decompressed & denoised image for level two.

We use two popular measurements to measure computation accuracy: (1) Mean Square Errors (MSE) between image without noise and decompressed & denoised image. The lower MSE is, the higher quality of decompressed & denoised image is. (2) Peak Signal-to-Noise Ratio (PSNR) between image without noise and decompressed & denoised image. The higher PSNR is, the higher quality of decompressed & denoised image is. We use execute time of each method to measure computation time.

4.3 Simulation results

Figure 2-5 and Table I-II show simulation results. H_1 and H_2 represent decompressed & denoised image based on Haar wavelet in level one and two respectively; B_1 and B_2 represent decompressed & denoised image based on Biorthogonal 1.5 wavelet in level one and two respectively; DCT_1 and DCT_2 represent decompressed & denoised image based on our algorithm in level one and two respectively.

Figure 2 - 5 show simulation results about "Lena" image Figure 2 and 4 show whole images. To highlight differences, we zoom out part of the images (right corner of Lena's forehead with part of her hair and hat) and get Figure 3 and 5. Figure 3 and 5 show that both wavelets methods generate artefacts resulting into pixelated-like images; by contrast, our algorithm generates much smoother images. These figures demonstrate that our algorithm produces higher quality of decompressed & denoised images than wavelets.

Table I shows simulation results about "Lena" image in a quantitative way. In the same level, our algorithm has the minimum MSE, maximum PSNR and minimum computation time. Furthermore, MSE of our algorithm in level two is lower than MSE of wavelets in level one; PSNR of our algorithm in level two is higher than PSNR of wavelets in level one; computation time of our algorithm in level two is less than computation time of wavelets in level one.

To show generalizability of our algorithm, we summed up simulation results about "MITForest" and "PCA" image database into table II. For each database, in the same level, our algorithm always has the minimum average MSE, maximum average PSNR and minimum average computation time.

Table I: MSE, PSNR and Computation Time for "Lena" Image

Image	Algorithm	Compress Ratio	MSE		PSNR		Computation Time	
			Gaussian	Salt & Pepper	Gaussian	Salt & Pepper	Gaussian	Salt & Pepper
Lena	$H1$	$a_1 < 4$	0.00193245	0.00217415	27.1389	26.6271	0.0746993	0.0898552
	$H2$	$a_2 < 16$	0.00233917	0.00241216	26.3094	26.1759	0.0848969	0.0945513
	$B1$	$a_1 < 4$	0.00213869	0.00239362	26.6985	26.2094	0.0900773	0.0840282
	$B2$	$a_2 < 16$	0.0025976	0.00264752	25.8543	25.7716	0.106538	0.0962408
	$\boldsymbol{DCT1}$	$\boldsymbol{a_1 > 4}$	**0.00118118**	**0.00135504**	**29.2768**	**28.6805**	**0.0351652**	**0.042245**
	$\boldsymbol{DCT2}$	$\boldsymbol{a_2 > 16}$	**0.00151805**	**0.0015847**	**28.1871**	**28.0005**	**0.0355801**	**0.0458064**

Table II: Average MSE, Average PSNR and Average Computation Time for "MITForest" and "PCA" Image Database

Image Database	Algorithm	Compress Ratio	Average MSE		Average PSNR		Average Computation Time	
			Gaussian	Salt & Pepper	Gaussian	Salt & Pepper	Gaussian	Salt & Pepper
MITForest	$H1$	$a_1 < 4$	0.00843341	0.00885358	21.3927	21.1373	0.0189051	0.0194192
	$H2$	$a_2 < 16$	0.0133399	0.013447	19.4387	19.3949	0.0232619	0.0225446
	$B1$	$a_1 < 4$	0.00900598	0.00945044	21.0843	20.8331	0.0198465	0.0199823
	$B2$	$a_2 < 16$	0.0139589	0.014064	19.2226	19.1817	0.0260882	0.0257389
	$\boldsymbol{DCT1}$	$\boldsymbol{a_1 > 4}$	**0.00744664**	**0.00774658**	**22.1253**	**21.8997**	**0.0108464**	**0.0114491**
	$\boldsymbol{DCT2}$	$\boldsymbol{a_2 > 16}$	**0.012383**	**0.012471**	**19.8235**	**19.7831**	**0.0132942**	**0.0134945**
PCA	$H1$	$a_1 < 4$	0.00414767	0.00484132	24.5298	23.662	0.0229396	0.019711
	$H2$	$a_2 < 16$	0.0069259	0.00708204	22.5379	22.3845	0.023457	0.0215927
	$B1$	$a_1 < 4$	0.00445186	0.00522658	24.2041	23.3075	0.0213808	0.0190785
	$B2$	$a_2 < 16$	0.00735283	0.00754082	22.2556	22.0835	0.0271085	0.0246231
	$\boldsymbol{DCT1}$	$\boldsymbol{a_1 > 4}$	**0.00320196**	**0.00365755**	**25.9275**	**25.1062**	**0.0122237**	**0.0122461**
	$\boldsymbol{DCT2}$	$\boldsymbol{a_2 > 16}$	**0.00572177**	**0.00582161**	**23.5727**	**23.4425**	**0.0138769**	**0.0128229**

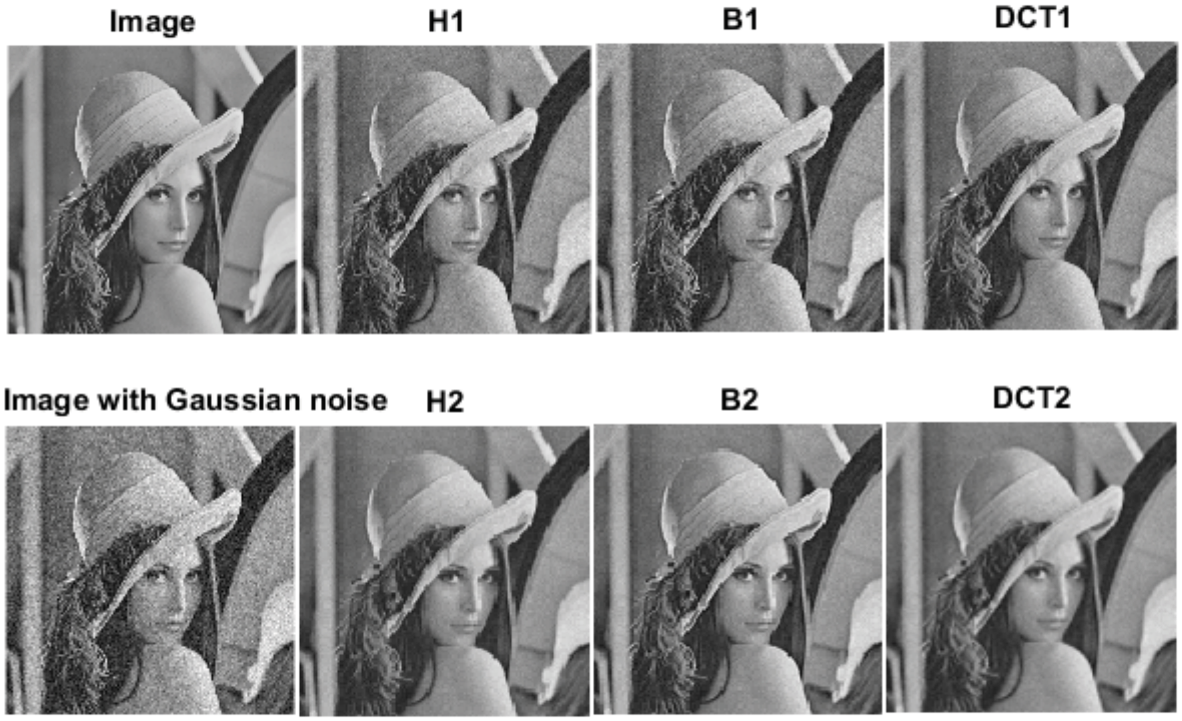


Fig. 2 “Lena” image under Gaussian noise with mean =0 & variance =0.005

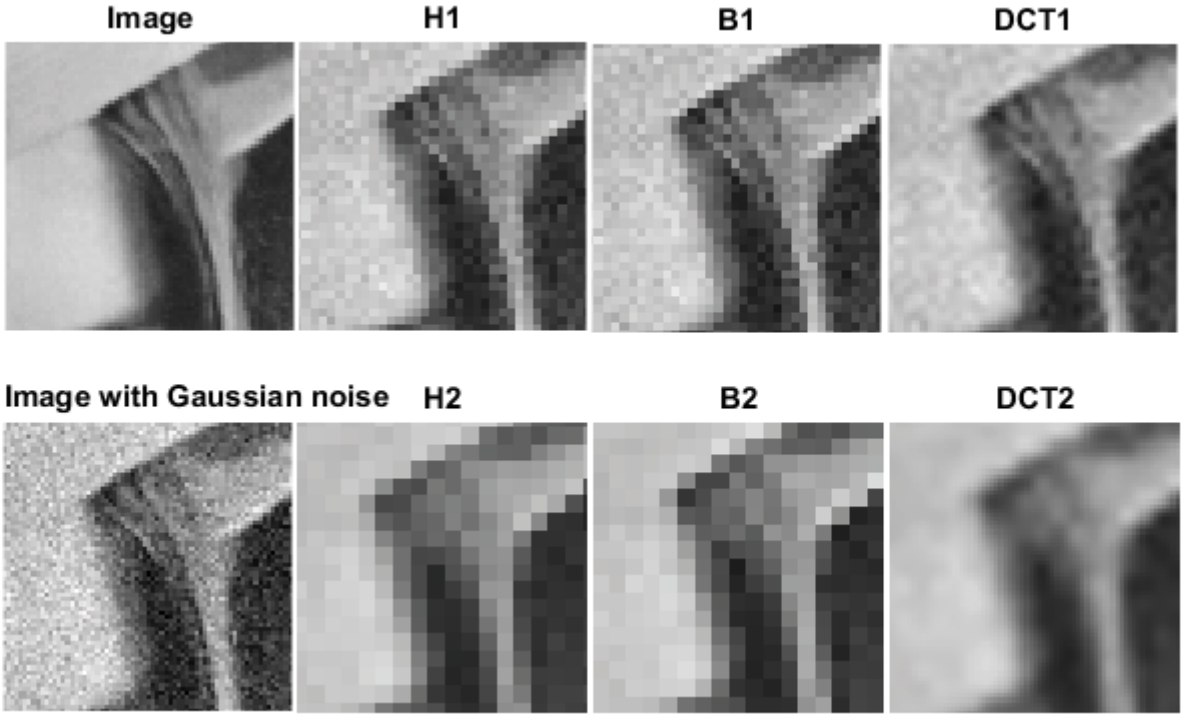


Fig.3 Fragment of “Lena” image under Gaussian noise with mean =0 & variance =0.005

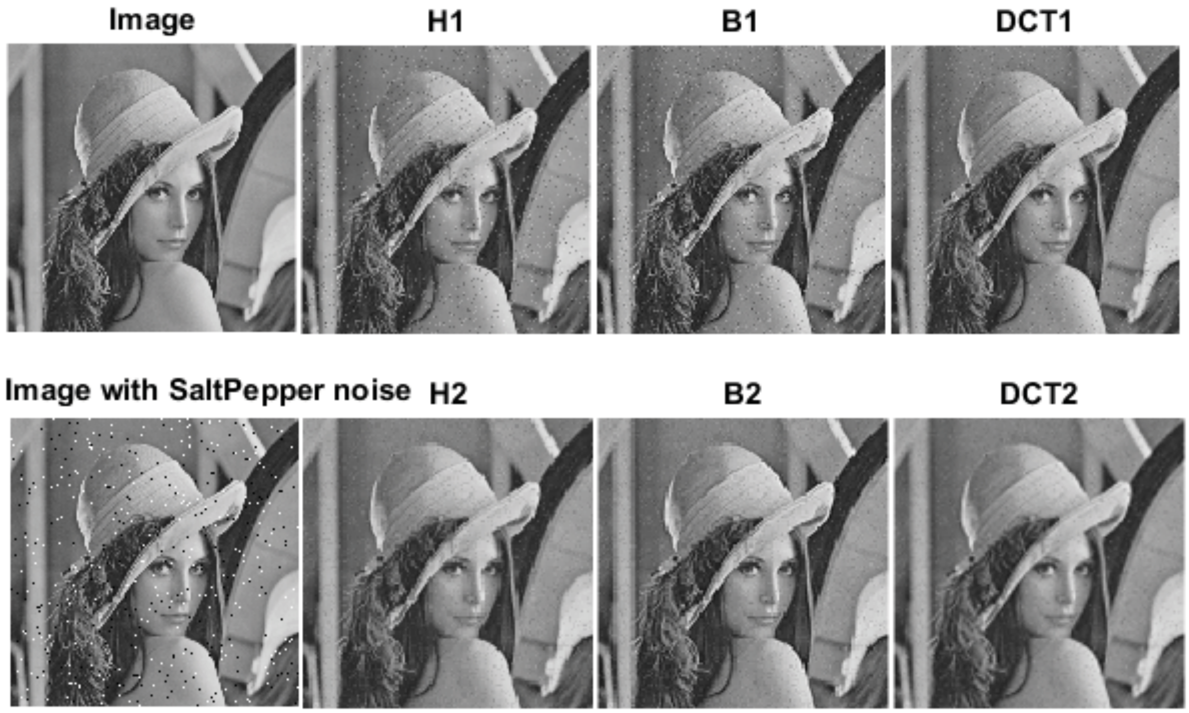


Fig. 4 “Lena” image under “Salt & Pepper” noise with noise density =0.02

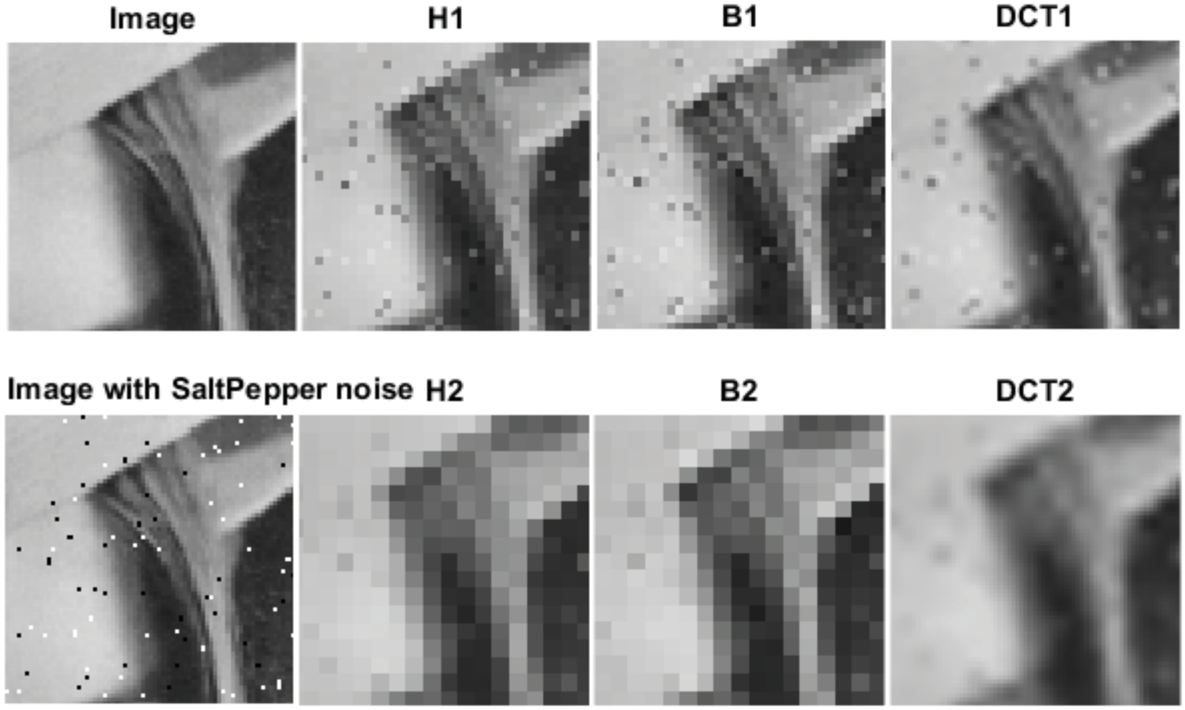


Fig. 5 Fragment of “Lena” image under “Salt & Pepper” noise with noise density=0.02

5 Conclusions

Exploring an image in different resolutions reveals its dominant information in comparison to redundant one. We propose a novel multi-resolution DCT; based on our multi-resolution DCT, we propose a novel algorithm for performing image compression and denoising simultaneously. Our algorithm avoids blocking artifacts, has excellent energy compaction property, achieves multi-resolution analysis, and is ideal for parallel computing. Compared to wavelet, our algorithm has good computation accuracy and efficiency.

6 References

[1] Mark Nelson and Jean-Loup Gailly. “The Data Compression Book”. The second edition. M&T Books, 1995.

[2] Rakesh Chalasani, Jose C. Principe and Naveen Ramakrishnan . "A fast proximal method for convolutional sparse coding". The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, U.S., pp. 1-5, August, 2013.

[3] Wang Yannan, Zhang Shudong and Liu Hui. "Study of Image Compression Based on Wavelet Transform". 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, Zhangjiajie, China, pp. 575-578, November, 2013.

[4] Simone Milani and Pietro Zanuttigh, "Compression of photo collections using geometrical information". 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, Italy, pp. 1-6, July, 2015.

[5] Guochao. Zhang, Shaohui Liu, Feng Jiang, Debin Zhao and Wen Gao. "An improved image compression scheme with an adaptive parameters set in encrypted domain". IEEE Conference on Visual Communications and Image Processing (VCIP), Kuching, Sarawak, Malaysia, pp. 1-6, November, 2013.

[6] Joaquin Zepeda, Christine Guillemot and Ewa Kijak. "Image compression using the Iteration-Tuned and Aligned Dictionary". 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, pp. 793-796, May, 2011.

[7] Rane, Shantanu, Petros Boufounos, Anthony Vetro and Yu Okada. "Low complexity efficient raw SAR data compression". In SPIE Defense, Security, and Sensing, pp. 80510W-80510W. International Society for Optics and Photonics, May, 2011.

[8] Rime Raj Singh Tomar and Kapil Jain. "Lossless Image Compression Using Differential Pulse Code Modulation and its Application". 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp. 543-545, Gwalior, MP, India, April, 2015.

[9] Vladimir V. Lukin, Mikhail S. Zriakhov, Nikolay N. Ponomarenko, Sergey S. Krivenko and Miao Zhenjiang. “Lossy compression of images without visible distortions and its application”. IEEE 10th International Conference on Signal Processing Proceedings, Beijing, China, pp. 698- 701, October, 2010.

[10] Sceuchin Chuah, Sorina Dumitrescu and Xiaolin Wu. "ℓ2 optimized predictive image coding with ℓ∞ bound". 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, pp. 1315-1319, May, 2013.

[11] John W. Woods and Sean D. O’Neil. "Subband coding of images". IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34, no. 5, pp. 1278-1288, October, 1986.

[12] Vivek K. Goyal. "Theoretical foundations of transform coding". IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 9-21, September, 2001.

[13] Meina Xu and Anthony Kuh "Image coding using feature map finite-state vector quantization". IEEE Signal Processing Letters, vol. 3, no. 7, pp. 215-217, July, 1996.

[14] Allen Gersho and Robert M. Gray. “Vector Quantization and Signal Compression”. The first edition, Springer US, 1992.

[15] N. Ahmed, T. Natarajan and K. R. Rao. "Discrete Cosine Transform". IEEE Transactions on Computers, vol. C-23, no.1, pp. 90-93, January, 1974.

[16] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard and Ya-Qin Zhang. "A comparative study of DCT- and wavelet-based image coding". IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 5, pp. 692-695, August, 1999.

[17] Martin Vetterli. "Wavelets, approximation, and compression". IEEE Signal Processing Magazine, vol. 18, no.5, pp. 59-73, September, 2001.

[18] Bryan E. Usevitch. "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000". IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 22-35, September, 2001.

[19] Rafael C. Gonzalez and Richard E. Woods. “Digital Image Processing”. The third edition, Pearson Prentice Hall, 2008.

SESSION

RECOGNITION SYSTEMS, TRACKING, DETECTION, AND APPLICATIONS

Chair(s)

TBA

TRACKING A TABLE TENNIS BALL FOR UMPIRING PURPOSES USING A MULTI-AGENT SYSTEM

Hnin Myint, Patrick Wong, Laurence Dooley
The Open University
United Kingdom
{hnin.myint, patrick.wong,
Laurence.Dooley}@open.ac.uk

Adrian Hopgood
HEC Liege – Management School
University of Liege
Belgium
adrian.hopgood@ulg.ac.be

ABSTRACT

Tracking a table tennis ball for umpiring purposes is a challenging task as, in real-match scenarios, the ball travels fast and can become occluded or merged with other background objects. This paper presents the design of a multi-view based tracking system that can overcome the challenges of tracking a ball in real match sequences. The system has been tested on a complete table tennis rally and the results are very promising. The system is able to continuously track the ball with only marginal variations in detection. Furthermore, the initialization of the multi-camera system means it is both a portable and cost-effective solution for umpiring purposes.

Index Terms— object detection, tracking, multi-view, multi-agent system, table tennis

Regular Research Paper – Computer Vision Application

1. INTRODUCTION

Table tennis is a popular Olympic sport with millions of regular players and tournaments held worldwide. In tournaments, one important task is to umpire the matches accurately. Umpiring a table tennis match is challenging even for professionally trained practitioners as many observations are required within a very short period of time. For example, during the service part of a rally, 31 observations are required within a second [1]. Furthermore, some of the observations and judgements described by the table tennis rules are very difficult for humans to make, such as to whether the ball touches the net, hits the edge of the table or exceeds the minimum height allowed during services. To this end, a purpose-built computer vision system that is able to evaluate and assist in identifying these difficult observations provides the motivation for this work. One important activity of such a system is to accurately and rapidly track the location of the ball during a match. While the ultimate objective is to develop an automatic umpiring system, the focus in this paper is upon the design of this purposed-built ball tracking system. In order for an automatic umpiring system to be widely applied, it needs to be portable, inexpensive, and able to accurately track in real-time objects such as a table tennis ball in real match scenes.

The remainder of the paper is organized as follows: Section 2 reviews the literature relating to tracking table tennis balls, while Section 3 describes the proposed multi-view ball detection and tracking framework. Section 4 presents the experimental set up, results and discussion, while Section 5 makes some concluding comments.

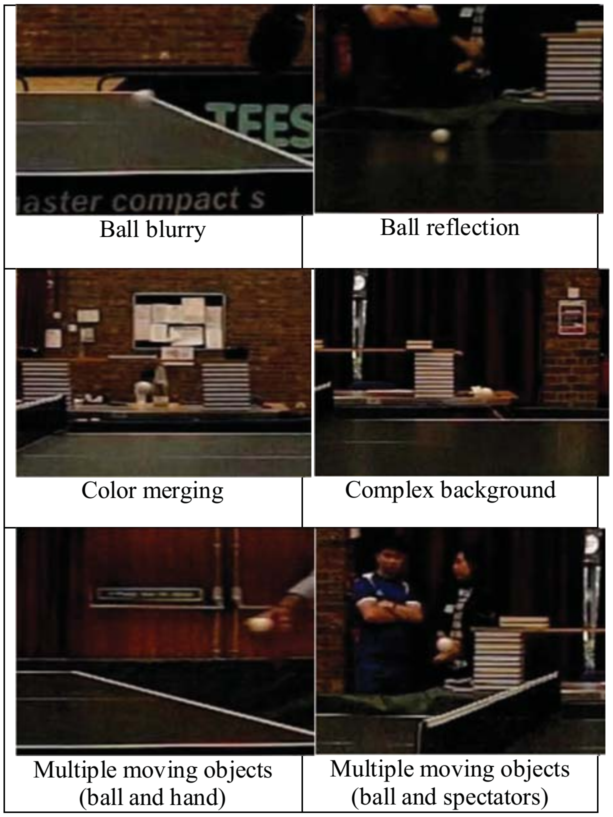
2. LITERATURE REVIEW

Although many publications related to object tracking are available, the techniques that produce satisfactory results are usually object and application specific [1]. Thus the survey of literature for this paper will focus on tracking a table tennis ball. Two previous solutions for ball tracking for umpiring purposes have been proposed [1] and [2], while the other relevant methods [4-7] have tended to involve playing robots rather than real matches. A two-pass color thresholding (CT) technique is proposed by [1] to identify the ball from a real match scene. While satisfactory results were achieved, it only covered the service part of the rallies during which the ball is not traveling at very high speed. In contrast, the automated scoring system in [2] tracks the table tennis ball in real time and relates to the table and net to determine when a point is scored. However, the setting is based on a laboratory environment with the ball painted a special neon green against a uniformly black background. Such a prescribed environment significantly reduces the ball detection challenge, and so this is unacceptable for formal tournaments, as the Laws of Table Tennis mandate the color of ball can only be matt white or orange [3]. The growth of sampled points method was proposed by [4], which aims to recover pixels unintentionally lost during the adjacent frame differencing. Although experimental results reveal promising ball detection performance, background pixels can be incorrectly classified as belonging to the ball, which degrades the overall detection accuracy. On the other hand, [5] attempted to improve the tracking performance by employing an iterative dynamic window tracking method. Despite the system being able to track the ball reasonably well, the computational complexity involved mitigated against real-time operation. An alternative approach to the ball tracking, involved a multi-view scheme with an aerodynamic model of the ball's three dimensional (3D) flight path [6] and [7]. It was further improved in [7] with an additional bouncing model, which takes into consideration the bouncing characteristic of the ball. While some degree of improvement in the tracking of a table tennis ball is achieved, the test video sequences used comprised relatively simple backgrounds, which does not resemble a real match scene, where the view can become obstructed by the players against complex backgrounds where ball merging can occur. Moreover, in the dual-camera based 3D trajectory reconstruction algorithm, it is critical that both cameras detect the ball at the same time in order to calculate the 3D position of the ball. This restriction means that the 3D position of the ball cannot be determined when one of the cameras fails to detect the ball. In addition, a vision system using multiple cameras must simultaneously process frames from different views, thus incurring a heavy workload which impacts on the system's performance.

In summary, no existing literature addresses the challenges of tracking the ball in a full real match rally, which is an essential requirement for a realistic automatic umpiring system. Furthermore, the success of current solutions heavily relies on the availability of aerial views of the scene, which renders the ball against a relatively simple background i.e., a uniform colored table and floor [4]-[7]. Obtaining aerial views is not always possible as most table tennis tournaments take place at multi-purpose sport venues and fixing cameras to the ceiling or high wall is not allowed. This paper thus proposes a novel and portable multi-view ball tracking system which is designed to manage challenging ball detection situations in real match scenes.

3. Multi-View Ball Tracking Framework

To umpire a match automatically, one basic requirement is to track the ball's real world 3D position and use it to compare with the positions of other objects such as the table and net. The main difficulties of tracking the ball in a match scene are that the view of the ball can be occluded, merged with objects in the background, its trajectory can change suddenly as it is struck and it travels fast. Six example frames reflecting these challenging detection situations are shown in Figure 1. One approach to overcome occlusion is to employ a multi-view system which monitors the ball at different angles. This approach also enables the derivation of the ball's 3D position. However, this requires a high degree of inter-view co-ordination as well as the ability to resolve conflicts when inconsistent information is received.


Ball blurry

Ball reflection

Color merging

Complex background

Multiple moving objects (ball and hand)

Multiple moving objects (ball and spectators)

Figure 1. Ball Detection challenging scenarios

To tackle this problem, a purposed-built Multi-Agent System (MAS) is proposed. A MAS consists of a number of inter-connected intelligent agents, which can either independently or jointly achieve goals. The main strengths of a MAS are: i) agents with various specialist knowledge can work together to complete complicated tasks; ii) it is a distributed structure that facilitates shared workloads via a network of computers; and iii) it is a modular design so the system is scalable [8]. This makes a MAS an ideal platform for the multi-view ball tracking application, where each camera can be controlled by ball detection agents, which possess the specialist knowledge in detecting balls from their views independently. Another agent can play the role of a coordinator controlling the flow of the information between the ball detection agents and resolve conflicts if conflicting information from different agents is received. The agents can also be run on a network of computers and this can enhance the system performance, especially if real-time tracking is required. When more cameras are needed by the tracking system, the system can be scaled up by simply adding more ball detection agents. Another important benefit is that the MAS software manages the networking and communication aspects of the system, so users do have to deal with these aspects. Furthermore, MASs have also been applied in other object tracking applications [9] and [10]. More details of the proposed MAS based ball tracking system are given in Section 3.1.

In terms of the color merging problem, a combined CT and motion-based *background subtraction* (BS) approach is adopted to segment the ball from the background. It dynamically adapts the color thresholds and jointly uses the CT and BS information to segment difficult objects [11]. To tackle sudden changes of trajectory, the proposed approach captures frames at 300 frames per second (fps). At this rate, the motion vector of the ball between two successive frames is relatively small and a simple second-order motion model can be used to model the trajectory. This approach also helps capture the fast traveling ball more clearly.

3.1. Camera Configuration and MAS Architecture

Although capturing the image of an object from multiple angles can increase the probability of detection, consideration has to be given to the design configuration of the multi-view system, i.e., how many cameras are employed, where each camera is placed and how they are paired with another camera to derive the 3D position of the object. To achieve a pragmatic balance between detection accuracy and cost, four cameras were used at the positions shown in Figure 2. They jointly track the ball spatially within an area comprising the table and players. Each camera covers approximately two thirds of the length of the table, but from different perspectives. In this arrangement, the main playing area is thus covered by four cameras, with the coverage overlapping around the net area, where particular attention is needed for umpiring. As each individual camera does not have to cover the entire table, the cameras can be placed closer to the objects of interest (e.g. ball and table) so that better depth resolution is achieved when deriving their 3D positions and the objects appear bigger in the views. The 3D positions of the objects can also be derived from either the opposite facing pairs or the side-by-side pairs using triangulation and geometry calculations. A limitation of this configuration is that when the position of the ball is at or near the vertical plane joining the principal points of the opposite facing cameras (see red and yellow dotted lines in Figure 2), the 3D positions cannot be determined using triangulation because the angle between the ball and the two cameras is zero. The position of the ball in these situations is thus extrapolated using a second-order equation of motion (SOEM) to model the trajectory of the table tennis ball during a rally.

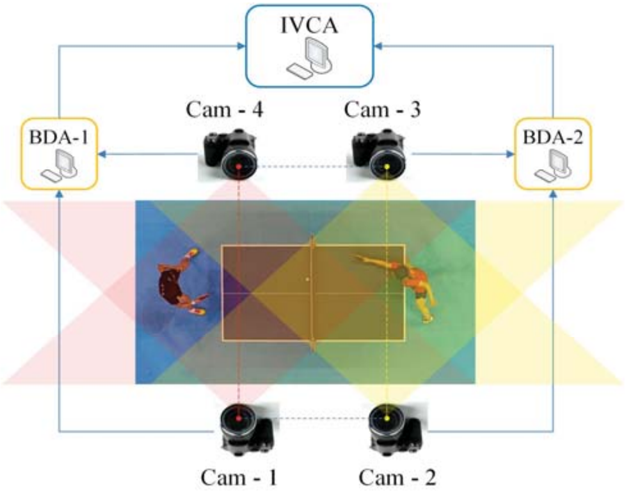


Figure 2. Multi-view camera setup

As shown in Figure 2, each opposite facing camera pair is connected to a *Ball Detection Agent* (BDA), which detects the two-dimensional (2D) position of the ball from each view. From this, the 3D real world position of the ball is derived and sent to the *Inter-View Correction Agent* (IVCA) for storage. IVCA checks the consistency of the ball's 3D position from multiple BDAs and uses this information to estimate the ball position. If a particular view gives an incorrect ball location, the corrected screen position of the ball is sent to the relevant BDA managing that view, for correction. As each BDA only monitors a portion of the table, the IVCA selectively instructs which BDA to report the ball position and which to hibernate. The IVCA then sends the 3D ball position to other agents (not shown in Figure 2 for clarity) for an umpiring decision.

The ball detection process employed by BDA uses a *modified Combined Adaptive Color Thresholding and Motion Detection* (MACTMD) algorithm [11], which exhibits robust ball detection but only in a conventional adjacent camera arrangement. Modifications have been introduced so the ball's 3D position can be derived from opposite-facing cameras. Also since ACTMD was written in C++ while the MAS framework employed is written in Java, a pipe based inter-process communication module was developed to bridge the employed MAS and modified ACTMD.

Figure 3 shows the connections and information flows between the MAS modules. At initialization, the IVCA instructs BDAs to report the ball position. If the ball is visible, the ball position is returned, otherwise the BDA states a "no ball" and is hibernated. BDAs obtain the ball position through a pipe connection to the MACTMD, which detects the ball from video signals from opposite-facing cameras. The process continues until the ball is detected in the overlapped zone where it is visible by both camera pairs (area near the middle of the table in Figure 2), at which time the IVCA will instruct the relevant BDA to wake up and report the ball position. When the ball eventually disappears from the view of the BDA which initially observed the ball, the IVCA instructs it to hibernate. This process continues until the end of the rally.

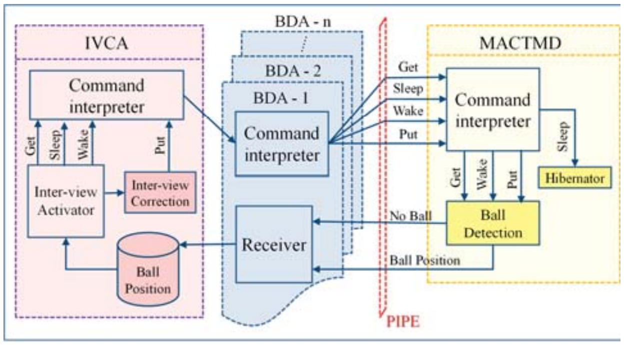


Figure 3. Architecture of the MAS

3.2. Compensating measuring errors

While the proposed camera configuration can provide robust ball detection, experimental results confirm that camera alignment is very challenging and time consuming. Any misalignment could cause measurement errors in the 3D position of the detected ball, thereby impacting upon the accuracy of the overall umpiring system. For this reason, an error model was developed to compensate for these errors, which are represented by 3-D vectors. Empirical analysis showed that the measurement error exhibited a non-linear relationship to the ball location, so quadratic surfaces were chosen for the error modelling, with data being fitted to surfaces using a standard Multivariate Polynomial Regression [12]. The error model equations are given in (1), where $\boldsymbol{E}(x, y, z)$ is the 3-D error vector, $F(x,y,z)$, $G(x,y,z)$, $H(x,y,z)$ are functions determining the magnitudes of the $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ components respectively, and (x, y, z) is the measuring ball location.

$$\boldsymbol{E}(x, y, z) = F(x,y,z)\boldsymbol{i},\ G(x,y,z)\boldsymbol{j},\ H(x,y,z)\boldsymbol{k} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

Thus the error model takes the 3D ball location as input and produces an error vector for that particular location.

Equations (2), (3) and (4) define the quadratic surfaces of $F(x,y,z)$, $G(x,y,z)$, $H(x,y,z)$ respectively, where a_n, b_n, c_n, d_n, e_n, f_n, g_n, h_n, i_n and j_n are coefficients of the surfaces, for $n = 1, 2$ and 3.

$$F(x,y,z)=a_1x^2+b_1y^2+c_1z^2+d_1xy+e_1xz+f_1yz+g_1x+h_1y+i_1z+j_1 \ldots\ldots\text{(2)}$$

$$G(x,y,z)=a_2x^2+b_2y^2+c_2z^2+d_2xy+e_2xz+f_2yz+g_2x+h_2y+i_2z+j_2 \ldots\ldots\text{(3)}$$

$$H(x,y,z)=a_3x^2+b_3y^2+c_3z^2+d_3xy+e_3xz+f_3yz+g_3x+h_3y+i_3z+j_3 \ldots\ldots\text{(4)}$$

For calibration purposes, a checker board which has 4 rows and 5 columns of identical sized black squares distributed evenly upon a white board, was employed. It was carefully placed at various known positions during sequence filming as shown in Figure 4. As the position of each square corner on the checkerboard can be easily calculated, this provided a rich set of reference points for training the error model.

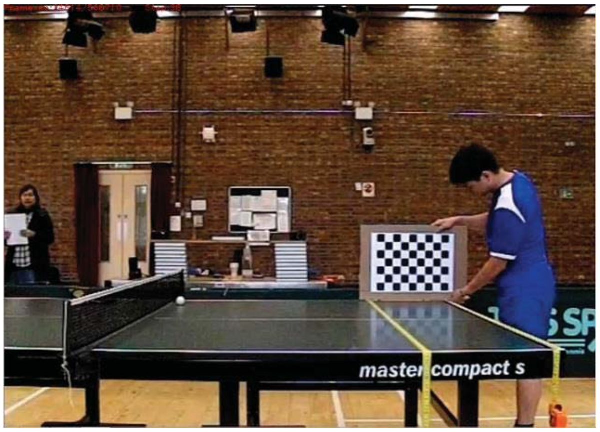
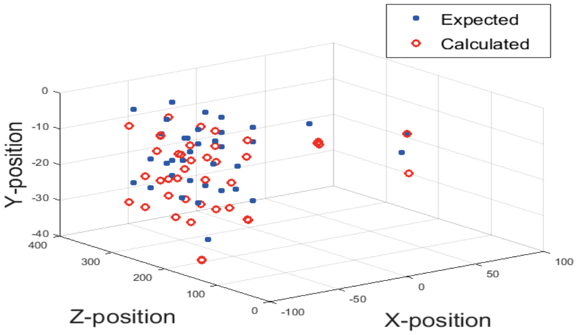
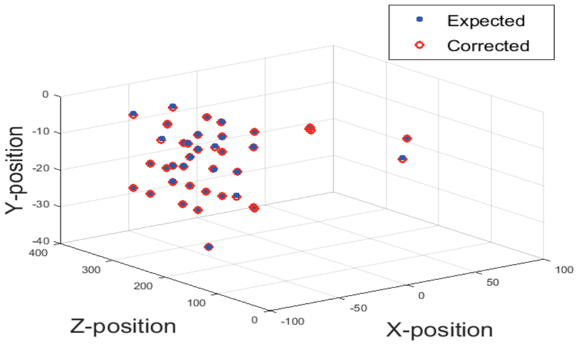


Figure 4 A double-sided checker board used for calibration

To prevent overfitting, a small subset of 32 points were randomly selected from some known positions on the checker board as training data and 45 "unseen" points were chosen for testing. Figure 5(a) shows the 45 uncompensated calculated (red) and expected (blue) ball locations, while Figure 5(b) shows the corrected ball locations after error compensation. It is evident the calculated and expected positions are much closer, with the average Euclidean distance between them being reduced from 4.8cm to only 0.1cm.



(a) Uncompensated ball positions



(b) Compensated ball positions

Figure 5. Expected and calculated 3D position of the ball

4. EXPERIMENTAL SETUP AND RESULTS

The MAS has been developed using a Java Agent DEvelopment Framework (JADE) [13], which is open-source and platform independent. It complies with the de-facto standard set by Foundation for Intelligent Physical Agents. The proposed system can be distributed across a network of computers and agents can be migrated from one machine to another if required. All experiments were conducted on a computer with an Intel® Core™ i7 CPU @ 2.80 GHz with the detecting and tracking algorithms implemented in C++ employing OpenCV [14].

To test the system, a 4-view match scene sequence was captured, consisting of 450 frames per view, with spatial resolution of 512×384 pixels and a capture rate 300 frames per second (fps). Both resolution and frame rates were limited by the entry-level high speed camera hardware. The sequence was of duration 1.5 seconds and consisted of a complete table tennis rally which included challenging ball detection conditions such as sudden changes in trajectory, occlusion, uneven illumination, multiple object motion and camera noise. The video is available at the Open University Table Tennis video database [15]. Furthermore, due to the high capturing rate and shutter speed, the video is of low contrast. Cyclic variations in illumination is also evident due to the higher capturing rate than the standard 50Hz used by the lights. As it is a complete rally, it consists of a service, ball traveling back and forth between the table and striking by the players, and an eventual foul. A summary of the characteristics of the test sequence is shown in Table 1. To evaluate the detection performance of the system, the detected 2D ball locations are reprojected to the 3D space and compared with the 3D reprojected ball locations identified by human volunteers.

Table 1 Characteristics of the test sequence

No. of view	4
No of frames per view	450 (for one complete rally)
Size of frame (pixels)	512×384
Capture rate	300 fps
Ball radius	4 pixels
Ball colour	White
Key detection Challenges	Complex background, color merging, ball reflection, multiple moving objects, occlusion, very small ball size

To establish a ground truth for detection, the set of ball locations were manually identified in each frame of each view (a total of 1800 locations) and then mapped into 3D space. As this is a very time consuming process, only one test sequence was produced, though this sequence does contain all the key events of a typical table tennis rally. An example frame from this sequence is shown in Figure 6, where the red circle indicates the detected ball position while the green square defines the region of interest (ROI) within which object detection takes place. Figure 7 shows the corresponding views for all four cameras and the detection results in the sample frame, with the ball clearly visible and successfully detected by all four cameras.

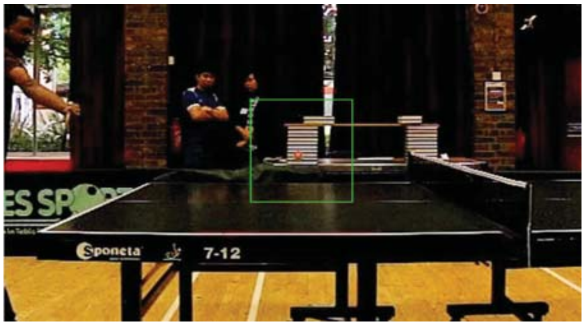


Figure 6. Ball detected against a very similar color background

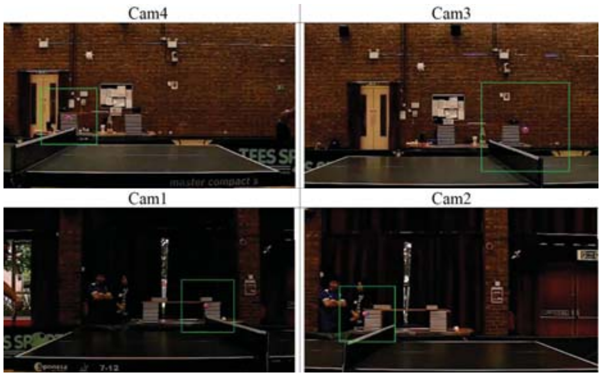
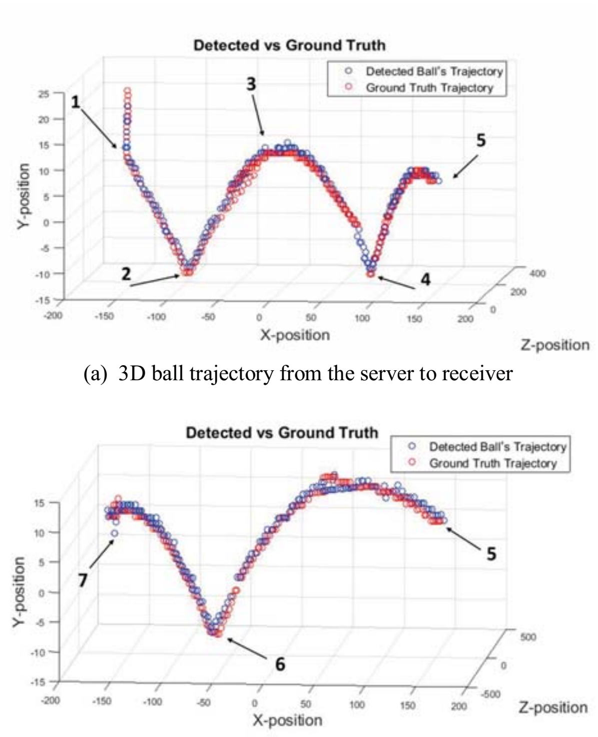


Figure 7. Example images of the tested sequence



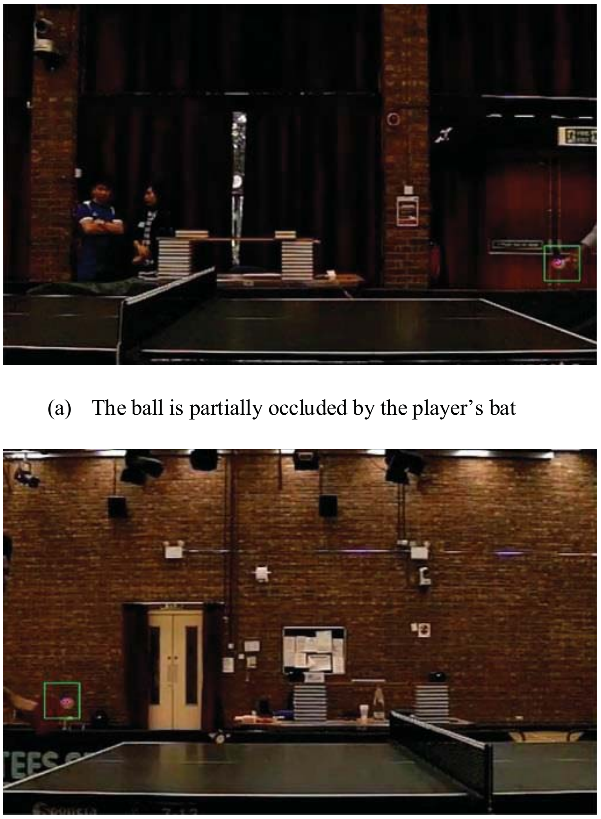
(a) 3D ball trajectory from the server to receiver

(b) 3D ball trajectory from the receiver to server

Labels:
1: Start of the service at Frame #1,
2: The ball bounces on the server side of the table
3: The ball crosses over the net
4: The ball bounces on the receiver side of the table
5: The ball is stroke by the receiver
6: The ball bounces on the server side of the table
7: The rally ends.

Figure 8. Trajectory comparison between the ground truth and the detected ball locations

Figure 8 shows the result comparison between the ground truth (red) and the detected ball locations (blue). The two trajectories almost overlap, indicating good ball detection accuracy, including crucially some of the detection challenges identified earlier. In particular, the system satisfactorily resolves problems like color merging and occlusion, by consistently detecting the ball. Figure 6 shows an example of these successful detections when the color of the background object (white) is very similar to the ball (color merging), while Figure 9 shows a successful detection when the ball is partially occluded in one of the views.



(a) The ball is partially occluded by the player's bat

(b) The ball is visible and detected by the corresponding opposite facing camera.

Figure 9 Examples of occlusion. The occluded ball is recovered using the inter-view correction method.

However, some errors do occur, especially in regions where the position of the ball is near the plane that joins the principal points of the opposite facing cameras. This means the ball positions estimated by the SOEM can be inaccurate when there are too much noise in the ball trajectory. Nevertheless, successful detections resume shortly after the ball passes that region.

The average Euclidean distance between the detected ball location and ground truth is 3 cm if the ball locations in the region near the plane that joins the principal points of the opposite facing cameras are omitted, otherwise it is 11 cm. The accuracy of umpiring is most important near the net or table edge, where the detection errors are relatively low.

The time taken to detect the ball in a frame is 2.8ms. However, if 4 BDA and an IVCA are employed instead and each agent runs on a separate computer, this time can be reduced to around 0.56ms. This means the full 450 frame test sequence (1.5 seconds) can be processed within 2.52 seconds. The time lag is only 1.02 seconds, which is an acceptable window for making an accurate umpiring decision.

5. CONCLUSION

This paper has presented a MAS-based ball tracking system, which is designed to be low-cost, portable and for umpiring purposes. A multi-view camera configuration was designed such that a minimum number of cameras were required yet it was able to cover a large area and provide enough video clarity for tackling the detection challenges. Furthermore, a measurement error correction model was derived and the model significantly reduced the detection errors. With no need to fix a camera to the ceiling, the system is portable, which is very important as most table tennis tournaments take place at multi-purpose sport venues where installation of fixed equipment is not permissible.

The ball tracking system has been tested on a complete rally sequence from a real match scene and the result is very promising. The detected ball locations are very similar to the ground truth. Although detection errors occur sometimes, the system is able to recover and resume successful detections. The errors could be reduced with a more accurate extrapolation model.

The agent-based design allows parallel computation and scalability. Although a single computer was used in these trials, individual agents can run on separate computers, thereby enabling real-time tracking. More cameras (and agents) can be added to the system without significantly changing the program or its performance.

6. REFERENCES

[1] K. C. P. Wong and L. S. Dooley, ‘High-motion table tennis ball tracking for umpiring applications’, in *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 2460–2463.

[2] G. Byrd, ‘21st Century Pong’, *IEEE*, vol. 48, no. 10, pp. 80–84, Oct. 2015.

[3] "Law of Table Tennis", International Table Tennis Federation Handbook, [Online]. Available: http://www.ittf.com/ittf_handbook/hb.asp?s_number=2 [Accessed: 22-Mar-2016].

[4] Z. Zhang, D. Xu, and M. Tan, ‘Visual Measurement and Prediction of Ball Trajectory for Table Tennis Robot’, *IEEE Trans. Instrum. Meas.*, vol. 59, no. 12, pp. 3195 –3205, Dec. 2010.

[5] J. Liu, Z. Fang, K. Zhang, and M. Tan, ‘Improved high-speed vision system for table tennis robot’, in *2014 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2014, pp. 652–657.

[6] X. Chen, Q. Huang, W. Zhang, Z. Yu, R. Li, and P. Lv, ‘Ping-pong trajectory perception and prediction by a PC based High speed four-camera vision system’, in *2011 9th World Congress on Intelligent Control and Automation (WCICA)*, 2011, pp. 1087 –1092.

[7] H. Bao, X. Chen, Z. Wang, M. Pan, and F. Meng, ‘Bouncing model for the table tennis trajectory prediction and the strategy of hitting the ball’, in *2012 International Conference on Mechatronics and Automation (ICMA)*, 2012, pp. 2002 – 2006.

[8] A.A. Hopgood, 'Intelligent Systems for Engineers and Scientists', 3rd edition, CRC Press, (2012), ISBN 9781439821206.

[9] M. Chakroun, A. Wali, and A. M. Alimi, ‘Multi-agent system for moving object segmentation and tracking’, in *2011 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011, pp. 424 –429.

[10] Y. Li, Y. Wang, Y. Qi, and H. Li, ‘Multi-agent based particle filter for moving object tracking’, in *2010 International Conference on Computer Application and System Modeling (ICCASM)*, 2010, vol. 4, pp. V4–124 –V4–128.

[11] H. Myint, P. Wong, L. Dooley, and A. Hopgood, ‘Tracking a table tennis ball for umpiring purposes’, in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, 2015, pp. 170–173.

[12] Rawlings, J.O., Pantula, S.G. and Dickey, D.A. “Applied Regression Analysis: A Research Tool” Second Edition, Springer-Verlag, 1998

[13] ‘JAVA Agent DEvelopment Framework’. [Online]. Available: http://jade.tilab.com/ [Accessed: 22-Mar-2016]

[14] ‘Open Source Computer Vision Library’. [Online]. Available: http://opencv.org/. [Accessed: 22-Mar-2016].

[15] ‘The Open University table tennis video database’. [Online]. Available: http://users.mct.open.ac.uk/xgmt/outtdb/. [Accessed: 22-Mar-2016].

A Cluster Method for Labelling Large Scale Vehicle Model Dataset based on Deep Learning

Yongbin Gao[1], Hyo Jong Lee [1, 2]
[1]Division of Computer Science and Engineering,
[2]Center for Advanced Image and Information Technology
Chonbuk National University, Jeonju 561-756, Korea

Abstract - *Vehicle make and model recognition is an important task for vehicle analysis, which can be used for automatic toll collection and public security. Vehicle make and model recognition (MMR) is a challenging task due to the close appearance between vehicle models. In this sense, MMR is a fine-grained classification task. To train a model for this fine-grained task, a large scale dataset is required. Labelling a large dataset with many classifies is difficult and tedious. In this paper, we proposed a cluster method incorporating with deep learning to assist faster labelling of a large scale vehicle make and model dataset. The proposed cluster method can automatically divide the images into groups, within each group, there is one or few classes. In addition, the cluster is performed in an incremental manner to refine the deep models. Experimental results show that our cluster method speedup the label task significantly compared with manually label of each image.*

Keywords: Cluster; Deep Learning; Vehicle Model Dataset.

1 Introduction

Deep learning is widely used in both academic and industry due to its impressive performance. In order to train deep network, large scale dataset is required, labelling of which is difficult and tedious. Thus, a cluster method is required for speedup the labelling process. For vehicle analysis, deep learning also achieved favorable performance [1], vehicle analysis is an essential component in many intelligent applications, such as automatic toll collection, driver assistance systems, self-guided vehicles, intelligent parking systems, and traffic statistics (vehicle count, speed, and flow). Specially, an electronic toll collection system can automatically collect tolls according to the identification of vehicle models. Also, the identification of vehicle models can provide valuable information to the police for searching suspect vehicles. The appearance of a vehicle will change under varying environmental conditions and market requirements. The shapes of vehicles between companies and models is very similar, which results in confusion in vehicle model recognition. This makes the vehicle model recognition a challenging task.

Vehicle detection is prerequisite of vehicle analysis. Background subtraction [2]–[5] is widely used to extract motion features to detect moving vehicles from videos. However, this motion feature is not available for still images. To address this problem, Wu et al. [6] proposed the use of wavelet transformation to extract texture features to locate possible vehicle candidates. Tzomakas and Seelen [7] found that the shadow of a vehicle is a good cue for detecting vehicles. Ratan et al. [8] localize the possible vehicles based on the detection of vehicle wheels and verify the candidate vehicle by a diverse density method.

The commonly used deep networks are convolutional neural network (CNN) [9] and convolution RBM [10]. In addition, many advanced techniques have been coupled into the CNN structure, such as dropout, maxout, max-pooling. By going deeper with the convolutional networks, CNN dominates the performance in various applications, such as AlexNet [11], Overfeat [12], GoogLeNet [13], and ResNet [14].

In this paper, we proposed a cluster method incorporating with deep learning to assist faster labelling of a large scale vehicle make and model dataset. The proposed cluster method can automatically divide the images into groups, within each group, there is one or few classes. In addition, the cluster is performed in an incremental manner to refine the deep models.

The remainder of this paper is organized as follows. Section II describes the framework of our designed system. We then introduce the cluster method based on deep learning in Section III. Section IV applies the above algorithm to cluster our car database, and presents the experiment results. Finally, we conclude this paper in Section V.

2 Framework of our system

Cluster algorithm aims to differentiate the images into groups so that the distance of different groups is maximum. Among which, K-means is efficient and suitable for large scale dataset. However, if we apply the K-means directly on the raw images, the accuracy is not favorable since the raw images are not discriminative, in addition, the computational time is high since raw images are of large size. Thus, in this study, we use the deep learning to first extract discriminative and simpler features before using the K-means. In order to use deep learning for feature extraction, we introduce a third-party dataset, which may have less number of images and car models

[1]. The framework of our system is shown in Fig. 1. We first detect the vehicle by frame difference and symmetrical filter, the frame difference is performed on image by shifting one image with moderate pixels to generate another image, the difference of these two images are used to detect vehicle by a symmetrical filter, which take use of the symmetrical structure of the vehicles. The third-party dataset is labeled, and used for training the deep network. The trained model is used to extract features of the unlabeled vehicle. The features is still high dimension for k-means, thus, we used principal component analysis (PCA) to reduce the dimension. Finally, the k-means is used to cluster the data and assign the group for each data. After that, the manual correction of the wrong clustered data is performed to get a new dataset, this dataset is add to the third-party dataset to train a more powerful deep networks. In this sense, the labeling is performed iteratively. Each iteration, we use 100,000 images as an incremental data.

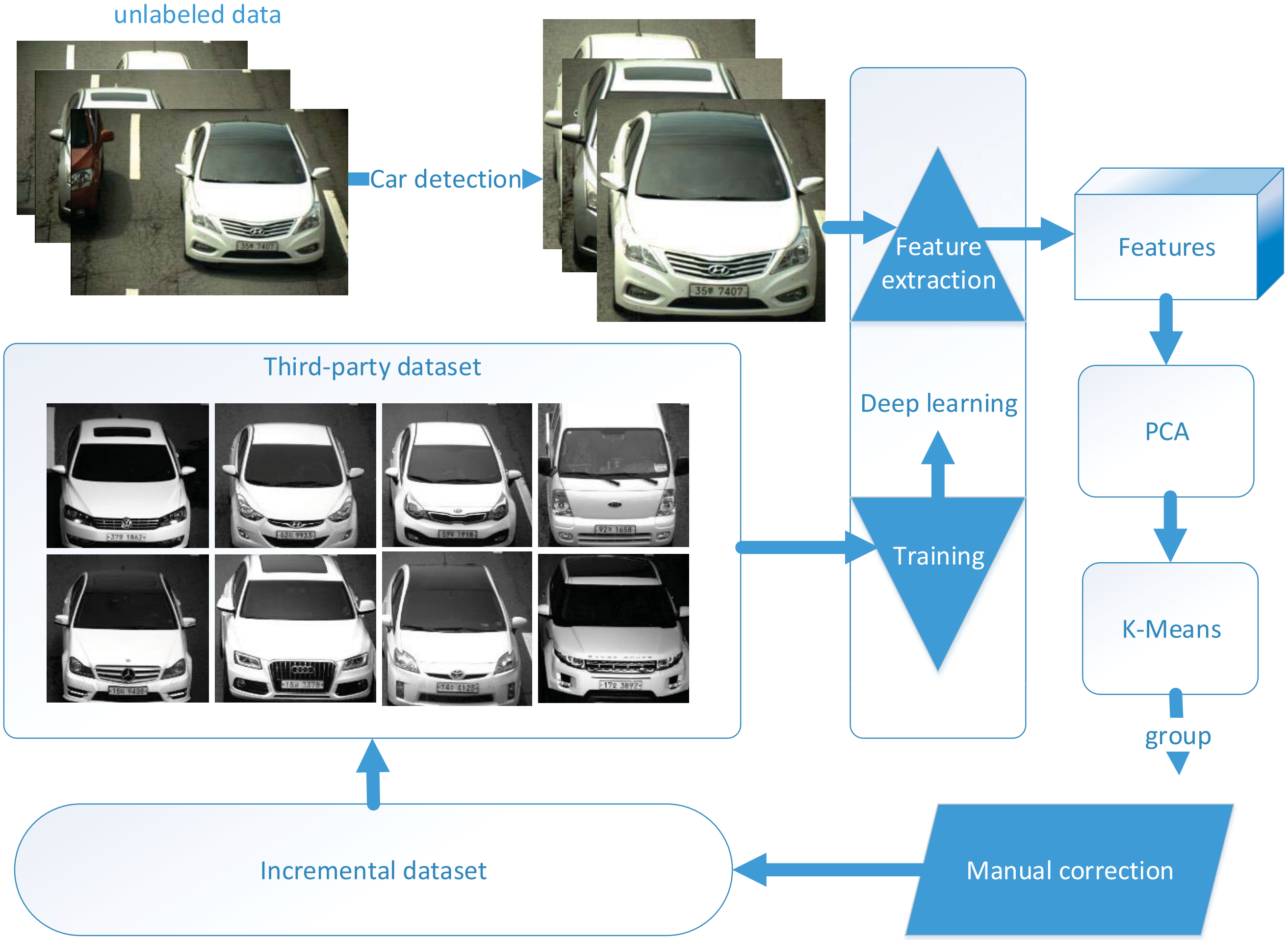


Fig. 1. Framework of proposed car detection and cluster method.

3 Cluster based on Deep Learning

This section shows the three principal algorithms in the framework of Fig. 1, which is deep learning network, PCA, and K-Means.

3.1 Deep network

The architecture of our proposed algorithm is shown in Fig. 2, which is referred as to Alexnet [11]. Each layer was marked by its type and name, Alexnet mainly consists of five convolutional layers and three innerproduct layers. The Relu layer and Max-pooling layer are added to provide the nonlinearity and translation invairnace. Some of the layers are interpreted in detail as following sections.

A. Convolutional Layer

Convolutional layer is the main layer of the network, where a kernel is defined to filter the input data. The input data is filtered by various kernels and results in different feature map. Support that the size of input image is $N \times N$, and we use a $m \times m$ filter as kernel and the number of resultant feature map is K, the k-th feature map at a given layer h^k is calculated as follows:

$$h_{ij}^{k} = \tanh((W^k * x)_{ij} + b_k) \qquad (1)$$

where W^k and b_k are the weights and bias of k-th feature map, the size of resultant feature map is N-m+1. The tanh(.) is a non linear function to fulfil the non-linearity property of convolutional neural network. In this way, the input image is characterized by different feature map using different filter.

Through layer by layer convolutional operation, we are able to learn progressive level of features. For example, the first layer is low-level features, such as edges, lines and corners.

B. Max-pooling Layer

In order to increase the robustness of CNN to handle translations, pooling layer is usually used after the convolutional layer. The max value or the average value of a local feature map is widely used as a pooling technical. In our architecture, we use max-pooling layer to ensure that the same result can be achieved even there are translations existed. Support that the size of local region is $n \times n$, the output layer size will be $\frac{N}{n} \times \frac{N}{n}$.

C. Relu Layer

The Relu layer is used to gain the non-linearity of the network. This layer uses the non-saturating function as $f(x) = max(0, x)$, which has the non-linear property and has no affection of the receptive fields of the convolution layer.

D. Softmax Loss Layer

Softmax loss layer is used to predict the probability of K mutually exclusive classes. The softmax loss layer is usually designed at the final layer as follows:

$$\mathcal{L}(y, z) = -\log\left(\frac{e^{z_y}}{\sum_{j=1}^{m} e^{z_j}}\right) \qquad (2)$$

where y and z are the class and predicted value of input data, respectively. $\mathcal{L}(y, z)$ is the probability of predicted value to be class y.

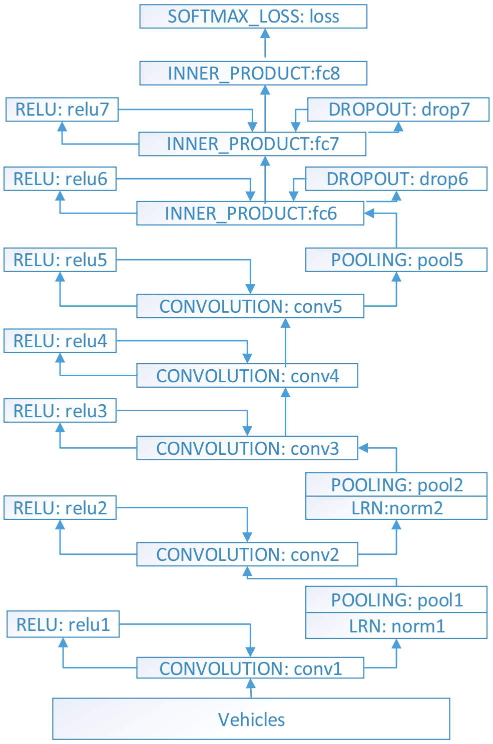


Fig. 2 Deep network architecture.

3.2 Principal Component Analysis

The extracted features from deep networks are further fed into the PCA, PCA is a popular algorithm for dimensional deduction as well as feature extraction, which acquainted us with its successful application to face recognition using eigenfaces [16]. We try to learn the principal component of the frontal view of a car prior to feeding it into a deep network.

Let the training set be $\mathrm{T}_1, \mathrm{T}_2, \dots \mathrm{T}_M$, where each image with size $N * N$ is unrolled to a vector T_i of dimension N^2. The average of training images is calculated as $\overline{T} = \frac{1}{M}\sum_{n=1}^{M} T_n$, each training image is deducted from the average resulting in a vector $D_i = \mathrm{T}_i - \overline{T}$. The covariance matrix of the difference image is calculated as follows:

$$C = \frac{1}{M}\sum_{i=1}^{M} D_i\, D_i^T \qquad (3)$$

Let e_k and λ_k be the eigenvector sand eigenvalue of the covariance matrix C, respectively. By ranking the eigenvalues, we are able to see the efficacy of their associated eigenvector in handling the image variation. A projection matrix P_K consisting of K eigenvectors is generated by seeking the largest K associated eigenvalues. This projection matrix enables us to transform the original N^2 image space to K-dimension subspace, which is a better representation of original image. As for a new image T, it can be projected into the subspace y as follows:

$$y = (T - \overline{T})\, P_K \qquad (4)$$

3.3 K-Means

The dimension of features is reduced significantly after the PCA. Suppose that the principal component of features are $(f_1, f_2, \dots, f_n)$, k-means cluster aims to partition the n features into k sets $s = \{s_1, s_2, \dots, s_k\}$, so as to minimize the winthin-cluster sum of squares, which can be depicted as:

$$\mathrm{argmin}_s \sum_{i=1}^{k} \sum_{x \in s_i} \|f - u_i\|^2 \qquad (5)$$

Where u_i is the mean of points in s_i. This can be solved by alternating the assignment step and update step, in which, assignment step assign each feature to the cluster whose mean yield the least within-cluster sum of squares, and the update step calculate the new means to be the centroids of the features in the new clusters. This idea is simple and suitable for large scale cluster task.

4 Results

As we stated that we need to use a third-party dataset to training a start model for feature extraction. For vehicle make and model dataset, we chosen “compcars” as a third-party dataset. This dataset is relatively larger to train a deep network for MMR, which consists of a surveillance-nature set and a web-nature set. Our study focus on the surveillance data, while the “compcars” contain only 44,481 images in 281 different models, which is not enough for practical use. Thus, we collect 304,447 images from the surveillance camera that set up on the real street, the collection time spans 2 years, we divide the images into two set, 152303 (set 1),152144 (set 2) images for each subset. We first use the “compcars” images to train a deep model, and extract features for the set 1, after clustering this set, we use these images to train a new model. The new model is used to cluster the set 2. Fig. 3 shows some example of clustered images. These images contains many kinds of variation, such as runny/sunny, daytime/nighttime, illumination, and partial missing due to the car detection failure. As a result, a large scale dataset containing 304,447 images has been labelled, which consists of 655 models. The accuracy of cluster of set 2 reached 94%, which means the second iteration of cluster reduced the manual label work of 94%. Thus, our cluster speedup the label work significantly.

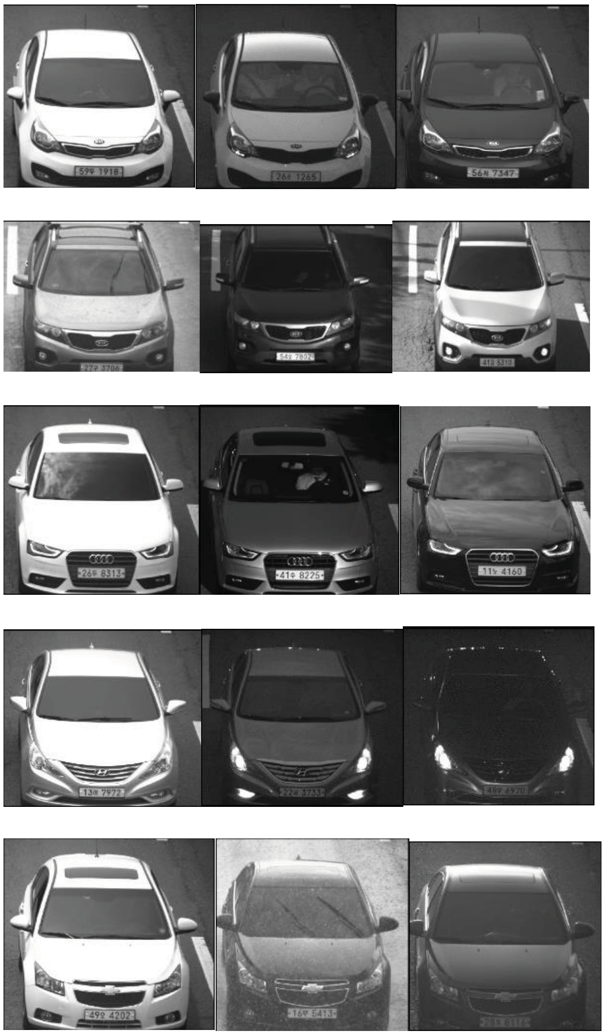


Fig. 3 Examples of clustered images

5 Conclusions

In this paper, we proposed a cluster method incorporating with deep learning to assist faster labelling of a large scale vehicle make and model dataset. The proposed cluster method can automatically divide the images into groups, within each group, there is one or few classes. In addition, the cluster is performed in an incremental manner to refine the deep models. Experimental results show that our cluster method speedup the label task significantly compared with manually label of each image. As a result, a large scale dataset containing 304,447 images has been labelled.

Acknowledgment: This work was supported by the Brain Korea 21 PLUS Project, National Research Foundation of Korea. This work was also supported by the Business for Academic-Industrial Cooperative establishments that were funded by the Korea Small and Medium Business Administration in 2015 (Grants No. C0221114). This research was also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R0992-16-1023) supervised by the IITP (Institute for Information & communications Technology Promotion).

6 References

[1] L. Yang, L. Ping, C. L. Chen, and X. Tang. "A large-scale car dataset for fine-grained categorization and verification," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3973-3981. 2015.

[2] A. Faro, D. Giordano, and C. Spampinato. "Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection"; IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1398–1412, Dec. 2011.

[3] H. Unno, K. Ojima, K. Hayashibe, and H. Saji. "Vehicle motion tracking using symmetry of vehicle and background subtraction"; in Proc. IEEE Intell. Veh. Symp., 2007, pp. 1127–1131.

[4] A. Jazayeri, H.-Y. Cai, J.-Y. Zheng, and M. Tuceryan. "Vehicle detection and tracking in car video based on motion model"; IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 583–595, Jun. 2011.

[5] G. L. Foresti, V. Murino, and C. Regazzoni. "Vehicle recognition and tracking from road image sequences"; IEEE Trans. Veh. Technol., vol. 48, no. 1, pp. 301–318, Jan. 1999.

[6] J. Wu, X. Zhang, and J. Zhou. "Vehicle detection in static road images with PCA-and-wavelet-based classifier"; in Proc. IEEE Conf. Intell. Transp. Syst., Aug. 25–29, 2001, pp. 740–744.

[7] C. Tzomakas and W. Seelen. "Vehicle detection in traffic scenes using shadow"; Inst. Neuroinf., Ruhtuniv., Bochum, Germany, Tech. Rep. 98-06, 1998.

[8] A. Lakshmi Ratan,W. E. L. Grimson, and W. M.Wells. "Object detection and localization by dynamic template

warping"; Int. J. Comput. Vis., vol. 36, no. 2, pp. 131–148, Feb. 2000.

[9] Y. LeCun, L. Bottou, Y. Bengio, Haffner, P, "Gradient based learning applied to document recognition," Proceeding of the IEEE, 1998.

[10] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations," In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009.

[11] K. Alex, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," In Proceedings of the Advances in neural information processing systems, Lake Tahoe, NV, USA, 3–6 December 2012.

[12] P. Sermanet, D. Eigen, X. Zhang, "Overfeat: Integrated recognition, localization and detection using convolutional networks," Available online: http://arxiv.org/abs/1312.6229 (accessed on 21 Dec 2013)

[13] C. Szegedy, W. Liu, Y. Jia, "Going deeper with convolutions," Available online: http://arxiv.org/abs/1409.4842 (accessed on 17 Sep 2014).

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition," Available online: http://arxiv.org/abs/1512.03385, 2015 (accessed on 10 Dec 2015).

Human Action Recognition using Improved Vector of Locally Aggregated Descriptors

Shi-Ping Yang and Jin-Jang Leou
Department of Computer Science and Information Engineering
National Chung Cheng University, Chiayi 621, Taiwan, Republic of China
E-mail: {ysp102m, jjleou}@cs.ccu.edu.tw

Abstract – *Recently, two high-dimensional encoding techniques for human action recognition, namely, Fisher vector (FV) and vector of locally aggregated descriptors (VLAD), are widely employed. In this study, a new human action recognition approach using improved VLAD with localized soft assignment (LSA) and second-order statistics is proposed. When encoding videos into VLAD, instead of considering only the nearest one, we utilize localized soft assignment, i.e., considering multiple nearest visual words. In general, LSA-VLAD captures only the first-order statistics of descriptors and visual words. In this study, LSA and second-order statistics are encoded into VLAD-like form, namely, LSA2-VLAD. Based on the experimental results obtained in this study, in terms of average accuracy, the performance of the proposed approach combining LSA-VLAD and LSA2-VLAD is better than those of 10 comparison approaches.*

Keywords: human action recognition, Fisher vector (FV), vector of locally aggregated descriptors (VLAD), localized soft assignment (LSA), bag-of-visual-words (BOVW), feature encoding, action representation.

1 Introduction

In general, human actions are important contents in videos and how to correctly recognize human actions in videos becomes a popular research issue in computer vision applications [1]. However, due to cluttered background, complex scenes, camera motions, illumination changes, and viewpoint variations, recognizing human actions in realistic videos is not an easy task.

Human action recognition includes multiple/single-view [2], egocentric [3], 3-D [4-5] and 2-D tasks. In this study, we focus on 2-D human action recognition. Many human action recognition approaches, based on the bag-of-visual-words (BOVW) model, usually transform video clips from low-level descriptors to global representations by six steps, namely, feature extraction, codebook generation, feature encoding, pooling, normalization, and classification [6].

Recently, two high-dimensional encoding techniques for human action recognition, namely, Fisher vector (FV) and vector of locally aggregated descriptors (VLAD), are widely employed. Fisher vector (FV) [7-8], a super vector-based encoding method, contains the high-order statistics, i.e., the mean and variance of assigned descriptors for each visual word. VLAD [9], as image representation encoding for image retrieval, can be treated as a simplified version of FV, since it only contains the first-order statistics by accumulating the differences between descriptors and their nearest visual words. When encoding videos into VLAD, traditional VLAD considers only the nearest one. In this study, we utilize localized soft assignment (LSA), i.e., we consider multiple nearest visual words. In general, LSA-VLAD captures only the first-order statistics of descriptors and visual words, i.e., the difference between the descriptor and the corresponding nearest visual word. In this study, LSA and second-order statistics are encoded into VLAD-like form, namely, LSA2-VLAD.

This paper is organized as follows. The proposed human action recognition approach using improved VLAD with localized soft assignment and second-order statistics is described in Section 2. Experimental results are presented in Section 3, followed by concluding remarks.

2 Proposed approach

As illustrated in Fig. 1, the proposed approach contains six stages. (1) Extract improved dense trajectory (IDT) features [10, 11], including histogram of oriented gradients (HOG), histogram of optical flow (HOF), and motion boundary histogram (MBH). (2) Reduce dimensionality of each type of descriptors with PCA-whitening (3) Learn codebooks from each type of descriptors by *K*-means. (4) Adopt localized soft assignment (LSA) to encode first-order and second-order statistics of descriptors and visual words as VLAD representations: LSA-VLAD and LSA2-VLAD, respectively. (5) Normalize LSA-VLAD and LSA2-VLAD by intra-, power-, and L2- normalizations. (6) Use linear support vector machine (SVM) with one-versus-all training for training and classification.

In this study, we will extract improved dense trajectory (IDT) features [11]. To obtain a dense trajectory (DT), feature points are sampled densely from each video frame with 5-pixel step size over 8 spatial scales. On each spatial scale, sampled point $p_t = (x_t, y_t)$ at frame *t* is tracked to $p_{t+1} = (x_{t+1}, y_{t+1})$ at frame *t*+1 by median filtering in dense optical flow field $O = (u_t, v_t)$ as

The work was supported in part by Ministry of Science and Technology, Taiwan, Republic of China under Grant MOST 102-2221-E-194-041-MY3.

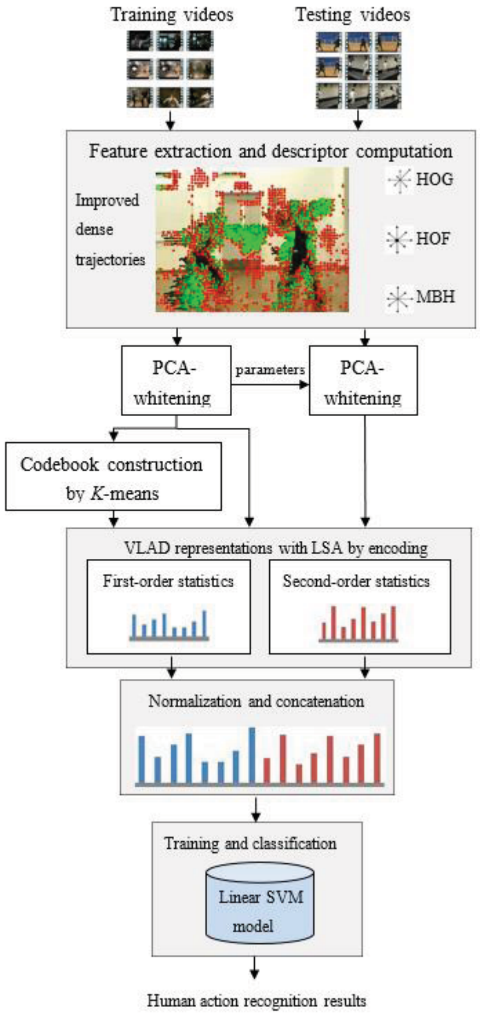

Training videos
Testing videos
Feature extraction and descriptor computation
Improved dense trajectories
HOG
HOF
MBH
PCA-whitening
parameters
PCA-whitening
Codebook construction by K-means
VLAD representations with LSA by encoding
First-order statistics
Second-order statistics
Normalization and concatenation
Training and classification
Linear SVM model
Human action recognition results

Fig. 1. The framework of the proposed approach.

$$p_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * O)\Big|_{(x_{t+1}, y_{t+1})}, \quad (1)$$

where M is a 3×3 pixel median filtering kernel and $*$ denotes the convolution operator. The tracking length L is empirically set to 15 frames and a dense trajectory is formed by the positions of tracked points in subsequent frames. To obtain IDT, camera motion is considered by estimating frame-to-frame homography and using human detector to exclude human-match features. Using estimated homography, the optical flow can be re-computed and camera motion-generated trajectories can be removed.

Local descriptors are computed along each IDT within an $N \times N$ pixel spatio-temporal volume, which is divided into $n_\sigma \times n_\sigma \times n_\tau$ cells, where $N = 32$, $n_\sigma = 2$, and $n_\tau = 3$ are default settings. An IDT is described by histogram of oriented gradients (HOG), histogram of optical flow (HOF), and motion boundary histogram (MBH). HOG captures the static information of local appearance and shape by quantizing intensity gradient orientations and magnitudes into 8-bin histogram. HOF captures local motion information of optical flow by quantizing flow vector into 9-bin histogram. As similar to HOF, MBH captures local motion information in optical flow field. In this study, the MBH descriptor is split into MBHx and MBHy descriptors and processed separately.

Because the original descriptors are often noisy and highly correlated, in this study, principal component analysis (PCA) and whitening [6] are adopted to decorrelate descriptors and perform dimensionality reduction. Principal component analysis (PCA) uses an orthogonal transformation to map a set of possibly correlated descriptors to a lower-dimensional space described by a set of linearly uncorrelated descriptors. Whitening is a decorrelation transformation, which ensures that all dimensions of a descriptor have the same variance. In this study, four types of descriptors, namely, HOG, HOF, MBHx, and MBHy, are PCA-whitened individually. The dimensionalities of HOG, HOF, MBHx, and MBHy are reduced by a factor of 1/4 from 96, 108, 96, 96 to 72, 81, 72, 72, respectively.

Before encoding descriptors into representations, we construct a codebook composed by a set of discriminative reference descriptors called "visual words." A codebook is constructed for each type of descriptors separately and visual words are learned from a large set by using the K-means clustering algorithm.

In this study, localized soft assignment (LSA) and vector of locally aggregated descriptors (VLAD) encoding are integrated to encode each type of descriptors into multi-representations: LSA-VLAD and LSA2-VLAD, by leveraging first-order and second-order statistics, respectively. Let $f = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m] \in \mathbb{R}^{d \times m}$ denote a descriptor set of a video clip consisting of m descriptors, d be the dimensionality of descriptor $\mathbf{f}_i$, and $C = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K] \in \mathbb{R}^{d \times K}$ denote a codebook with K visual words. The VLAD representation of a video clip is obtained by aggregating the differences between descriptors and the assigned visual words as

$$\boldsymbol{V}^{(1)} = \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} [\boldsymbol{r}_{\boldsymbol{i}1}(\mathbf{f}_{\boldsymbol{i}} - \mathbf{c}_1), \boldsymbol{r}_{\boldsymbol{i}2}(\mathbf{f}_{\boldsymbol{i}} - \mathbf{c}_2), \ldots, \boldsymbol{r}_{\boldsymbol{iK}}(\mathbf{f}_{\boldsymbol{i}} - \mathbf{c}_{\boldsymbol{K}})], \quad (2)$$

where $V^{(1)} \in \mathbb{R}^{d \times K}$ and r_{ik} performing on $\mathbf{c}_{\boldsymbol{k}}$ and its nearest visual word of $\mathbf{f}_{\boldsymbol{i}}$, is defined as

$$\boldsymbol{r}_{\boldsymbol{ik}} = \begin{cases} 1, \text{if } \boldsymbol{i} = \arg\min\limits_{\boldsymbol{j}} \left\| \mathbf{f}_{\boldsymbol{j}} - \mathbf{c}_{\boldsymbol{k}} \right\|_2^2, \\ 0, \text{otherwise.} \end{cases} \quad (3)$$

Inspired by [6], VLAD encoding with localized soft assignment (LSA) is employed. Here, $\boldsymbol{r}_{\boldsymbol{ik}}$ in Eq. (3) is redefined as

$$\boldsymbol{r}_{\boldsymbol{ik}} = \begin{cases} 1, \ \text{if } \mathbf{c}_{\boldsymbol{k}} \text{ is the } \boldsymbol{r}^{\boldsymbol{th}} \text{ visualword of } \mathbf{f}_{\boldsymbol{i}} \text{ and } \boldsymbol{r} \leq \boldsymbol{NN}_1, \\ 0, \ \text{otherwise,} \end{cases} \quad (4)$$

where $\boldsymbol{NN}_1$, the number of nearest visual words corresponding to LSA-VLAD, is empirically set to 5. Then, the LSA-VLAD representation $\boldsymbol{V}^{(1)}$ of a video clip can be obtained by Eqs. (2) and (4).

Similar to VLAD, LSA-VLAD keeps only the first-order statistics, i.e., the differences between descriptors and the corresponding visual words. Inspired by Peng et al. [12], in this study, we integrate second-order statistics with LSA-VLAD, namely, LSA2-VLAD. Note that a ranked attribute weighting scheme, namely, rank-order centroid (ROC) weight [13] is employed. The LSA2-VLAD representation $\boldsymbol{V}^{(2)}$ of a video clip is obtained by aggregating the weighted differences between the diagonal covariance of visual word clusters and the squared errors of assigned descriptors as

$$\boldsymbol{V}^{(2)} = \sum_{i=1}^{m}[w_{i1}((\mathbf{f}_i-\boldsymbol{\mu}_1)^2-\sigma_1^2), w_{i2}((\mathbf{f}_i-\boldsymbol{\mu}_2)^2-\sigma_2^2),..., w_{iK}((\mathbf{f}_i-\boldsymbol{\mu}_K)^2-\sigma_K^2)], \quad (5)$$

where $\boldsymbol{V}^{(2)} \in \mathbb{R}^{d\times K}$, $\boldsymbol{\mu}_k$ and σ_k^2 are the mean and the diagonal elements of covariance matrix of the k^{th} cluster, respectively, and $\boldsymbol{w_{ik}}$, a weighting function combining ROC weight and LSA, is defined as

$$\boldsymbol{w_{ik}} = \begin{cases} (\frac{1}{NN_2})\sum_{j=r}^{NN_2}\frac{1}{j}, & \text{if } \mathbf{c}_k \text{ is the } r^{th} \text{ visual word of } \mathbf{f}_i \text{ and } r \le NN_2, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $\boldsymbol{NN_2}$, the number of nearest visual words corresponding to LSA2-VLAD, is empirically set to 10.

In this study, three normalizations, namely, intra-, power-, and L2-normalizations, are successively applied on all VLAD vectors. If $\boldsymbol{V} = [v_1, v_2,..., v_n] \in \mathbb{R}^n$ denotes a raw VLAD vector, intra-normalization is defined as

$$\widetilde{\boldsymbol{V}} = [\mathbf{v}^1/\|\mathbf{v}^1\|_2, \mathbf{v}^2/\|\mathbf{v}^2\|_2,..., \mathbf{v}^K/\|\mathbf{v}^K\|_2], \quad (7)$$

where $\mathbf{v}^{\boldsymbol{k}}$ denotes the sum-of-differences vector of the $\boldsymbol{k}^{th}$ visual word and $\|\cdot\|_2$ denotes L2-norm. In power-normalization, vector $\boldsymbol{V} = [v_1, v_2,..., v_n]$ is component-wise normalized into $\widetilde{\boldsymbol{V}} = [\widetilde{v}_1, \widetilde{v}_2,..., \widetilde{v}_n]$ by

$$\widetilde{v}_i = \boldsymbol{sign}(v_i)|v_i|^{\alpha}, \quad (8)$$

where $0 \le \alpha < 1$ is empirically set to 0.5. In L2-normalization, vector $\boldsymbol{V} = [v_1, v_2,..., v_n]$ is divided by its L2-norm as

$$\widetilde{\boldsymbol{V}} = \boldsymbol{V}/\|\boldsymbol{V}\|_2 = \boldsymbol{V}/\sqrt{\sum_{i=1}^{n} v_i^2}. \quad (9)$$

Both LSA-VLAD $\boldsymbol{V}^{(1)}$ and LSA2-VLAD $\boldsymbol{V}^{(2)}$ of each descriptor type are successively normalized by intra-, power-, and L2-normalizations. By representation concatenation, the final representation $\boldsymbol{V_{Final}}$ of a video clip is obtained as

$$\begin{aligned} \boldsymbol{V_{Final}} &= [\boldsymbol{V}^{(1)}; \boldsymbol{V}^{(2)}] \\ &= [\boldsymbol{V}^{(1)}_{\boldsymbol{HOG}}; \boldsymbol{V}^{(1)}_{\boldsymbol{HOF}}; \boldsymbol{V}^{(1)}_{\boldsymbol{MBHx}}; \boldsymbol{V}^{(1)}_{\boldsymbol{MBHy}}; \boldsymbol{V}^{(2)}_{\boldsymbol{HOG}}; \boldsymbol{V}^{(2)}_{\boldsymbol{HOF}}; \boldsymbol{V}^{(2)}_{\boldsymbol{MBHx}}; \boldsymbol{V}^{(2)}_{\boldsymbol{MBHy}}], \quad (10) \end{aligned}$$

where *D* is the sum of dimensionalities of HOG, HOF, MBHx, and MBHy after PCA-whitening.

Finally, we use a linear support vector machine (SVM) with one-versus-all training for training and classification.

3 Experimental results

The proposed approach is implemented on an Intel Core i7-4790K 4GHz CPU with 32GB main memory for Linux / Ubuntu 14.04 LTS / 64-bit platform using MATLAB 8.0.1 (R2013a). The performance of the proposed approach is evaluated on a human motion database (HMDB51) [14], which contains 6766 video clips (51 action classes). The 51 action classes can be grouped into five types, including (1) general facial actions, (2) facial actions with object manipulation, (3) general body movements, (4) body movements with object interaction, and (5) body movements for human interaction.

The results of VLAD-*k* proposed by Peng et al. [6] are chosen as our baseline. In terms of average accuracy (%), performance comparison between the proposed approach and Peng et al.'s approach [6] on HMDB51 is shown in Table 1, whereas performance comparison between the proposed approach and 10 FV-based and VLAD-based approaches on HMDB51 is shown in Table 2.

Table 1. In terms of average accuracy (%), performance comparison between the proposed approach and Peng et al.'s approach [6] on HMDB51.

	Peng et al. [6]	Proposed
HOG	39.30%	45.62%
HOF	49.00%	51.94%
MBHx	43.03%	44.07%
MBHy	47.02%	49.96%
All descriptors	60.09%	**61.24%**

Table 2. In terms of average accuracy (%), performance comparison between the proposed approach and 10 FV-based and VLAD-based approaches on HMDB51.

	Approaches	**Average Accuracy**
FV-based	Wang and Schmid [11]	57.20%
	Heilbron et al. [15]	59.20%
	Murthy et al. [16]	59.40%
	Peng et al. [17]	59.70%
	Peng et al. [6] (FV+SVC-*k*)	61.10%
VLAD-based	Murthy and Goecke [18]	49.90%
	Jain et al. [10]	52.10%
	Wu et al. [19]	56.36%
	Peng et al. [12]	59.80%
	Peng et al. [6] (VLAD-*k*)	60.09%
	Proposed	**61.24%**

4 Concluding remarks

In this study, a human action recognition approach using improved VLAD with localized soft assignment and second-order statistics is proposed. Based on the experimental results obtained in this study, in terms of average accuracy, the performance of the proposed approach is better than those of 10 comparison VLAD-based and FV-based approaches.

5 References

[1] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2009, pp. 1996–2003.

[2] A. A. Liu, Y. T. Su, P. P. Jia, Z. Gao, T. Hao, and Z. X. Yang, "Multiple/single-view human action recognition via

part-induced multitask structural learning," IEEE Trans. on Cybernetics, vol. 45, no. 6, pp. 1194–1208, June 2015.

[3] Y. Yan, E. Ricci, G. Liu, and N. Sebe, "Egocentric daily activity recognition via multitask clustering," IEEE Trans. on Image Processing, vol. 24, no. 10, pp. 2984–2995, Oct. 2015.

[4] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bombo, "3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold," IEEE Trans. on Cybernetics, vol. 45, no. 7, pp. 1340–1352, July 2015.

[5] N. C. Tang, Y. Y. Lin, J. H. Hua, S. E. Wei, M. F. Weng, and H. Y. M. Liao, "Robust action recognition via borrowing information across video modalities," IEEE Trans. on Image Processing, vol. 24, no. 2, pp. 709–723, Feb. 2015.

[6] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: comprehensive study and good practice," in Computing Research Repository of Computer Vision and Pattern Recognition, arXiv:1405.4506, 2014, pp. 1–22.

[7] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition, " in Proc. of Asian Conf. on Computer Vision, 2012, pp. 572–585.

[8] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked Fisher vectors," in Proc. of European Conf. on Computer Vision, 2014, pp. 581–595.

[9] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311.

[10]M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2013, pp. 2555–2562.

[11] H. Wang and C. Schmid, "Action recognition with improved trajectories," in Proc. of IEEE Int. Conf. on Computer Vision, 2013, pp. 3551–3558.

[12] X. Peng, L. Wang, Y. Qiao, and Q. Peng, "Boosting VLAD with supervised dictionary learning and high-order statistics," in Proc. of European Conf. on Computer Vision, 2014, pp. 660–674.

[13] F. H. Barron and B. E. Barrett, "Decision quality using ranked attribute weights," Journal of Management Science, vol. 42, no. 11, pp. 1515–1523, Nov. 1996.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in Proc. of IEEE Int. Conf. on Computer Vision, 2011, pp. 2556–2563.

[15] F. C. Heilbron, A. Thabet, J. C. Niebles, and B. Ghanem, "Camera motion and surrounding scene appearance as context for action recognition," in Proc. of Asian Conf. on Computer Vision, 2014, pp. 583–597.

[16] O. V. R. Murthy, I. Radwan, and R. Goecke, "Dense body part trajectories for human action recognition," in Proc. of IEEE Int. Conf. on Image Processing, 2014, pp. 1465–1469.

[17] X. Peng, L. Wang, Y. Qiao, and Q. Peng, "A Joint evaluation of dictionary learning and feature encoding for action recognition," in Proc. of Int. Conf. on Pattern Recognition, 2014, pp. 2607–2612.

[18] O. V. R. Murthy and R. Goecke, "Ordered trajectories for large scale human action recognition," in Proc. of IEEE Int. Conf. on Computer Vision, 2013, pp. 412–419.

[19] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," in Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2014, pp. 2577–2584.

Semi-Supervised Learning with Visual Pixel-Level Similaries for Object Detection

Nguyen Dang Binh
Faculty of Information Technology
Hue University of Sciences, Vietnam
ndbinh@hueuni.edu.vn

Abstract – **We introduce a novel approach for detection of objects from aerial images at the level of pixels using semi-supervised learning. Buildings in aerial images are complex 3D objects which are represented by features of different modalities include visual information and 3D height data. Semi-supervised learning is a classification which additional unlabeled data can be used to improve accuracy. This aims to use semi-supervised boosting learning offer an interesting solution to this problem by learning from both labeled and unlabeled data. The major advantage of this approach is the simplicity with which the prior knowledge is incorporated into the semi-supervised learning mechanism. We demonstrate an early result of using semi-supervised boosting learning algorithm to indeed detect building in aerial digital imagery to a satisfactory and useful level of completeness.**

***Keywords**: Semi-supervised learning, pixel-level, aerial image, building detection.*

1. INTRODUCTION

One of the fundamental problems in the area of digital image processing is the automated and detailed understanding of image contents. Methods for building detection from aerial imagery have became very popular, especially in remote sensing, due to a rapidly increasing number of applications, like urban planning and monitoring, change detection, navigation support, cartography or photogrammetric survey. Large areas of the world are now being mapped at human-scale detail to support these applications. A center task is the detection of building structures. Automatic building detection has been a very active research topic in photography and computer vision. Buildings are complex objects with many architectural details and shape variations. Buildings are located in urban scenes that contain various objects from man-made to natural ones. Many of those are in close proximity or disturbing, such as trees, parking lots, vehicle, street lamps, etc. Some objects are covered with shadows or clutter. These difficulties make the problem of a general building detection challenging. A comprehensive survey of building detection from aerial imagery is given in [14]. In recent years an intense research on automatic methods for building modeling at a city-scale has been undertaken [15, 16]. The proposed approaches heavily differ in the use of data sources, extracted feature types, the applied models or the evaluation methods [5, 6, 7, 9, and 10]. Standard learning algorithms such as Naïve Bayes, logistic regression, support vector machines (SVM) assume that the training data is independent and identically distributed.

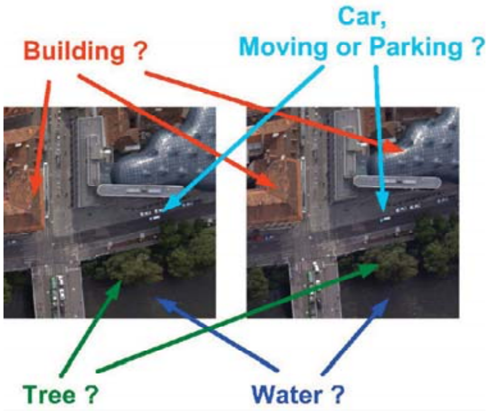


Fig. 1: An illumination of object detection from aerial image in a scene of Graz city, Austria. Semantic interpretation aims at assigning class labels to mapped objects. We focus on a visual pixel-level similaries scene understanding by using semi-supervised learning.

This is inappropriate in many cases, as image pixels possess dependencies, e.g. if a pixel is labeled as building, it is likely that a neighboring pixel tend to be next to other non-building; non-building pixels tend to be next to other non-building pixels. The spatial dependencies should be properly exploited to improve the classification performance rather than classifying each of the image sites independently. In addition, other approaches for general image classification/detection problems are mainly based on locally extracted features and a learned classifier discriminating the object from background. Visual information describing the appearance, such as color and textures are mixed together in a single future vector to represent the object instance. A concatenating of multiple feature types into a single vector may cause the problem of over-fitting due to redundancy and correlation in the input data and one feature type that may inhibit the performance of another. Moreover, supervised approaches obtain high recognition rates if enough labeled training data is available. However, for most practical problems it is important that if there is enough labeled data available or not, whereas hand-labeling is tedious and expensive, then in some cases not even be feasible. This is especially true for applications in computer vision like object recognition and categorization from images and videos, where the human effort is needed to determine the true contents of the media. Semi-supervised methods offer an interesting solution to this problem by learning from combining labeled and unlabeled training samples. Semi-supervised learning algorithms solve the problem of classification under the circumstance that only subsets of the training data are labeled. In contrast to the purely-supervised setting, semi-supervised learning assumes that the probability density of the data is important to discovering the decision boundary. Semi-supervised learning is motivated by situation where copious training data are available, but hand-labeling the data is expensive. Recently Mallapragada et al. [1] have proposed a semi-supervised boosting method which outperforms other approaches on standard machine learning benchmark problem. The supervised approach was extended in order to include unlabeled data [10], based on the idea of considering pairs of samples which are connected via a similarity measure. Similar samples (i.e., a labeled and an unlabeled sample) should share the same label. Our framework develops based on Semi-Boost proposed in [1] and [12] to aerial image building detection problem. In this paper we treat the problem of scene understanding as a semantic labeling process, where each pixel is assigned a specific object class. Due to huge variability the task of visual scene understanding is still a largely unsolved problem. Among occlusions, illumination, viewpoint and scale changes, natural or man-made structures, shape-defined (things) or formless objects (stuff), transparent or highly textured regions complicate the task of finding a meaningful object representation for effective scene understanding. The main contribution of this paper is to combine semi-supervised boosting and visual pixel-level similarity learning using manifold regularization. In particular, we use a limited amount of labeled samples in order to, first, train a discriminated distance function and, second, use this distance function as a metric in order to guide a variant of Semi-Boost [1] through exploiting a huge set of unlabeled data with pixel-level similarity learning. This paper is organized as follows: An introduction to building detection using semi-supervised learning, also a review of related research is given in section 1. Section 2 we introduce Semi-supervised learning with visual pixel-level similarities model for object detection. Section 3 presents Experiments on building detection from aerial image and shows results and evaluation done with implemented detection framework, and finally Section 4 gives a conclusion and outlook to further research.

2. SEMI-SUPERVISED LEARNING MODEL FOR OBJECT DETECTION

2.1 Semi-Supervised Learning

To train an accurate classifier for a specific task, a large amount of labeled training data is needed. This training data often has to be produced by humans and thus is very expensive. Semi-supervised learning [17] tries to benefit from unlabeled data to enhance the performance of a system. Thus, the training data is extended to X = $X^L \cup X^U$ containing labeled X^L and unlabeled examples X^U. It shows, that the performance of the used classifiers can be improved by this simple method using cheap unlabeled sample data. There in principle exist two basic paradigms on how to learn from training data. In the first, supervised learning, training samples are provided together with their corresponding class labels and in the second, unsupervised learning, training samples are given without their labels. Semi-supervised learning is somewhere between supervised and unsupervised methods, thus learns a classifier from both labeled X^L and unlabeled X^U data. Usually, one assumes that there are much more unlabeled samples available than labeled. We assume a typical semi-supervised learning setting in form of a labeled data set.

$$X^L = \left\{(x_1, y_1),\ldots,(x_{|X^L|}, y_{|X^L|})\right\} \subseteq XxY \qquad (1)$$

where $x_i \in X = R^d$ and $y_i \in Y$. In addition, we focus on the binary classification problem, therefore $Y = \{+1,-1\}$ and the samples are split into two sets $X^L = X^+ \cup X^-$ of all samples with a positive class and the set of all samples with negative class, respectively. The goal is to learn a boosted classifier H: X → Y which is trained using both labeled and unlabeled samples. Semi-supervised boosting is a manifold-based approach and uses the following unlabeled loss function:

$$l_u = \sum_{x \in X^U}\left(C \sum_{x' \in X^U} S(x,x')e^{H(x)-H(x')} + \sum_{(x',y') \in X^L} S(x,x')e^{-2y'H(x)} \right) \qquad (2)$$

where C is a trade-off parameter and S(x, x') is a similarity measure between two data samples. Pairs of labeled x and unlabeled x' examples should share the same label as they have a high similarity $S(x,x')$. The unlabeled loss function is a combination of two terms: the first term regularizes only over the unlabeled samples, while the second term uses both labelled and unlabeled data samples. If one uses a symmetric similarity measure, i.e., S(x, x') = S(x', x), the unlabeled loss can be simplified as:

$$l_u = \sum_{x \in X^U}\left(C \sum_{x' \in X^U} S(x,x')\cosh(H(x)-H(x')) + \sum_{(x',y') \in X^L} S(x,x')e^{-2y'H(x)} \right. \qquad (3)$$

As in general manifold-based semi-supervised learning approaches, S(x, x') enforces the graph-based smoothness over the data, i.e., that if x and x' are very similar also the labels should be the same. The standard loss over labelled data. However, we observed that this term can be informative and, therefore, we propose the following loss function

$$\sum_{(x,y) \in X^L} e^{-yH(x)} + \sum_{x \in X^U}\left(\lambda_u \sum_{x' \in X^U} S(x,x')\cosh(H(x)-H(x')) + \lambda_l \sum_{(x',y') \in X^L} S(x,x')e^{-2y'H(x)} \right) \qquad (4)$$

where λ_u and λ_l determine how much the unlabeled loss terms can influence the training. In the following, we will use this formulation of Semi-Boost.

2.2 Image Presentation and Pixel-Level Features

The problem is to fine the most likely configuration of the labels $Y = \{y_i\}$, where $y_i \in \{c_1,..,c_k\}$. For an image labeling, a site is a pixel location and a class may be a building, car, street, tree, etc. For the task of the building segmentation each pixel in the aerial image, represented by a feature vector x_i, is mapped to a bit $y_i \in \{-1,+1\}$, corresponding to either building or non-building. Our work at object-level for the status model at pixel-level matching. Considering an aerial image is a set of buildings $B = \{b_1,\ldots,b_M\}$. The object state vector is $b_k = \{p_1,...,p_N\}$. Each point p_i is defined as $p_i = \{(x, y), (R, G, B), \alpha\}$, where (x, y) are the coordinates with respect to the pixel, (R,G,B) are the color components and $\alpha \in [0,1]$ is likelihood to belong to the object. Let the observed data from an input image be $X = \{x_i, 0 < i, < |X|\}$, where x_i is the data from a site i.

2.3 Semi-Supervised Boosting Learning with Visual Pixel-level Similaries

The learning of a boosting model consists of finding weak learners *h_i(.)* and their weights α_i sequentially. Let $h_i(x) = X \to \{-1, +1\}$ denote the 2-class classification model that is learned at the *i-th* iterration by Boosting algorithm. Let $H(x) = X \to R$ denote the combinated classification model learned the first T interation. It is computed as a linear combination of the first T classification models. This means we need to solve $(\alpha_i, h_i) = \arg\min_{\alpha,h} L$, where L is the loss function. We now have to solve the following optimization problem:

$$\arg \min_{h(x),\alpha} = \sum_{x' \in X^U} \left(\sum_{x' \in X^U} S(x,x')e^{-2y(H(x')+\alpha h(x'))} \right.$$
$$\left. + \lambda_u \sum_{x' \in X^U} S(x,x')e^{((H(x')-H(x))}e^{\alpha(h(x)-h'(x'))} \right) \quad (5)$$

Mallapragada et al. [1] suggest the following approximations in order to simplify the minimization of the objective function. First, $e^{(\alpha(h_i-h_j))}$ is bounded as $e^{(\alpha h_i - h_j)} \leq \frac{1}{2}\left(e^{2\alpha h_i} + e^{-2\alpha h_j}\right)$

This results in an overall upper bound of the objective function

$$F \leq \sum_{x' \in X^U} \left(\sum_{x' \in X^U} S(x,x')e^{-2y(H(x')+\alpha h(x'))} \right.$$
$$\left. + \lambda_u \sum_{x' \in X^U} \frac{S(x,x')}{2} e^{((H(x')-H(x))}\left(e^{2\alpha h(x)} + e^{-2\alpha h'(x')}\right) \right) \quad (6)$$

The objective function can be further written as

$$\overline{F} \leq \sum_{x' \in X^U} e^{(-2\alpha h(x'))} p_i e^{(2\alpha h(x'))} q_i$$

where p_i and q_i are two different terms depending on the label of x' and are formulated as

$$p_x = \lambda_l \sum_{(x',y') \in X^L} I(y'=1)S(x,x')e^{-2H(x')} + \frac{\lambda_u}{2} \sum_{x \in X^U} S(x,x')e^{H(x')-H(x)} \quad (7)$$

and

$$q_x = \lambda_l \sum_{(x',y') \in X^L} I(y'=-1)S(x,x')e^{-2H(x')} + \frac{\lambda_u}{2} \sum_{x \in X^U} S(x,x')e^{H(x)-H(x')} \quad (8)$$

where I(.) is the indicate function. Moreover, p_x and q_x can be considered as the confidence for a sample of being positive and negative, respectively. Using these two terms, we compute the pseudo-labels and weights of unlabeled samples by:

$\widehat{y}_x = sign(p_x - q_x)$ and $w_x = |p_x - q_x|$

Calculating the derivative of the loss function with respect to _ and setting it to zero yields the optimal for the weak classifier as

$$\alpha = \frac{1}{4} \ln \frac{\sum_{x \in X^U} p_i I(h(x)=1) + q_i I(h(x)=-1))}{\sum_{x \in X^U} p_i I(h(x)=-1) + q_i I(h(x)=1))} \quad (9)$$

Therefore, at each step of boosting, we compute the pseudo-labels and weights of unlabeled samples and use them to find the best weak learner and its corresponding weight similar to the AdaBoost algorithm. Note that if no unlabeled data is used, the algorithm reduces to standard boosting. Semi-Boost has the power to exploit both labelled and unlabelled samples if a similarity measure S(x, x') is given. The similarity can be obtained from a distance measure d(x, x') by using a radial basis function $S(x,x') = e^{\left(-\frac{d(x,x')^2}{\sigma^2}\right)}$, where σ^2 is the scale parameter, e.g. $\sigma^2 = 0.001$ and The crucial point is how to measure the distance d(x, x') between points. A more powerful and flexible approach is to learn the distance function from labelled data, which is also known as metric learning. The advantage of discriminative learning of distance functions is that the metric can much better support task-specific classification. We use boosting to learn pair-wise distance functions. The learning problem can be defined as a learning problem on the product space as $H^p : X \times X \to Y = [-1 \quad 1]$.

Algorithm 1: Semi-Boost

Require: Labeled training data $(x, y) \in X^L$ and unlabeled data $x^{\cdot} \in X^U$
Require: Similarity measure S(x, x')
Require: Strong classifier H(x) (initialized randomly)
Require: Weak learners h_i
Require: Weight parameters λ_u, λ_l
Require: max iterations T

```
for i = 1, 2,…,T do
    Computer p_i and q_i for every given sample
    $\widehat{y}_x = sign(p_i - q_i)$
    $w_x = |p_i - q_i|$
    Compute $\alpha_i$ using Equation (9)
    $H(x) \leftarrow H(x) + \alpha_i h_i(x)$
End for
```

To train a maximum margin classifier the training set

$$D^p = \{(x, x', +1) | y = y', x, x' \in D^L\} \cup \{(x, x', -1) | y \neq y', x, x' \in D^L\}$$

is built by taking pairs of images of “same” and “different” class. Using pairs allows us to create a large number of training samples while having only a few labelled starting samples. In particular, if we omit pairs with self-similarities such as (x; x), we can create $\frac{n.(n-1)}{2}$ training pairs out of n positive samples. The symmetry of the distance is not satisfied automatically, therefore it has to be enforced by introducing each pair twice, i.e., both (x, x’) and (x’, x). This also means that the number of training samples in fact becomes $n(n-1)$. The trained and normalized classifier $H^d(x,x')\in\begin{bmatrix}-1 & 1\end{bmatrix}$ is interpreted as a distance $d(x,x')=\frac{1}{2}(H^d(x,x')+1)$ The conversion into a similarity S(x, x’). This learned similarity can now be used as a prior for Semi-Boost as depicted in Figure 2.

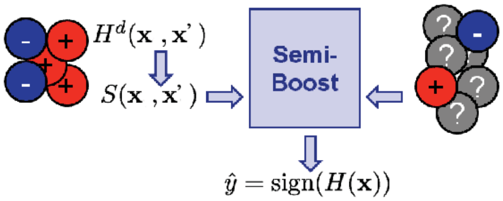

-
+
+
-
+
$H^d(\mathbf{x},\mathbf{x}')$
$S(\mathbf{x},\mathbf{x}')$
Semi-Boost
?
-
?
+
?
?
$\hat{y}=\mathrm{sign}(H(\mathbf{x}))$

Fig. 2: An approach to semi-supervised boosting model.

2.3.1 Labelled samples

We use the exponential loss for samples $x\in X^L$ with correct label y_i as

$$L(x,y)=e^{-2yH(x)} \quad (10)$$

2.3.2. The loss between labelled and unlabeled samples

Given a sample $x_i\in X^L$ labelled with y_i and a second unlabeled sample $x'\in X^U$.

$$L^{LU}(x_i,y_i,x')=S(x_i,x')e^{-2yH(x')y_i} \quad (11)$$

where $S(x_i,x')$ is a similarity measure of the two samples.

2.3.3 The loss of two unlabeled samples

Given two unlabeled samples $x_i,x'\in X^U$. The loss function is the combined loss over the labelled and unlablled samples, respectively; we define a combined objective function.

$$L=\frac{1}{|X^L|}\sum_{X\in X^L}e^{-2yH(x)}+\frac{1}{|X^L||X^U|}\sum_{x_i\in X^L}\sum_{x_j\in X^U}S(x_i,x_j)e^{-2y_iH(x}$$

$$+\frac{1}{|X^U||X^U|}\sum_{x_i\in X^U}\sum_{x_j\in X^U}S(x_i,x_j)e^{\mathrm{H(x_i)}-H(x_j)} \quad (12)$$

2.3.4 Using arbitrary classifiers as similarity-priors

The training a pair-wise classifier which can serve as a distance measure for Semi-Boost. However, sometimes it is the case that we have already given a non-pair-wise classifier that is able to deliver confidence-rated predictions, for instance, an object detector, which can already (partially) solve our problem. From a practical perspective and as can be seen in Figure 3, it would be now beneficial to incorporate such a prior into Semi-Boost in order to exploit unlabeled data. Therefore, we have to make it approximate the pair-wise similarity S(x, x’) using the non-pair-wise prior $H^P(x)$. We will denote such a classifier as prior classifier $H^P(x)$. In the following, we first show how the training can be done. Second, as for evaluation we can use the prior by combining it with the newly trained classifier. Thereby, we benefit from the information which is already encoded in the prior classifier. Roughly speaking, the newly trained classifier can be rather “small”, only correcting the mistakes of $H^P(x)$.

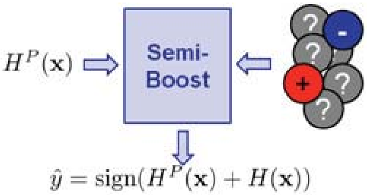


Fig. 3: Semi-Boost combined with a learned similarity measure from given labeled.

We assume that we have access to a prior $H^p: X\times X\rightarrow Y=\begin{bmatrix}-1 & 1\end{bmatrix}$ (e.g., an already trained face detector). The classifier has to provide a confidence measure of its classification. The more confident the decision is the higher the absolute value of the response reaches . We can use boosting for training such a classifier and the responds can be translated into a probability. We can now incorporate this classifier into Semi-Boost using the following approximation

$$|H(x,x')|\approx|H(x)-H(x')| \quad (13)$$

In other words, a discriminative pair-wise function measuring the distance between x and x’ is approximated by the difference of a conventional classification function for two samples. If H(x, x’) is a large margin function and two samples are identical H(x, x’). H(x, x’) will be zero whereas H(x, x’) will be large the more dissimilar x and x’ are. The same holds for taking H(x)-H(x’) which will also be zero if the two samples are identical and large the more dissimilar they are. Hence, approximating the similarity using a non-pair wise classifier corresponds to indirectly measuring the distance to the decision boarder. The principle is visualized in Figure 4.

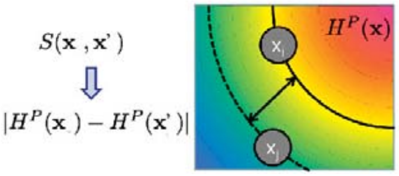

$S(\mathbf{x},\mathbf{x}')$
$|H^P(\mathbf{x})-H^P(\mathbf{x}')|$
$H^P(\mathbf{x})$
x_i
x_j

Fig. 4: The similarity between two samples x and x' is approximated by the difference of the responses from a prior given classifier $H^P(x)$.

2.3.5 Training classifier

Boosting now solves the objective function in a greedy manner by stage-wise selecting the best weak $h_n(x)$ with weight α_n and adding them to the ensemble H. Formally,

$$(\alpha_n, h_n) = \arg \min_{\alpha_n, h_n} (\mathcal{L}) \qquad (14)$$

where $\mathcal{L}$ is loss function in equation (10).

$$\mathrm{h_n(x)} = \arg\min_{\mathrm{h_n}} \left(\frac{1}{|X^L|} \sum_{\substack{x \in X^L \\ h_n(x) \neq y}} \mathrm{w_n(x,y)} - \frac{1}{|X^U|} \sum_{x \in X^U} (p_n(x) - q_n(x)).\alpha_n \mathrm{h_n}(x) \right) \qquad (15)$$

$$\alpha_n = \frac{1}{4} \ln \left(\frac{\frac{1}{|\mathrm{X^U}|} \left(\sum_{\substack{\mathrm{x \in X^U} \\ h_n(x)=1}} p_n(\mathrm{x}) + \sum_{\substack{\mathrm{x \in X^U} \\ h_n(x)=-1}} q_n(\mathrm{x}) \right) + \frac{1}{|\mathrm{X^L}|} \sum_{\substack{\mathrm{x \in X^L} \\ h_n(x)=y}} \mathrm{w_n(x,y)}}{\frac{1}{|\mathrm{X^U}|} \left(\sum_{\substack{\mathrm{x \in X^U} \\ h_n(x)=1}} p_n(\mathrm{x}) + \sum_{\substack{\mathrm{x \in X^U} \\ h_n(x)=-1}} q_n(\mathrm{x}) \right) + \frac{1}{|\mathrm{X^L}|} \sum_{\substack{\mathrm{x \in X^L} \\ h_n(x) \neq y}} \mathrm{w_n(x,y)}} \right) \qquad (16)$$

where:

X^L is the set of pixels labled, inferred: $X^L = \{ X^+ , X^- \}$. X^+ is a set of pixel labled by 1. It mean that pixel belongs to object detection; X^- is a set of pixel labled by -1, It mean that pixel does not belong to object detection; $| X^L|$: total number of pixels of X^+ and X^-. X^U is a set of pixels labled by 0. It means that the pixels are unlabelled. We have $|X^U| = \text{Rows.Cols} - |X^L|$ (Rows.Cols is number of pixels in a training image).

$$w_n(x, y) = e^{-2.y.h_{n-1}(x)} \qquad (17)$$

$w_n(x,y)$ is the weight of a labled sample.

Let $X^+ = \{<x,y> | x \in X^L , y=1\}$ be the set of positive samples and Let $X^- = \{<x,y> | x \in X^L , y= -1\}$ the set of negative samples then the terms

$$p_n(x) = e^{-2h_{n-1}(x)} . \frac{1}{|X^L|} \sum_{x_i \in X^+} S(x, x_i) + \frac{1}{|X^U|} \sum_{x_i \in X^U} \left(S(x, x_i) e^{h_{n-1}(x_i) - h_{n-1}(x)} \right) \qquad (18)$$

and

$$q_n(x) = e^{2h_{n-1}(x)} . \frac{1}{|X^L|} \sum_{x_i \in X^-} S(x, x_i) + \frac{1}{|X^U|} \sum_{x_i \in X^U} \left(S(x, x_i) e^{h_{n-1}(x_i) - h_{n-1}(x)} \right) \qquad (19)$$

can be interpreted as confidences of an unlabeled sample belonging to the positive (Eq. 18) and negative class (Eq. 19), respectively. The classifier is trained in order to minimize the weighted error of the samples. For a labeled sample $x \in X^L$ this is the same as comment boosting with weight $W_n(x)$. The second term considers the distance between the unlabeled sample and the labeled samples. Each unlabeled sample $x \in X^U$ gets the (pseudo)-label $\hat{y}_n(x) = sign(p_n(\mathrm{x}) - q_n(x))$ and should be sampled according to the confidence weight $w_n(x) = |p_n(\mathrm{x}) - q_n(x)|$. Summarizing, the algorithm minimizes an object function which takes distance among semi-labeled samples into account using a given similarity measure between samples. When no unlabeled data is used (i.e. $X^L = \{ X^+ , X^- \}$) Equation (15) and (16) reduce to the well known AdaBoost formulation. After the training, we have a strong classifier similar to standard boosting.

2.3.6 Classifiers Combination

We train a Semi-Boost classifier H(x) using the prior classifier $H^P(x)$ as similarity measure, it makes sense to use this prior knowledge for the final classification process as well (i.e., combine the two classifiers). Similar to the standard boosting we can take a look at the expected value of the loss function [13] and compared to logistic transformation into a probability in form of $P(y=1|x) = \frac{e^{H^C(x)}}{e^{H^C(x)} + e^{-H^C(x)}}$ with $H^C(x) = H^P(x) + H(x)$ we get for the combined classifier $P(y=1|x) = \frac{e^{H^P(x)+H(x)}}{e^{H^P(x)+H(x)} + e^{-H^P(x)-H(x)}}$. If we are only interested in the decision we see that a sample is classified as positive if we set $P(y=1|x) \geq 0.5$ and after some mathematical rewriting we get

$$\hat{y} = sign(\sinh(H^P(x) + H(x))) = sign(H^P(x) + H(x))$$
$$= sign(\cosh(H(x))\sinh(H^P(x)) + \cosh(H^P(x)) + \sinh(H(x)))$$

Algorithm 2: Semi-Supervised Learning with Visual Pixel-Level Similarities

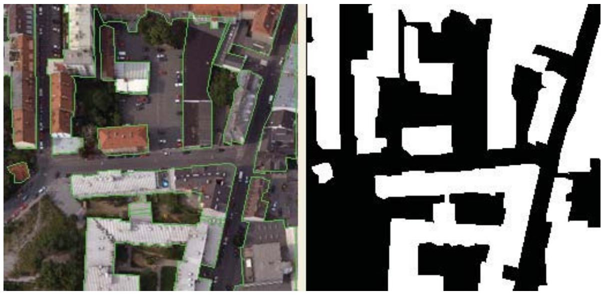
```
Require: Training data   ${(x, y)} \in X$
Required: Prior classifier $H^P(x)$
          (can be initialized by training on $X^L$)
Require: Similarity measure S(x, x')
Require: Strong classifier (initialized randomly)
Require: Weight parameters $\lambda_u, \lambda_l$ (initialized with 1)
Require: max iterations T
Trained prior classifier $H^P(x)$
Using $H^P(x)$ determine $X^+$ and $X^-$. $X^L = X^+ \cup X^-$;
  $X^U$ and $X = X^U \cup X^L$
Initilizated H(x) = $H^P(x)$
for n=1, 2,…,T do
    Using $h_n(x)$ computer S(x, x');
    Computer $p_n(x)$ and $q_n(x)$ for every given sample
     using Equation (18) and (19)
    $\hat{y} = sign(\sinh(H^P(x) + H(x))) = sign(H^P(x) + H(x))$
    $w_x = |p_x - q_x|$
    Computing $\alpha_n$ using Equation (16)
    Computing $h_n(x)$ using Equation (15)
    $H(x) = H^P(x) + H(x)$
End for
```

3. EXPERIMENTS AND RESULTS

The aim of our experiments is to demonstrate the efficiency of semi-supervised learning with visual pixel-level similarities and robustness of our framework for building detection from aerial images. They also demonstrate the successful of our approach. In following, we first give a description of the used data sets, we then present the training process, and report the performance of our system.

3.1 Dataset

In this work we use two different datasets. The first dataset was acquired in the summer of 2005 from the city center of Graz, Austria. It consists of 155 images flown in 5 strips. We present results for aerial imageries of different characteristics. The data set Graz shows a colorful appearance with challenging buildings. The imageries are taken with Microsoft Ultracam in overlapping strips (80% along-track overlap and 60% across-track overlap). This camera procedure 4 color channels in red-green-blue-near infrared, and the images used initially are at a ground resolution of 8 cm. The radiometry is presented with 16 bit per color channel, with a verified range between 12 and 13 bit, where each image has a resolution of 11500 x 7500 pixels with ground sampling distance of approximately 10 cm. The initial training and testing sets are selected on two non-overlapping parts of the aerial images. In our approach we exploit hand-labeled group truth maps for training the classifier. We used to have 48 experimental images which are divided into two subsets, training set and testing set with the dissimilar images. Each subset whose number is 24 images. The size of each image selected is trained and tested at 256 x 256.



Labelled building image Ground-true image for training

Fig. 5 Illustration of an our annotation tool to build a database of annotated aerial images.

3.2 Training process

3.2.1 Accuracy Measures

For a quantities evaluation, we use so-called recall-precision curves (RPC) [11]. We manually establish a reference set of buildings, representing the ground truth with #nP pixels of Buildings. Of the total detected pixels of building, #TP are the true positives and #FP the false positives. The precision rate (PR) shows the accuracy of the prediction of the positive class. The recall rate (RR) shows how many of the known total number of positive samples we are able to identify. The precision rate shows how accurate we are at predicting the positive class. The recall rate tells us how many of the total positives we are able to identify. For detection there is always a compromise between precision and recall. This is evaluated by the F-measureas the harmonic mean. For a visual evaluation of the detector, we plot RR against 1-PR. What is a “correct detection”? We accept detection as “correct” if and only if the pixels of building of the detection correspond to the annotated ground truth buildings.

$$PR = \frac{\#TP}{\#TP + \#FP};\; RR = \frac{\#TP}{\#nP + \#FN};\, and\;\, Fm = \frac{2.RR.PR}{RR + PR}$$

3.2.2 Training Evaluation

The experiments serve to assess and demonstrate the efficiency of the semi-supervised training and robustness of our framework for building detection from aerial imageries.

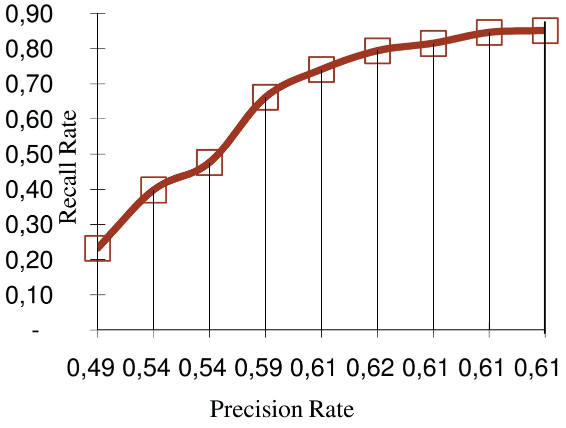


Fig. 6. RPC of the system on dataset

3. 3 Results

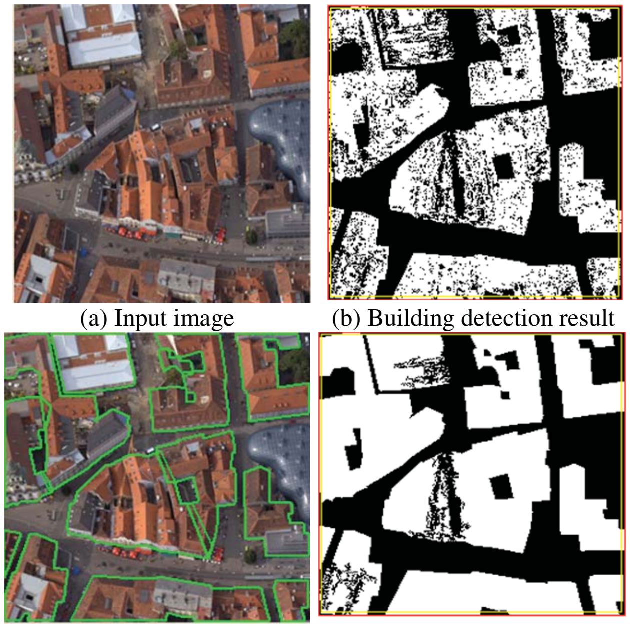
To illustrate the advantages of the described semi-supervised detection approach and all other extensions that have been added within this work, a number experiments have been done. All experiments have been run several times and averaged for confident results.

TABLE I: EXPERIMENTAL RESULTS OF BUILDINGS DETECTION FROM AERIAL IMAGES

	Testing results	
	Image	***Percental result (%)***
1	*01.jpg*	93.9
2	*02.jpg*	94.3
3	*03.jpg*	83.9
4	*04.jpg*	88.3
5	*05.jpg*	90.9
6	*06.jpg*	87.8
7	*07.jpg*	85.2
8	*08.jpg*	93.2
9	*09.jpg*	93.2

TABLE II: COMPARED WITH THE RESULTS OF THE RESEARCH AUTHORS HAVE RECENTLY PUBLISHED WORKS

Graz dataset	**The accuracy of the classification (%)**		
Methods	**Overall**	**Building**	**Non-Build**
SVM	88.2	91.5	85.8
Random Forest (RF)	85.4	77.0	91.5
Stack RF model	88.4	91.5	88.4
Stack Graph Model	91.7	93.4	91.1
Our approach	90.1	91.8	89.1

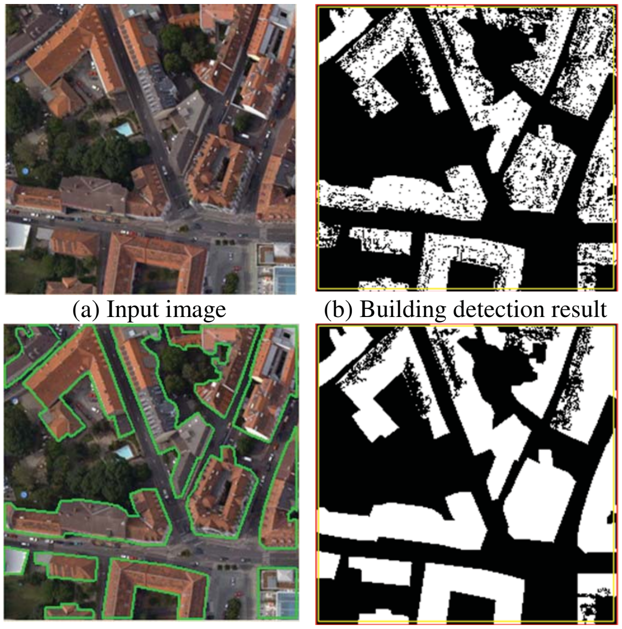


(a) Input image (b) Building detection result

(c) the final result (d) Result after post-processing

Fig. 7: An illumination of the result of building detection from aerial image (*02.jpg)* in a scene of Graz city, Austria.

.



(a) Input image (b) Building detection result

(c) the final result (d) Result after post-processing

Fig. 8: An illumination of object detection from aerial image (*07.jpg)* in a scene of Graz city, Austria.

4. CONCLUSION

The work conducted for this paper and the experiments have clearly shown the advantages of the semi-supervised with visual pixel-level similarities approach. Some main results were obtained as follows: (1) Developping an experimental model Semi-boost approach to improve features extraction in the appropriation of the object. (2) Proposed construction of a model of aerial images include buildings based on the multi-grayscale pixel. (3) Solving the problem effectively detect buildings in aerial images of high resolution. (4) Construction is a simple tool to anotation label the object in the image data, manually. Another benefit is the integration of labeled and unlabeled data. All things considered, this approach is very interesting concerning two major problems of adaptive object detection. First, there is always a lack of labeled data and it is a big advantage to be able to include also unlabeled data into the data set easily. The second big advantage is the robustness to visual pixel-level similarities. We applied our approach to different imageries and demonstrated large-scale capability with low time consumption. The described technique is not only restricted to building detection applications but also works for many other detection applications. This wide applicability offers a promising future for this approach. We expect that also the 3D models will be improved by the knowledge of buildings.

REFERENCES

[1] P. K. Mallapragada, R. Jin, A. K. Jain and Y. Liu, "Semiboost: Boosting for semi-supervised learning", PAMI, 2009.

[2] B. Sirmacek and C. Unsalan, "Building detection from aerial images using invariant color features and shadow information", in Proceedings of International Symposium on Computer and Information Sciences.

[3] C. Jaynes, E. Riseman and A. Hanson, "Recognition and recognition of buildings from multiple aerial images", Journal of Computer Vision and Image Understanding, vol. 90(1), pp. 68-98,2003.

[4] S. Kluckner, T. Mauthner, P. M. Roth and H. Bishoft, "Semantic classification in aerial imagery by integrating appearance and height information", in Proceedings Asian Conference on Computer Vision,2S. 2009.

[5] S. Mueller and D. W. Zaum, "Robust building detection in aerial images", in Proceedings International Society for Photography and Remote Sensing, Workshop CMRT, 2005.

[6] M. Persson, M. Sandvall and T. Duckett, "Automatic building detection from aerial images for mobile robot map",in Proceeding Symposium on Computational Intelligence in Robotics and Automation, 2005.

[7] M. Xie, K. Fu and Y. Wu, "Building recognition and reconstruction from aerial imagery and lidar data, in Proceeding on Radar, 2006.

[8] P. Meiner, F. Leberl, "Discribing building by 3-dimensional detail found in aerial photography", in Symposium " 100 Years ISPRS- Advanving Remote Sensing Science", 2010.

[9] F. Lafarge, X. Descombes, J. Zerubia and M. P. Deseilligny, "Automatic building extraction form dems using an object approach and application to the 3d-city modeling", Journal of Photogrammetry and Remote Sensing, vol. 63(1), pp. 365-381, 2008.

[10] B. Matei, H. Sawhney, S. Samarasekera, J. Kim and R. Kumar, "Building segmentation for densely built urban regions using aerial lidar data", in Proceedings Computer Vision and Pattern Recognition, 2008.

[11] S.Agarwal, A. Awan and D. Roth, "Learning to detect objects in images via a sparse, part-based representation", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26(11), pp. 1475-1490, 2004.

[12] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking". In Proc. ECCV, 2008.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting. Annals of Statistics", vol. 28(2), pp.337–407, 2000.

[14] H. Mayer, "Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings". Computer Vision and Image Understanding, 74(2):138–149, 1999.

[15] G. Vosselman and S. Dijkman, S, "3D Building Model Reconstruction from Point Clouds and Ground Plans". International Archives of Photogrammetry and Remote Sensing, XXXIV(3):37–44, 2001.

[16] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Structural Approach for Building Reconstruction from a Single DSM". IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(1):135–147, 2010.

[17] O. Chapelle, B. Scholkopf, and A. Zien, A., "Semi-Supervised Learning". MIT Press, Cambridge, MA, 2006.

A process for text recognition of generic identification documents over cloud computing

Rodolfo Valiente, Marcelo T. Sadaike, José C. Gutiérrez, Daniel F. Soriano, Graça Bressan, Wilson V. Ruggiero
Laboratory of Computer Architecture and Networks, Escola Politénica da Universidade de São Paulo, São Paulo, SP, Brazil

Abstract - *A process for accurate text recognition of generic identification documents (ID) over cloud computing is presented. It consists of two steps, first, a smartphone camera takes a picture of the ID, which is sent to a cloud server; second, the image is processed by the server and the results are sent back to the smartphone. In the cloud server, image rectification, text detections and word recognition methods are combined to extract all non-manuscript information from the ID. Rectification is based on an improved Hough Transform and the text detection approach consists of a contrast-enhanced Maximally Stable Extremal Region (MSER), using heuristics constraints to remove non-text regions. Experimental results demonstrate several advantages in the proposed process.*

Keywords: *image rectification, text recognition, cloud computing, OCR, ID.*

1 Introduction

Automatic text recognition is one of the hardest problems in computer vision. Although many text detection methods have been studied in the past, the problem remains open [1-7]. An essential prerequisite for text recognition is to locate text on images. This still remains a challenging task because of the wide variety of text appearance, due to variations in font, thickness, color, size, texture, and geometric distortions [8].

Technology development is growing rapidly in the last decade, and having an impact on the use of gadgets in the daily activities. As sophisticated technology is utilized, human needs are also increasingly changing. Manual work will be minimized and changed as much as possible by applying a computer. Identification documents (ID) are one of the main sources for obtaining information about a citizen. Some business sectors require the information contained in ID cards to perform registration processes. In general, registration processes uses forms to be filled in accordance with the data from the ID cards, which will be converted into digital data by manually entering the information [4]. However, a system can extract this information by processing the image of the ID and generating text as a result. This approach is known as optical character recognition (OCR). The information extracted must be accurate, since a recognition error can generate a mistake during the registration, making the system complex and requiring intensive processing [9]. Devices with low computational power can use the cloud for intensive processing.

Related works focus on text recognition,[7, 10, 11], text location and segmentation [12-14] or text rectification [15-17]. These approaches have been considered separately in a number of other recent studies [5, 6, 11, 14]. These techniques have been used for complete ID recognition using a template [4], and for Business Cards OCR in Android Operating Systems [9].

We here propose an implementation of a character recognition process for generic documents. Figure 1 shows a diagram of the system. It consists of two phases: in phase **1**, the ID image is captured with a smart phone and sent to the server, and in phase **2**, a Java server processes the image for text recognition. All image processing algorithms are implemented and debugged in Matlab. Using Matlab compiler SDK, we generate a java package (code.jar) with all the functions previously implemented. Finally, this package is run by the server. The image processing includes four steps: **(a)** preprocessing, **(b)** text-area location and segmentation, **(c)** rectification and **(d)** word recognition.

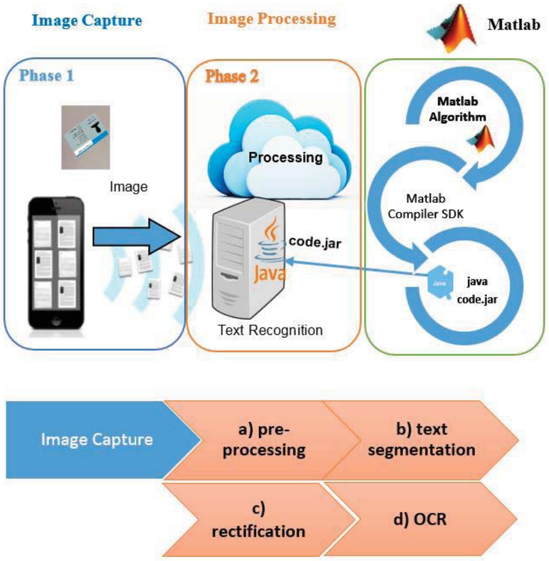


Figure 1. System Architecture. (1) phase 1, image capture. (2) phase 2, image processing. (a, b, c, d, are the steps of the text recognition process)

The rest of the paper is structured as follows: In Section 2, the Text recognition process is presented, in Section 3, the experimental evaluation is provided. The conclusions are in Section 4.

2 Text recognition process

The algorithm consists of four processing steps A, B, C e D, shown in Figure 2. The algorithm optimizes the conditions for text recognition before providing the image to OCR Tesseract.

Tesseract assumes that its input is a binary image with optional polygonal text regions defined. The first step is a connected component analysis, which the outlines of the components are stored and gathered together, converting into Blobs. Blobs are organized into text lines, and the lines and regions are analyzed for fixed area or proportional text size. Text lines are divided into words according to the character spacing and fuzzy spaces[18].

Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in from the text. Each word that is satisfactory is passed to an adaptive classifier as training data. A second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate smallcap text[18].

2.1 Preprocessing

As images are taken in different illumination conditions and at various distances from the smartphone, the images need to be preprocessed to improve the next stages [8, 11] (Figure 2, A).

The pre-processing improves the input image to reduce the noise, hence to enhance the processing speed. The image is converted into gray level image and then into binary image. The following steps are used for preprocessing: automatic contrast adjustment and noise reduction by performing a median filter combined with morphological operators.

2.2 Text segmentation

Following preprocessing, the text segmentation separates the text from the background (Figure 2, B). An adaptive threshold with a contrast-enhanced MSER algorithm is designed to extract the character candidates. Next, simple geometric constraints are applied to remove the non-text regions.

2.2.1 Adaptive Threshold

Concerning the improvement of MSER, the image is subdivided into blocks; for each block all local thresholds are computed. The Adaptive thresholding performs the reduction of a grayscale image to a binary image. The algorithm assumes that in an image there are foreground (black) pixels and background (white) pixels. It then calculates the optimal threshold that separates the two pixel classes such that the variance between the two is minimal. Next, the morphological operations: opening, closing, and dilation, using square structuring elements, are performed. The final result overlays the original image to delete the background area.

2.2.2 Text detection using MSER

The MSER feature detector works well for finding text regions[12, 14] because the consistent color and the high contrast of the text lead to stable intensity profiles.

MSER detects a set of connected regions from an image, where each region is defined by an extremal property of the intensity function within the region to the values on its outer boundary. MSERs are invariant to continuous geometric transformations and affine intensity changes and are detected at several scales. MSER are further considered as the fastest interest point detection method, since algorithms for calculating MSERs in linear time are available [12, 14].

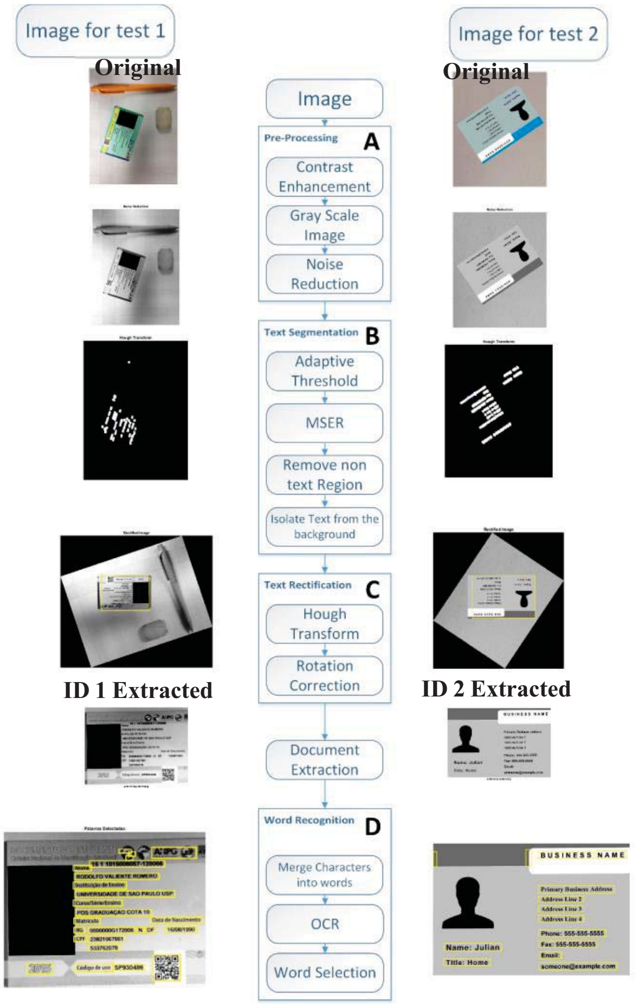


Figure 2. Complete Text Recognition Diagram. (1) Image captured from a real ID. (2) Image from a standard Microsoft Publisher template.

2.2.3 Basic Geometric Properties Filtering

Although the MSER algorithm detects most of the text, it also detects many other stable regions on the image that are not text. There are several geometric properties that are good for discriminating between text and non-text regions, shown in (1)-(7) [8, 12].

$$\text{Aspect ratio} = \frac{\max(\text{width}, \text{height})}{\min(\text{width}, \text{height})} \tag{1}$$

$$\text{Solidity} = \frac{\text{area}}{\text{convex area}} \tag{2}$$

$$\text{Occupy rate} = \frac{\text{area}}{\text{height} * \text{width}} \tag{3}$$

$$\text{Occupy rate convex area} = \frac{\text{convex area}}{\text{height} * \text{width}} \tag{4}$$

$$\text{Stroke width size ratio} = \frac{\text{Stroke width}}{\max(\text{height}, \text{width})} \tag{5}$$

$$\text{Max stroke width size ratio} = \frac{\text{Max stroke width}}{\max(\text{height}, \text{width})} \tag{6}$$

$$\text{Stroke width variance ratio} = \frac{\text{Stroke width variance}}{\text{Stroke width}} \tag{7}$$

The convex area is the area of the convex hull, which is the smallest convex polygon that contains the region. A stroke is a contiguous part of an image that forms a band of a nearly constant width. Characters are made of strokes which have consistent stroke width. The Stroke Width Transform (SWT) is a local image operator that computes per pixel the width of the most likely stroke containing the pixel.

For an ID, the best distinctive properties obtained are: Aspect ratio, Eccentricity, Solidity and Area. Table 1 shows the range of the selected values for each property.

Table 1. Heuristic Properties. t= number of pixels of the image

Properties	Value (v)
Aspect ratio	$0.2 < v < 5$
Eccentricity	$v < 0.995$
Solidity	$v > 0.3$
Area	$1/10^4 *t < v < 1/10^3 *t$

2.2.4 Stroke Width

Stroke width is another common metric used to discriminate between text and non-text; it is defined as the length of a straight line from a text edge pixel to another along its gradient direction. The stroke width remains almost the same in a single character. However, there is a significant change in stroke width in non-text regions as a result of their irregularity. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations [12]; thus, regions with larger variations are removed.

Some text candidates can be erroneously rejected, especially those with a high aspect ratio. Therefore, to bring back the mistakenly removed characters, we take consider that adjacent characters are expected to have similar attributes, such as height and stroke width. Then, by integrating the stroke width generated from the skeletons of those candidates, the remaining false positives are rejected[12]. Finally, all text regions are extracted (extracted text area).

2.3 Text Rectification

During the process of acquiring the picture of the ID, the text image will inevitably have different degrees of tilt and cause certain difficulties to the following OCR process [16]. Thus, image rectification is necessary and is executed by performing an improved Hough transform (Figure 2, C). Hough transform is not applied to the original image, but to the extracted text area. The horizontal and vertical line segments of the text area are extracted, and the line segments serve as an image feature for image rectification [15-17]. Next, the transformation to be applied is computed and the image is rectified (Figure 2, C). From the rectified image the ID is cropped and the text extracted is used for the next step.

2.4 Word Recognition

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines (Figure 2, D), which carry more meaningful information than just the individual characters [11, 12]. The characters are grouped into words based on distance, orientation, and similarities between characters. Then, the recognition process is performed (Figure 2, D) for each word in the bounding boxes.

Using Algorithm 1, the results are improved by applying OCR to the image of the document extracted (Figure 2, ID Extracted) and to the image with the text segmented (without background); both images are shown in figure 3. Therefore, by repeating the OCR twice and taking into account the confidence value (percent of accuracy) given by the OCR function, the most accurate OCR result is selected for each word.

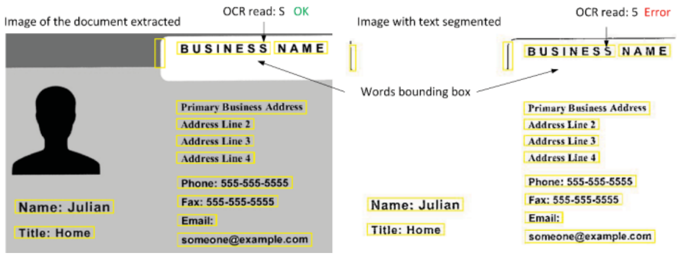


Figure 3. Images used to apply OCR, and improve the results

2.4.1 Merge characters into words

The individual text regions are merged into words or text lines by finding neighboring text regions. This is achieved by expanding the bounding boxes of each individual character computed earlier, until they overlap. The overlapping bounding boxes can be merged to form a chain of horizontal single bounding boxes around individual words or text lines.

Before showing the final detection results, false text detections are suppressed by removing the bounding boxes with just one text region inside. This process removes isolated regions that are unlikely to be a word, taking into account that every text is usually found in groups (words and sentences).

2.4.2 Perform OCR and select best words

After detecting the word regions, the OCR function is used to recognize the text within each bounding box. Word recognition is improved by obtaining the optimal width of the bounding box and by performing OCR on two images (Figure 3). The procedure is outlined in Algorithm 1. The output of the Algorithm 1 is the best result for each word and improves the final word recognition process. For example, in Figure 3, for the bounding box of the word "BUSINESS", in the image on the left, the OCR output is "BUSINESS", with 91% of accuracy, and for the image on the right, the OCR output is "BUSINE*5*S", with 83% of accuracy; then, the algorithm selects the first, which is the better result. The same process is performed for all the bounding boxes.

```
Algorithm 1- Finding the best words for each bounding boxe in each image
Input: A=image set, B= Word bounding boxes (location of each word)
Output: W= selected best words recognized
Initialize W (row, column)          //in the first column we save the
                    //word in the second column the accurate of the OCR
  for i= each image in A do          // iterate over each image and select
                                     //the best recognized words
                                     // i is the image
    for j=each B in image i do       // j is the bounding box
      W_temp(j,1) = OCR (word in the bounding box j )
      W_temp (j,2) = OCR (word in the bounding box j ).confidence
                    // confidence value =percent of accuracy of the OCR
      If (W_temp (j,2) > W (j,2)) //save word with the best OCR accuracy
        W (j,2)= W_temp (j,2)
        W (j,1)= W_temp (j,1)
      end if
    end for
  end for
return W
```

(A) In our case we use two images. We could choose other processed images from the original.

3 Experiments and results

We used 20 different images with 5 different points of view, for a total of 100 images. One of these images is a standard Microsoft Publisher template (Figure 2). The IDs under study are subjected to a variety of adverse conditions, including variable illumination, background, rotation, and text flow. The performance of the all process is evaluated, from the application on the phone to the word recognition by the server. Furthermore, the results of Algorithm 1 are assessed.

3.1 Complete process

Figure 2, shows the results of the full process for two images. In both cases, 100% of the bounding box of the words are detected. In image 2, 100% of accuracy is obtained after performing OCR (Figure 4). In image 1, 21 sentences are detected, from which 19 are read correctly, obtaining a 90% accuracy. The two incorrect words are shown in Figure 5. The error is assumed to be related to the OCR engine lack of training for specific typography.

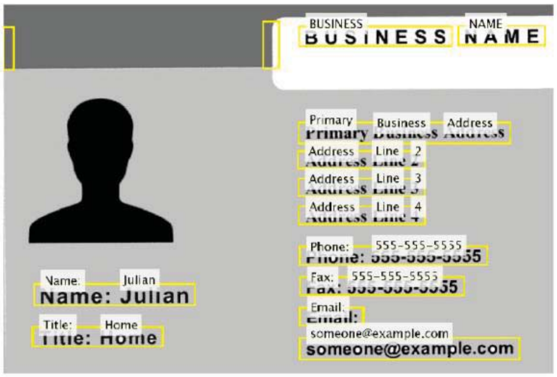


Figure 4. Result from Word Recognition in test image 2 (Figure 2). 12 sentences (100% of the document) with a 100% of accurate OCR were detected.

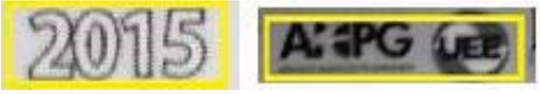


Figure 5. Incorrectly read words in figure 2 (test image 1). For the first word, 2o15 was read and, for the second, A.-*

Images with complex background and typographies different from the trained data have worse results, since the background introduces complex noise into the image and OCR is not trained for these typographies. Therefore, an improved robust process that can get similar results with all documents and environments is necessary.

3.2 Performance of OCR Algorithm

To evaluate the performance of Algorithm 1, 100 images are processed. We compute a sum of: a) the total words in the IDs, b) all the words found after applying OCR, c) words with accuracy better than 80% (confidence of OCR Tesseract), and we calculate d) process accuracy as in equation 8.

$$\text{process accuracy} = \frac{\text{words with OCR} > 80\% \text{ accuracy}}{\textit{total words}} \qquad (8)$$

We calculated that in four different cases (Table 2):
1- OCR over the original image (Figure 2, Original).
2- OCR over the rectified image (Figure 2, ID Extracted).
3- OCR over image with bounding box (Figure 2).
4- OCR using the Algorithm 1.

Table 2. Comparative output result from the OCR function

Case of Study	**a)**Total words	**b)**found words	**c)**OCR > 80% accuracy	**d)**%process accuracy
1-OCR without rectification	1100	63	1	0.09
2-OCR with just rectification	1100	487	344	3.,27
3-OCR with bounding box	1100	1034	871	79.18
4-OCR with our method	1100	1034	963	**87.54**

The method proposed achieves better results than the other available alternatives. For case 1, the image is not rectified; therefore, dreadful results are obtained because OCR tesseract works with words in horizontal direction. For case 2, the image is rectified, but the bounding box is not extracted, thus, not achieving the desirable results. For case 3, the OCR is applied over just one image and it produces some mistakes. For case 4, the OCR is improved by using Algorithm 1 the best results are acquired with an accuracy of 87,54 % as the average for the 100 images.

4 Conclusions

This paper proposed a process for text recognition of generic identification documents over cloud computing. It efficiently combines MSER, a locally adaptive threshold method for text segmentation and a rectification correction using the Hough transform algorithm. Some of the operations described are too intensive to be run in a mobile phone; therefore, cloud computing is used for processing. By using Algorithm 1, experimental results confirm that the process enhances the result of OCR, allowing it to obtain better accuracy of words recognition. As a mobile device and cloud computing are used, the result is highly functional and interesting. Nevertheless, the recognition process has some inaccurate results for images with complex backgrounds and different typographies; the process described will thus be improved in future works.

5 Acknowledgement

The authors would like to thank Scopus Soluções em TI and also Fundação de Apoio à Universidade de São Paulo (FUSP). This work is in the context of the project LARC/MEC-MDIC-MCT-1-11.

6 References

[1] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognition,* vol. 31, pp. 2055-2076, Dec 1998.

[2] B. Epshtein, E. Ofek, Y. Wexler, and Ieee, "Detecting Text in Natural Scenes with Stroke Width Transform," in *2010 Ieee Conference on Computer Vision and Pattern Recognition*, ed Los Alamitos: Ieee Computer Soc, 2010, pp. 2963-2970.

[3] C. Yao, X. Bai, W. Y. Liu, Y. Ma, Z. W. Tu, and Ieee, "Detecting Texts of Arbitrary Orientations in Natural Images," in *2012 Ieee Conference on Computer Vision and Pattern Recognition*, ed New York: Ieee, 2012, pp. 1083-1090.

[4] M. Ryan and N. Hanafiah, "An Examination of Character Recognition on ID card using Template Matching Approach," *International Conference on Computer Science and Computational Intelligence (Iccsci 2015),* vol. 59, pp. 520-529, 2015.

[5] J. M. Lukas Neumann, "Efficient Scene Text Localization and Recognition with Local Character Refinement," presented at the International Conference on Document Analysis and Recognition (ICDAR), 2015.

[6] L. Sun, Q. Huo, W. Jia, and K. Chen, "A robust approach for text detection from natural scene images," *Pattern Recognition,* vol. 48, pp. 2906-2920, Sep 2015.

[7] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading Text in the Wild with Convolutional Neural Networks," *International Journal of Computer Vision,* vol. 116, pp. 1-20, Jan 2016.

[8] A. Gonzalez, L. M. Bergasa, J. J. Yebes, S. Bronte, and Ieee, "Text Location in Complex Images," in *2012 21st International Conference on Pattern Recognition*, ed New York: Ieee, 2012, pp. 617-620.

[9] N. L. Sonia Bhaskar, Scott Green, "Implementing Optical Character Recognition on the Android Operating System for Business Cards," 2011.

[10] I. W. a. H.-C. Chang, "Signboard Optical Character Recognition," 2013.

[11] A. Gonzalez and L. M. Bergasa, "A text reading algorithm for natural images," *Image and Vision Computing,* vol. 31, pp. 255-274, Mar 2013.

[12] Y. Li, H. C. Lu, and Ieee, "Scene Text Detection via Stroke Width," in *2012 21st International Conference on Pattern Recognition*, ed New York: Ieee, 2012, pp. 681-684.

[13] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications,* vol. 41, pp. 8027-8048, Dec 2014.

[14] X. C. Yin, X. W. Yin, K. Z. Huang, and H. W. Hao, "Robust Text Detection in Natural Scene Images," *Ieee Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, pp. 970-983, May 2014.

[15] S. Yonemoto, "A Method for Text Detection and Rectification in Real-World Images," pp. 374-377, 2014.

[16] F. Yin, D. Chen, and R. Wu, "A distortion correction approach on natural scene text image," pp. 1058-1061, 2011.

[17] Y.-m. L. a. Z.-y. H. Yu-peng Gao, "Skewed Text Correction Based on the Improved Hough Tranform," 2011.

[18] R. Smith, "An Overview of the Tesseract OCR Engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, pp. 629-633.

Real-time Image Scanning Framework Using GPGPU – Face Detection Case Study

Mahmoud Fayez[1,2], H. M. Faheem[1,2], Iyad Katib[3] and Naif Radi Aljohani[3]
1: Faculty of Computer and Information Science, Ain Shams University, Egypt
2: Fujitsu
3: Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia

Abstract—**The image scanning problem has been tackled several times from different research perspectives. Speeding up image scanning is a major factor in significantly reducing the time needed to perform face detection operations. In this paper, we introduce a novel framework that easily allows the user to define the scaling factor and the scanning window displacement. The proposed framework deploys the NVIDIA GPGPU in a very efficient way by sorting the working threads to achieve the highest memory throughput. A face detection case study using the Viola-Jones algorithm has been investigated. Results show an increase in frames processed per second, approaching 37 fps for HD images.**

Index Terms—**face detection; GPGPU; Haar classifier; image scanning**

I. Introduction

Image scanning, used for face detection, is considered one of the classical computer vision problems. Over the past few decades, real-time image scanning has been considered as a challenge. Modern image capturing devices continue to raise the bar on this challenge each year. Fortunately, CPUs and co-processors are gaining more raw power to process the new images and videos captured by HD cameras. The challenge now is how to optimally deploy image scanning algorithms and exploit the parallel nature of the problem into suitable paradigms that best fit the modern architectures. Guided by a large number of cores, GPUs are now able to execute massively parallel programs in the form of SIMD (single instruction multiple data) through the redesign of existing algorithms. This has opened a new horizon for high performance computing on many-core architectures. It is currently very promising to design a framework based on the utilization of the GPUs to speed up real-time image scanning.

The challenges to optimally redesign the face detection algorithm for parallel execution have been studied for a decade. Until now there has been no standard measure to determine the efficiency of an implementation and how it compares to another. Many published works claim to offer the fastest parallel face detection implementation. Some trials have used a reduced number of classifier stages and others have increased the window step size. We present here a standard measure of the speed of image scanning implementations. The contribution of this paper includes the design of a framework for an image pyramid scanning technique that is applicable to any scalable classifier while providing a new performance measure based on the number of scanned windows per second to accurately compare results with other implementations. The framework also considers the use of different window sizes and different scale-up factors. In addition, a case study for face detection will be presented.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 describes the proposed framework. Section 4 presents a case study to evaluate the performance. Section 5 clarifies the results. Section 6 contains some concluding remarks and directions for future work.

II. Related Work

Viola and Jones made the first face detector that relies on Haar-like features [1]. The classifier consists of stages as shown in Fig. 1. Each stage consists of individual features, as shown in Fig. 2. Each feature can be computed by subtracting the total sum of the pixels in the white rectangle(s) from the total sum of pixels in the dark rectangle(s), as shown in Fig. 3. There will be overlapping regions among the features. The Viola-Jones algorithm solves the repeated calculations problem by introducing a new image representation, the 'integral image'. The new image encodes the sum area table of each pixel as shown in (1). This allows the sum area table (integral image) to be computed once and the features/classifiers to be applied many times without recalculating the sum of each dark/light region shown in Fig. 3. This means each dark/light rectangle can be calculated form the integral image in constant time and requires only two addition and two subtraction operations [1].

$$IntegralImage(x,y) = \sum_{i\le x}\sum_{j\le y} Image(i,j) \qquad (1)$$

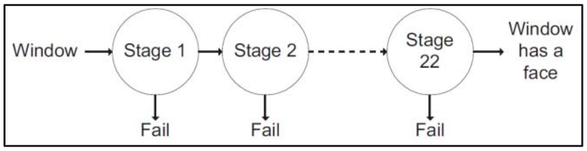


Fig. 1. Viola-Jones classifier stages.

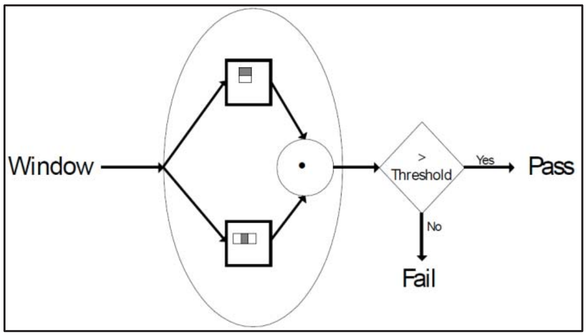


Fig. 2. Viola-Jones stage consists of a set of Haar-like features.

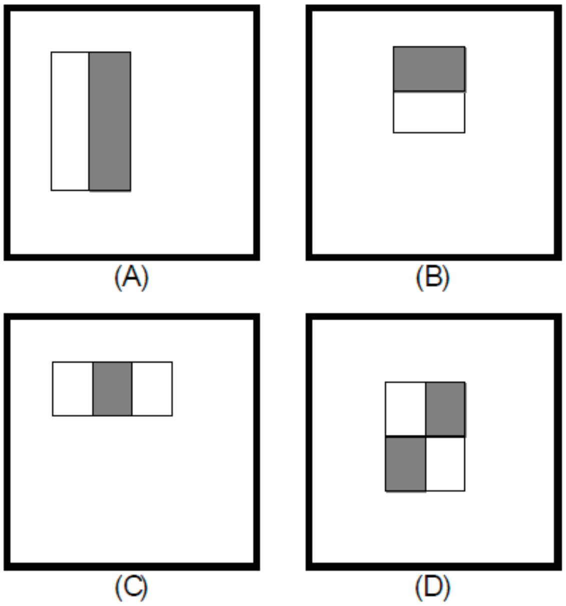


Fig. 3. Haar-like features used by the Viola-Jones algorithm.

Several trials are conducted to speed up the Viola-Jones face detection algorithm. Hefenbrock et al. attempted to speed up the Viola-Jones algorithm using multiple GPUs [2]. They divided the problem into multiple kernel launches where each kernel launch scanned a window of a certain size. This resulted in limiting the number of possible concurrent threads as each group must wait for the previous group to complete. They tried to fix this problem by doubling the number of threads working on each window to achieve good performance on a single GPU of 3.8 fps. A group of four GPUs has achieved 16 fps. However, the GPUs are not well utilized.

Oro et al. have provided a significant improvement by calculating the integral image on the GPU side [3]. They used the GPU built-in texture functionality to generate an image pyramid; however, this required calculating the integral image for each scale. These extra steps have been offloaded to the GPU. The paper also relied on modern GPUs to allow different concurrent kernel launches. The paper stated that the sliding window step size is 1 pixel. However, the calculations in the paper mention that only 256 scanning windows of size 96×96 were evaluated; this indicates that the sliding window step size equals the sliding window size. This leads to a high frame rate on HD images of up to 35 fps. The paper did not mention how many features are evaluated per frame. Also, the large step size would cause some faces to be missed or ignored during the scanning process.

I. Problem Definition

Image scanning can be used to search for a specific feature or a set of features that describe something inside the image. Face detection relies mainly on image scanning techniques. There are two approaches to performing the scanning. The first assumes a fixed image size while the second assumes a fixed scanning window size. In the fixed image size approach, the scanning window size can be changed to cover different areas inside the image. In the fixed window size approach, the scanning window size remains unchanged while the image can be scaled. Fig. 4 shows different image scanning techniques. In this paper, we will use the feature pyramid set. Having an image of size (H×W), where H represents the height and W represents the width of the image, we can extract scanning windows of size (h×w), where h represents the height and w represents the width of the scanning window. If the horizontal and vertical displacements between the extracted windows are identical and equal to s pixels, then the number of windows that can be extracted from an image w_count is described as in (2). Different scanning window sizes are used in the scanning technique, and the number of different scales N is shown in (3). Consequently, the total number of scanning windows w_total that will be processed in each frame is characterized by (4).

This paper solves this computationally intensive problem on an NVIDIA GPGPU using the CUDA parallel computing paradigm. The GPGPU architecture imposes complications that minimize off-chip memory access and achieve the highest utilization of the scalar processors (SP) of the GPGPU as excessive use of the local variables and/or shared memory would reduce the number of active SPs per stream multiprocessor (SM), resulting in idle SPs, which would decrease the overall GPU utilization.

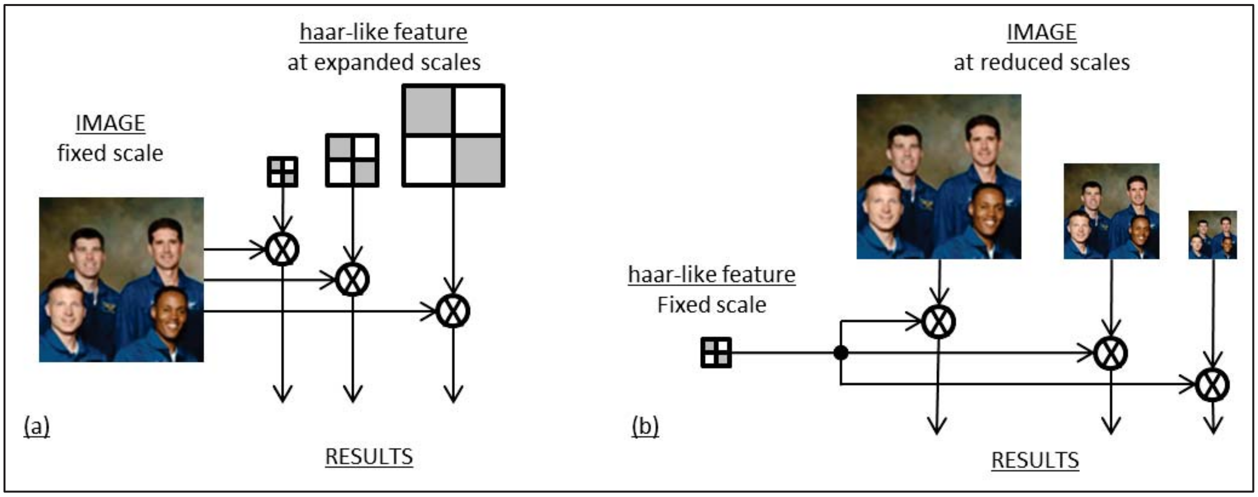


Fig. 4. Feature set pyramid vs image pyramid.

$$w\_count_i = \left(\frac{W - w_i}{S_i} + 1\right)\left(\frac{H - h_i}{S_i} + 1\right) \tag{2}$$

$$N = \frac{\log\left(\frac{w_{\max}}{w_{\min}}\right)}{\log(scale)} \tag{3}$$

$$w\_total = \sum_{i=0}^{N} w\_count_i \tag{4}$$

where:

- i is the index related to the different window sizes from 0 to N, and
- S_i is the step in both the horizontal and vertical directions. The step size is 2 pixels at minimum, or 12.5% of the window size.

II. The Image Scanning Framework

The image scanning framework is shown in Fig. 5. The framework deploys the NVIDIA GPGPUs (*device*) as a hardware component to run the CUDA kernel function while the CPU (*host*) is deployed to run the following software components:

- The feature set pyramid generator (FSPG) generates a pyramid of feature sets.
- The scanning window generator (SWG) generates all the possible scanning windows.
- The result aggregator (RA) merges overlapping detected scanning windows.

A. Operation

Initially, the framework user should define a set of inputs to the framework that includes

- The input image size (W×H),
- The minimum and the maximum scanning window size ($w_{\min}$×$h_{\min}$) and ($w_{\max}$×$h_{\max}$),
- The windows displacement value (in pixels),
- The scale-up factor of the scanning window,
- The initial feature set that is targeting the minimum window size,
- The scale-up function to scale up the features,
- The preprocessing function used to compute the integral image (if necessary), and
- The CUDA Kernel function.

The FSPG shown in Fig. 7 generates a pyramid of feature sets in which the top one has the smallest scanning window size. The FSPG then scales up all the features based on the scale-up factor. The FSPG continues generating the scaled up feature sets until it reaches the maximum scanning window size defined by the user. The generated feature sets are then transferred to the GPU global memory.

The SWG pseudo code, shown in Fig. 6, starts generating different scanning windows and sending them to the GPU's scalar processors (SP), which loads the user-defined CUDA kernel and applies it on the scanning windows. Each SP fetches the corresponding feature set that matches the scanning window size. The most important task of the SWG is to sort the generated list of scanning windows before sending them to the SPs. This guarantees that the coalescing memory access pattern is achieved and SPs in the same block share the same feature set. After finishing the execution of the CUDA kernel against all scanning windows, the results aggregator merges overlapping scanning windows and then sends the results to the user.

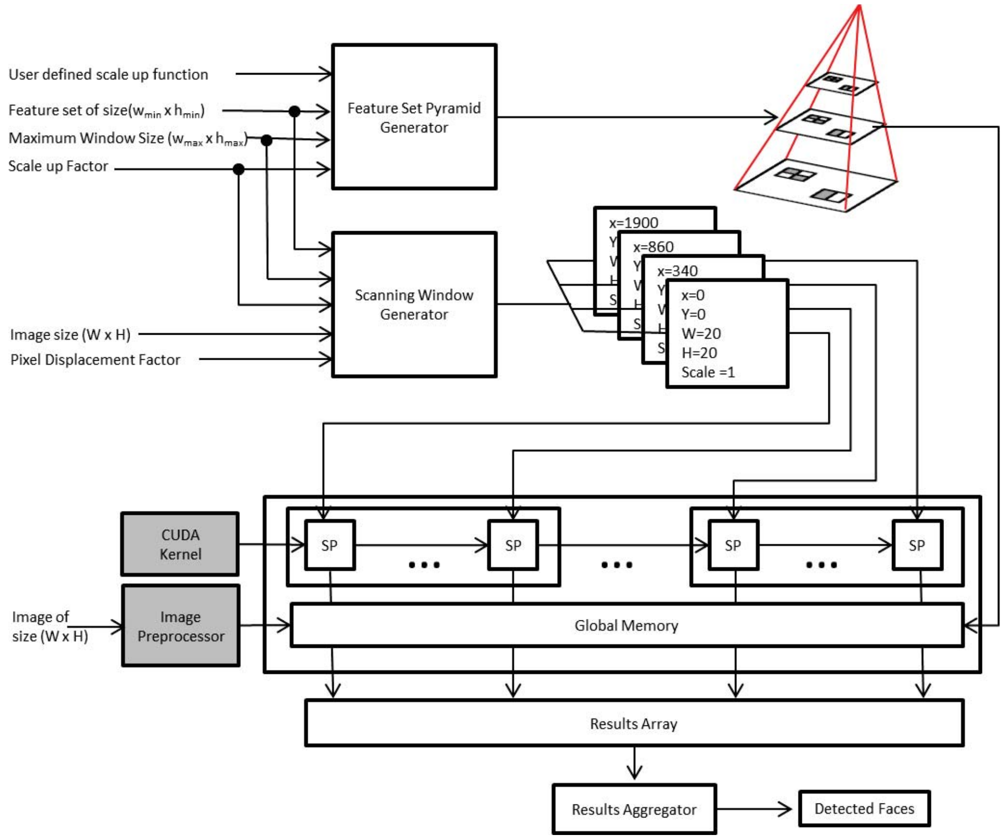

User defined scale up function
Feature set of size(w_{min} x h_{min})
Maximum Window Size (w_{max} x h_{max})
Scale up Factor
Feature Set Pyramid Generator
Scanning Window Generator
Image size (W x H)
Pixel Displacement Factor
x=1900
x=860
x=340
x=0
Y=0
W=20
H=20
Scale =1
CUDA Kernel
Image of size (W x H)
Image Preprocessor
SP
SP
SP
SP
Global Memory
Results Array
Results Aggregator
Detected Faces

Fig. 5. Image scanning framework components.

```
$\textbf{Subroutine ScanningWindowGenerator}$
$Input : W$
$Input : H$
$Input : ScaleUpFactor$
$Input : MaxWindowSize$
$Input : MinWindowSize$
$Output : ListOfWindows$
$\textbf{BEGIN}$
$\textbf{FOR}\ scale = MinWindowSize /20\ \textbf{to}\ MaxWindowSize / 20\ \textbf{STEP}\ ScaleUpFactor$
    $\textbf{FOR}\ x = 0\ \textbf{to}\ W - scale * MinWindowSize$
    $\textbf{FOR}\ y = 0\ \textbf{to}\ H - scale * MinWindowSize$
    $ListOfWindows.Add(x, y, MinWindowSize * scale, scale)$
    $\textbf{END}$
    $\textbf{END}$
$\textbf{END}$
$Sort(ListOfWindows)$
$\textbf{return}\ ListOfWindows$
$\textbf{END}$
```

Fig. 6. The scanning window generator pseudo code.

```
$\textbf{Subroutine FeatureSetPyrmaidGenerator}$
$Input : ScaleUpFactor$
$Input : MaxWindowSize$
$Input : MinWindowSize$
$Input : InitialFeatureSet$
$Output : ListOfFeatureSets$
$\textbf{BEGIN}$
$\textbf{FOR}\ scale = MinWindowSize /20\ \textbf{to}\ MaxWindowSize / 20\ \textbf{STEP}\ ScaleUpFactor$
    $\textbf{FOR EACH}\ feature\ \textbf{IN}\ InitialFeatureSet$
     $feature = ScaleFeature(feature, scale)$
     $ListOfFeatureSets.add(feature)$
    $\textbf{END}$
$\textbf{END}$
$\textbf{return}\ ListOfFeatureSets$
$\textbf{END}$
```

Fig. 7. The feature set pyramid generator pseudo code.

B. *Framework Advantages*

The proposed framework has some advantages, among which are the following. 1) The framework uses a feature set pyramid, which allows flexible displacement of the scanning window, while the image pyramid approach only allows the displacement to be a multiple of the scaling factor, i.e., a half-octave image with window displacement of 1 pixel means the window displacement on the initial image is 2 pixels. Thus, the minimum displacement for a half-octave image is 2 pixels. 2) The SWG sorts the scanning windows based on *x* and *y* positions; if the windows are at the same position they are sorted based on scale-up factor. This guarantees the memory coalescing access pattern of the GPU global memory. 3) All image-independent calculations are made only once, avoiding any delays in the critical execution path of the framework.

III. Face Detection Case Study

The Viola-Jones algorithm is used in this case study. The framework needs a pre-trained feature set, for example, Haar-like features that are found in OpenCV for face detection. The user-defined function responsible for scaling up the Haar-like features scales the position, size, and threshold such that the same feature is still valid for larger scanning window sizes. The framework requires the window processing CUDA kernel function to be defined. The CUDA kernel is not using any shared memory, which proves that the framework does not require a highly optimized CUDA kernel. The challenges that were addressed to ensure the kernel function would achieve the target performance are as follows:

- Reducing the on-chip memory usage to increase the number of concurrent threads per stream multiprocessor (SM);
- Reducing the global memory access to create a coalescing memory pattern, avoiding high memory latency due to the limited number of LD/ST units per SM;
- Maximizing the shared memory among the block threads; this was a challenge that we could not achieve because satisfying the coalescing memory pattern mandates sorting the work items based on spatial locality regardless of the work items' targeted scale. However, if the global memory access can be avoided, then the framework will be flexible enough to accommodate a new sorting function; and
- GPUs don't allow very large one-dimensional (1D) problems, so it is mandatary to start a two-dimensional (2D) grid of threads on the GPU, then use index translation to convert it to 1D. Each thread in the 1D grid fetches the work item parameters generated by the SWG.

IV. Results

The proposed framework was benchmarked using a K20 GPU against the OpenCV Haar-like feature set, which contains 2135 features divided across 22 stages. The image size varies from a VGA image to a 4K image taken from movie trials and downloaded from YouTube. The scalability of the face detector is shown in Fig. 8; it shows that the system can processes 37.2 HD frames per second. This means the framework was able to process 37.2×2135 features per second, which is our new measuring unit for image scanning speed.

It is clear that our proposed implementation outperforms those in [2]–[6], and we have comparable results with Chouchene et al. [7]; however, they did not provide results for images larger than 1024×1024 pixels. All implementations in [2]–[6] are listed in TABLE I. The table shows the image size and the fastest speed reported by the authors. None of the presented implementations mentioned how many features are processed per frame except [4].

Hefenbrock et al. used four GPU cards to achieve 16 frames per second [2]. Oro et al. achieved 35 frames per second using a very large window displacement that would cause skipping of faces from detection [3]. Krpec et al. used only 15 stages of the classifier and achieved 3.5 frames per second [4]. Jiangang et al. achieved very low speed-up using very small image size [5]. Cheng et al. [6] used only 17 stages and 1D image size (720×480) and they used a very small image pyramid, resulting in a high frame rate of 32.5 frames per second.

Chouchene et al. used shared memory to fully optimize the Viola-Jones algorithm and achieved 112 frames per second [7]; however, they did not provide results for larger images. Also, they did not show how many features or stages were used for their classifier. They also did not mention the window displacement and the scaling factor. Rather, they focused on presenting the CPU/GPU profiling comparison of their implementations. Consequently, we were not able to identify the way in which their results were obtained.

TABLE I. CURRENT PUBLISHED FACE DETECTORS AND THEIR CORRESPONDING SPEED

Implementation	Image Size (pixels)	Frames per Second
[2]	640×480	16
[3]	1920×1080	35
[4]	1280×1024	3.5
[5]	340×240	20
[6]	720×480	32.5

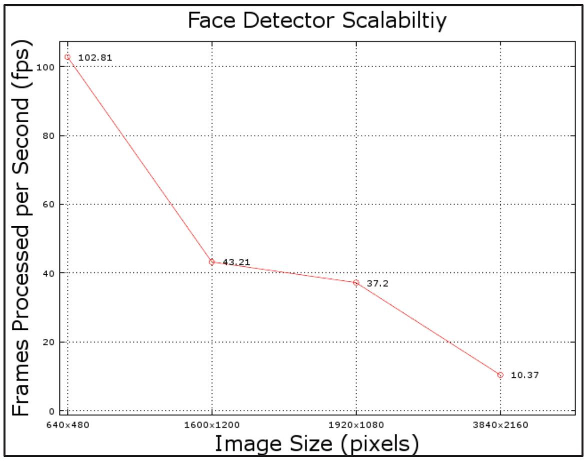


Fig. 8. Face detection frame rate for different image resolutions.

V. CONCLUSION

In this paper, we have presented a framework for performing image scanning. The proposed framework does not necessarily require highly optimized GPU CUDA kernels as long as the work items (CUDA threads) are sorted to achieve maximum memory throughput. This was shown in the face detection case study, in which we achieved a high frame rate compared to previous well-described optimized kernels. The case study shows how many windows and how many features will be processed in HD images to ensure that the results of later implementations can be correctly compared.

We believe that this framework is a step towards defining accurate performance evaluation in the field of image scanning and consequently its dependent applications.

VI. ACKNOWLEDGMENT

This work was supported in part by the High Performance Computing Project in King Abdulaziz University and all experiments were conducted on Aziz supercomputer.

I. REFERENCES

[1] P. Viola and M. Jones, “Robust real-time object detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2001.

[2] D. Hefenbrock, J. Oberg, N. T. N. Thanh, R. Kastner, and S. B. Baden, “Accelerating Viola-Jones Face Detection to FPGA-Level Using GPUs,” *Field-Programmable Cust. Comput. Mach. (FCCM), 2010 18th IEEE Annu. Int. Symp.*, 2010.

[3] D. Oro, C. Fernandez, J. R. Saeta, X. Martorell, and J. Hernando, “Real-time GPU-based face detection in HD video sequences,” *2011 IEEE Int. Conf. Comput. Vis. Work. (ICCV Work.*, pp. 530–537, 2011.

[4] J. Krpec and M. Němec, “Face detection CUDA accelerating,” in *ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions*, 2012, pp. 155–160.

[5] K. Jiangang and D. Yangdong, “GPU Accelerated Face Detection,” in *International Conference on Intelligent Control and Information Processing*, 2010, pp. 584–588.

[6] B. Y. Cheng, J. S. Lee, and J. I. Guo, “An AdaBoost object detection design for heterogeneous computing with OpenCL,” in *Consumer Electronics - Taiwan (ICCE-TW), 2015 IEEE International Conference on*, 2015, pp. 286–287.

[7] M. Chouchene, F. E. Sayadi, H. Bahri, J. Dubois, J. Miteran, and M. Atri, “Optimized parallel implementation of face detection based on GPU component,” *Microprocess. Microsyst.*, vol. 39, no. 6, pp. 393–404, 2015.

License Plate Detection Based on Rectangular Features and Multilevel Thresholding

Ihsan Ullah[1], Hyo Jong Lee [1, 2,*]
[1]Division of Computer Science and Engineering,
[2]Center for Advanced Image and Information Technology
*Corresponding author
Chonbuk National University, Jeonju 561-756, Korea
Ihsanullah736@gmail.com, hlee@chonbuk.ac.kr

Abstract – *Rapid advancement of technology in artificial intelligence and computer science knowledge and then feel the need to search and secure automated systems are because of the appearance of intelligent systems based on image processing and spread this knowledge. One of these intelligent systems is license plate recognition (LPR) system. LPR plays an important role in intelligent transportation system; however, plate region extraction is the key step before the final recognition. In this paper, an effective license plate extraction algorithm is proposed based on geometrical features and multilevel thresholding to identify and segment the license plate from the image. Experimental results show that the technique achieved promising accuracy.*

Keywords: License Plate Detection, Image Processing, LPD, Object Detection.

1 Introduction

License plate recognition (LPR) has been adopted widely into numerous applications such as unattended parking, security control, and stolen vehicle verification. In the LPR system, license plate detection is the most crucial step. It is extremely difficult to detect license plate from a cluttered background efficiently because of the affection of varying illumination, perspective distortion, interference characters, etc. Most of the previous license plate detection algorithms are restricted to certain working conditions, such as fixed backgrounds, known the color, or fixed size of the license plates [1-4]. Therefore, detecting license plate under various complex environments is still a challenging problem.

Recently, LPR has become popular due to its practical importance in image processing applications. Several improvements are proposed in the literature [5, 6] which present efficient and accurate algorithms to detect license plate and identify the numbers and character on the license plate. The license plate localization is a technique to detect and isolate the plate in the image. Several methods have been developed for this purpose. This technology has been used in various intelligent applications, such as the access-control systems, automatic toll collection, intelligent parking systems [7, 8] and traffic analysis, vehicle tracking system, and identification of stolen vehicle can provide valuable information to police for searching the suspected vehicles.

License plate numbers uniquely identify a particular vehicle which varies from region to region because every country has their own license plate layout which differs in their sizes and colors. So there is a necessity for them to develop the LPR system suitable for the vehicle License Plate format. In general, the LPR system has the following parts: the acquisition of the image, the image preprocessing, detection of the license plate, segmentation and the character recognition [9]. The basic block diagram of the system is shown in Figure 1.

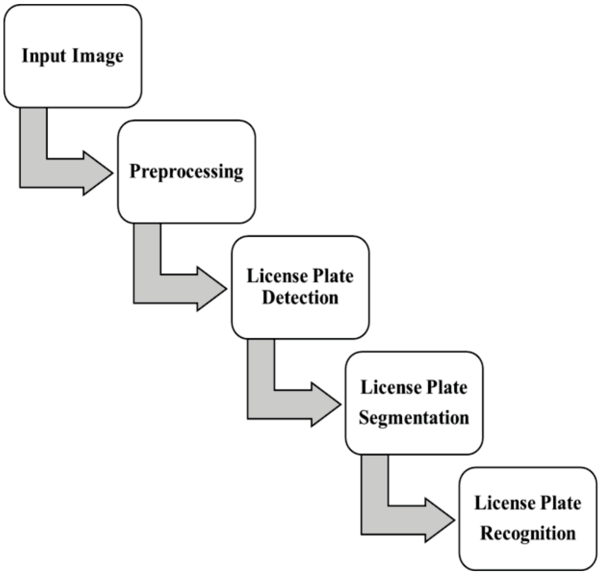


Fig. 1. The basic block diagram of the License Plate Recognition (LPR) System.

The detection steps have been focused in this work, in other words, the determination of the zone where the license plates are located. The proposed algorithm is based on the extraction of plate region. The captured image is processed through the system to obtain the output. There are several detection methods such as [5-10] method and so on. However, in this paper, we focused on geometrical features and thresholding methods. In the literature, many techniques for this step have been reported. A segmentation method based on thresholds is proposed in [14]. In [11, 18] the finding of the plates is based on the analysis of connected components of four different binarization of the image. [5] And [12] discusses the edge detection of license plate using Sobel and canny edge detector respectively, Edge detection by means of gradient and morphological techniques are presented is [13]. The line detection using the Hough transformation is proposed in [14]. [15] proposed an algorithm in which vertical edges and edge density features are utilized to find candidate regions, the candidates are filtered out based on geometrical and textural properties. Learning techniques and Neural Networks have also been studied in this problem. In [16] the approach mainly based on Artificial Neural network while the steps proposed was (1) Plate Localization: - Canny Edge Detector used for the image localization purpose. (2) Character segmentation: - Histogram approach was taken into account for contrast extension while median filtering for noise reduction (3). Feature Extraction: Artificial Neural Network (ANN) was proposed in this process. Two separate ANN used one for character and the other for character extraction because confusion was high when combined approach was applied to both character and numbers so to increase the success rate separate ANN was implemented. (4) Character Recognition: - a Multi-layered perceptron (MLP) model of the ANN was used for the character recognition purpose. The research [17, 18] also describe the LP detection using neural networks. They proposed the methods for license plate detection problem which describe a strategy of multiple classifications based on an MLP.

Every method gives the best results under some certain conditions, but every technique has its own limitations. The variations of the plate types or environments cause challenges in the detection and recognition of license plates. They are summarized as follows.

1) Plate variations:

a) Location: plates exist in different locations of an image;

b) Quantity: an image may contain no or many plates;

c) Size: plates may have different sizes due to the camera distance and the zoom factor;

d) Color: plates may have various characters and background colors due to different plate types or capturing devices.

2) Environment variations:

a) Illumination: input images may have different types of illumination, mainly due to environmental lighting and vehicle headlights

b) Background: the image background may contain patterns similar to plates, such as numbers stamped on a vehicle, bumper with vertical patterns, and textured floors.

The rest of the paper is organized as follows: Section 2 presents the proposed method for license plate detection, Section 3 demonstrates the experimental results and conclusions are drawn in Section 4.

2 Proposed Method

In this paper, we present a method for license plate as shown in Figure 2. The design is considered for the specific characteristics of Korean license plates. The vehicle images were obtained with different backgrounds, illumination, license plate angles, distance from the camera to a vehicle, light conditions and different size and type of LPs.

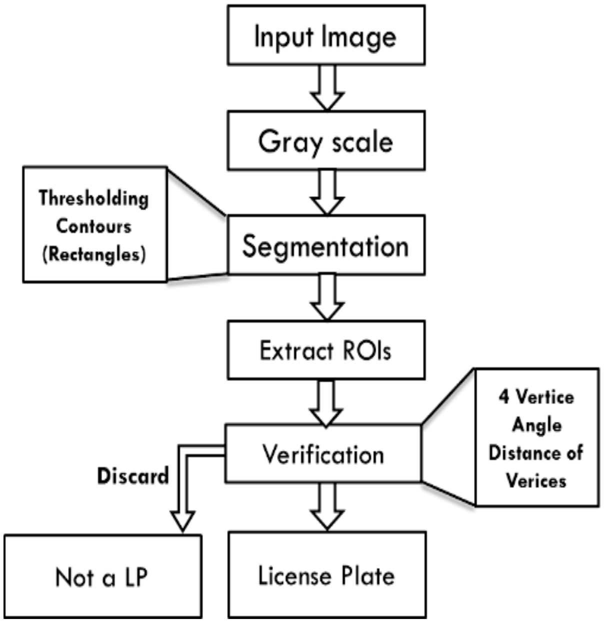


Fig. 2. Flowchart of the proposed method

2.1 Preprocessing

This section deals with the preprocessing procedure, Images taken from the camera were processed by the preprocessing module. The purpose of this module is to enrich the edge features. This will improve the success rates of the license plate detection module. The algorithms sequentially used in this module are graying and noise removal. After having obtained a grey-scale image, we use the image pyramid to reduce the image noise. The resulted images are used as inputs for the license plate detection module.

2.2 License-Plates Detection Algorithm

In order to detect regions of the plate - candidate image, before applying contour algorithm we apply multi-threshold levels to the image. As adaptive thresholding typically takes a grayscale or color image as input and, in the simplest implementation, outputs a binary image representing the segmentation. For each pixel in the image, a threshold has to be calculated. If the pixel value is below the threshold it is set to the background value, otherwise, it assumes the foreground value. We use adaptive threshold instead of zero thresholds to catch possible plate region with gradient shading as shown in Figure 3 while for the rest of levels the system uses binary thresholding as shown in Figure 4.

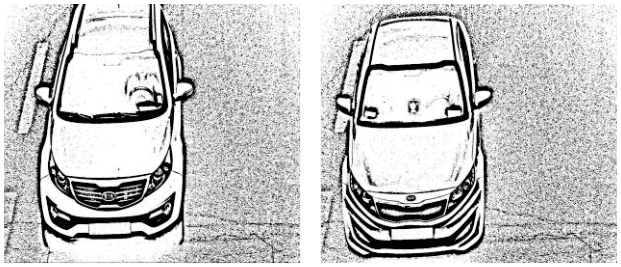


Fig. 3. Adaptive thresholding

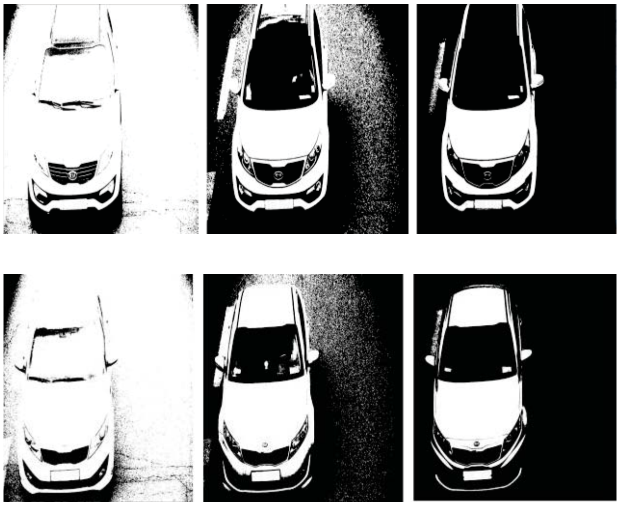


Fig. 4. Several levels of binary thresholding

Contour algorithm is applied to each threshold image which detects the polygons which have 4 vertices. A number of candidate evaluation algorithms are applied to contour images obtained from the contour algorithm to separate regions of interest which may contain license plate, however some false regions were also detected as plate-candidates. To reject such incorrect candidates, we implemented a module for evaluating whether a candidate is a plate or not before verifying the ROIs the algorithm de-skew the ROIs which can help to get good results in license plate recognition.

2.3 Verification of License Plate

As there are many ROIs received as a plate candidate, therefore, the system evaluates plate-candidates' algorithm based on three main steps, which are taken sequentially. The three steps are (1) evaluating the angle and set some parameter of angle, (2) ratio between height and width of candidates, and (3) area of plate candidate using Euclidian distance. After verification the license plate is segmented from the image and rectangle is drawn on the original images as shown in Figure 5.

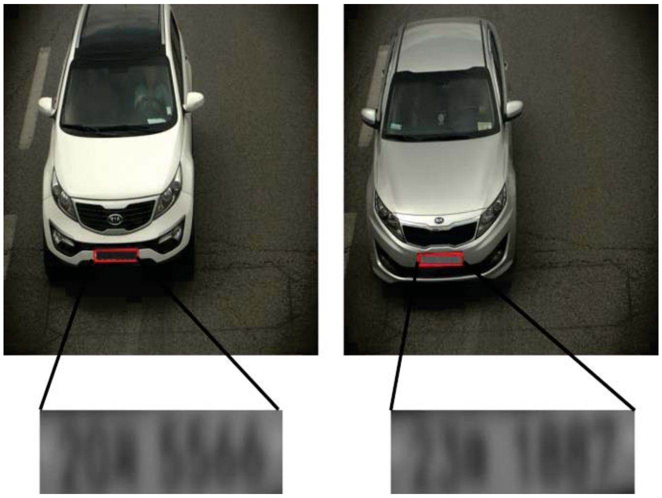


Fig. 5. License Plate Detection

3 Results

The system used in this work is built with Intel Core i5 3.6GHz CPU, 8GB RAM. In the experiments, we use 3000 dynamic images which contain images of different vehicles. These images were captured in different environmental conditions and with different angles. The results show 75% average accuracy while detecting the vehicle license plate (VLP). In the system, 6 different datasets of total size 500 were employed, which is shown in the bar graph with the accuracy of each dataset. In dataset 1 about 76% images were detected correctly, while in dataset 2 the accuracy is increased by 2%, in dataset 3 and 6 the accuracy is 74 %, the accuracy is increased by 1% compared to dataset 3 and 6, while the accuracy has drastically decreased to 73% due to some bad images or miss detection as shown in Figure 6. The overall accuracy is 75%. The correctly detected license plates are shown in Figure 7.

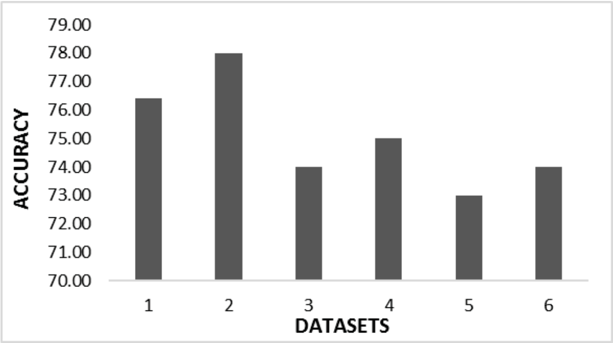


Fig. 6. License plate detection accuracy

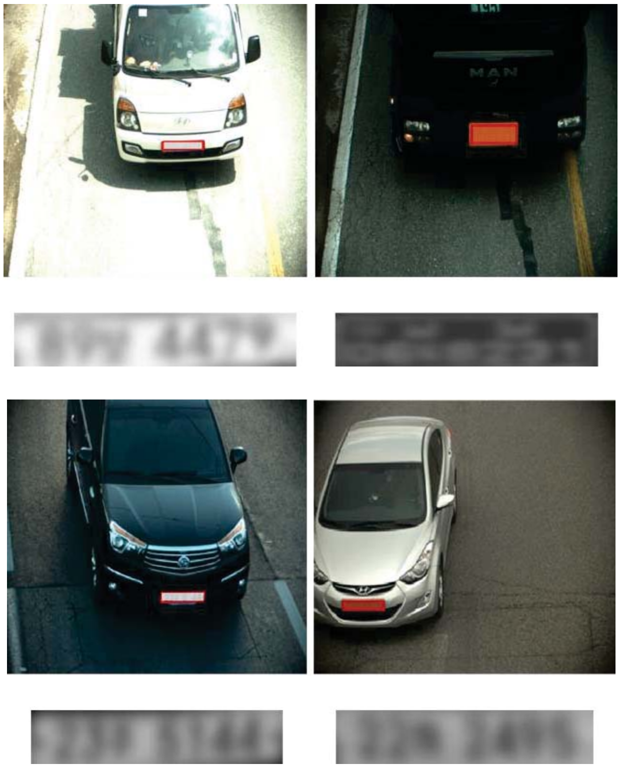


Fig. 7. License plate detection

The rest of the images were either missed or wrongly detected as shown in Figure 8. In image A, there are more than one ROIs because sometimes more than one geometrical object fulfills the rules which are set in the algorithm, in Image B the detection is failing due to the unclear and dusty plate. In Image C in Figure 8 the license plate is not distinguished because of the damage LP region. It is also noted that light intensity and angle variation can also cause miss detection.

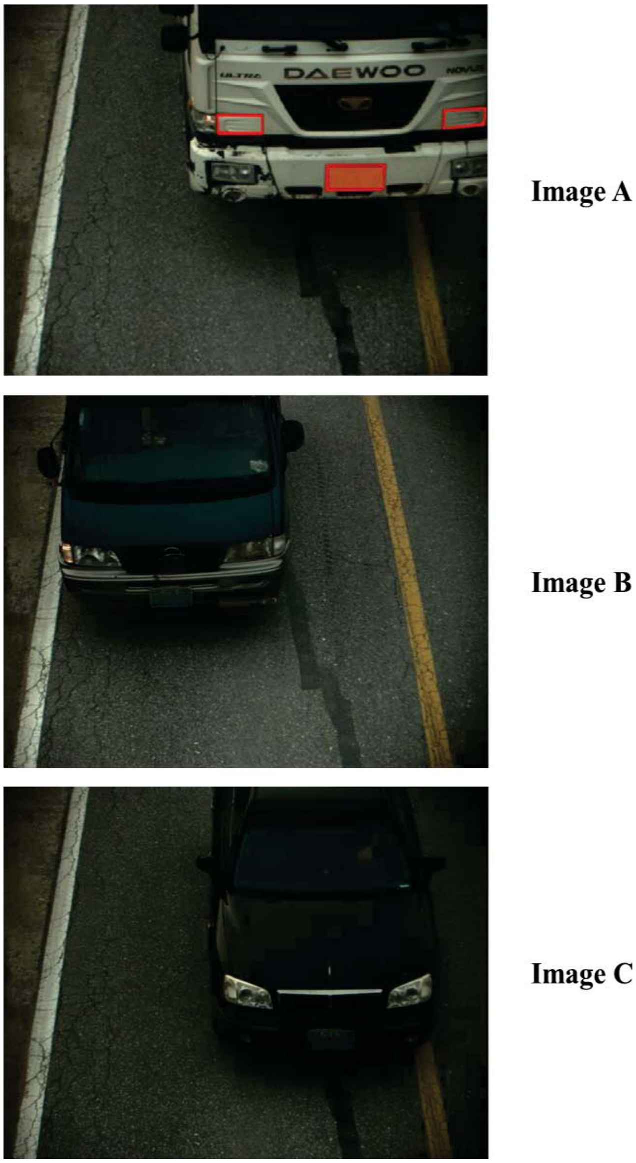


Image A

Image B

Image C

Fig. 8. Wrong and Missed Detection

4 Conclusions

This algorithm detects the license plate using the geometrical feature and multilevel thresholding to identify and segment the license plate from the image. This algorithm performs well on various types of LP images. However, it still has some challenges for example when dealing with bad quality and damaged plates. We are working on a number of algorithms in the preprocessing module. The purpose is to detect regions that are likely plate regions first and thus to reduce the computation cost of the VLP detection algorithm. In addition, we intend to combine a number of texture· based approaches, and machine learning methods to evaluate plate-candidates'. We believe these will improve the accuracy of the algorithm furthermore.

Acknowledgements

This work was supported by the Brain Korea 21 PLUS Project, National Research Foundation of Korea. This work was also supported by the Business for Academic-Industrial Cooperative establishments that were funded by the Korea Small and Medium Business Administration in 2015 (Grants No. C0221114). This research was also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R0992-16-1023) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

[1] Hongliang, Bai, and Liu Changping. "A hybrid license plate extraction method based on edge statistics and morphology." Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 2. IEEE, 2004.

[2] S. K. Kim, D. W. Kim and H. J. Kim, A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation, in Proc. Of International Conference on Image Processing, p.661-664 vol.2, 1996.

[3] S. Kim, D. Kim, Y. Ryu and G. Kim, A Robust License-plate Extraction Method under Complex Image Conditions, in Proc. of International Conference on Pattern Recognition, p.216-219 vol.3, 2002.

[4] W. Jia, H. Zhang, X. He and M. Piccardi, Mean Shift for Accurate License Plate Localization, in Proc. of International Conference on Intelligent Transportation Systems, p.566-571, 2005.

[5] Zheng, Danian, Yannan Zhao, and Jiaxin Wang. "An efficient method of license plate location." Pattern Recognition Letters 26.15 (2005): 2431-2438..

[6] Faradji, Farhad, Amir Hossein Rezaie, and Majid Ziaratban. "A morphological-based license plate location." Image Processing, 2007. ICIP 2007. IEEE International Conference on. Vol. 1. IEEE, 2007.

[7] Yusnita, R., Fariza Norbaya, and Norazwinawati Basharuddin. "Intelligent Parking Space Detection System Based on Image Processing." International Journal of Innovation, Management, and Technology 3.3 (2012): 232.

[8] Al-Kharusi, Hilal, and Ibrahim Al-Bahadly. "Intelligent Parking Management System Based on Image Processing." World Journal of Engineering and Technology 2014 (2014).

[9] Salahshoor, Mohammad, Ali Broumandnia, and Maryam Rastgarpour. "Application of intelligent systems for Iranian license plate recognition." Intelligent Systems (ICIS), 2014 Iranian Conference on. IEEE, 2014.

[10] Al-Hmouz, Rami, and Subhash Challa. "License plate localization based on a probabilistic model." Machine Vision and Applications 21.3 (2010): 319-330.

[11] Llorens, David, et al. "Car license plates extraction and recognition based on connected components analysis and HMM decoding." Pattern Recognition and Image Analysis. Springer Berlin Heidelberg, 2005. 571-578.

[12] Mousa, Allam. "Canny edge-detection based vehicle plate recognition." International Journal of Signal Processing, Image Processing and Pattern Recognition 5.3 (2012): 1-8.

[13] Zhang, Cheng, et al. "A rapid locating method of vehicle license plate based on characteristics of characters' connection and projection." 2007 2nd IEEE Conference on Industrial Electronics and Applications. 2007.

[14] Duan, Tran Duc, Duong Anh Duc, and Tran Le Hong Du. "Combining Hough transform and contour algorithm for detecting vehicles' license-plates." Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on. IEEE, 2004.

[15] Tarabek, Peter. "Fast license plate detection based on edge density and integral edge image." Applied Machine Intelligence and Informatics (SAMI), 2012 IEEE 10th International Symposium on. IEEE, 2012.

[16] Kocer, H. Erdinc, and K. Kursat Cevik. "Artificial neural networks based vehicle license plate recognition." Procedia Computer Science 3 (2011): 1033-1037.

[17] Carrera, Luis, et al. "License plate detection using neural networks." Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. Springer Berlin Heidelberg, 2009. 1248-1255.

[18] Anoual, Hinde, et al. "Vehicle license plate detection in images." Multimedia Computing and Systems (ICMCS), 2011 International Conference on. IEEE, 2011.

Adaptive Frame-Rate Optimization for Energy-Efficient Object Tracking

Yusuke Inoue[1], **Takatsugu Ono**[2], **and Koji Inoue**[2]
[1]Graduate School and Faculty of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan
[2]Faculty of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan

Abstract— *On-line object tracking (OLOT) has been a core technology in computer vision, and its importance has been increasing rapidly. Because this technology is utilized for battery-operated products, energy consumption must be minimized. This paper describes a method of adaptive frame-rate optimization to satisfy that requirement. An energy trade-off occurs between image capturing and object tracking. Therefore, the method optimizes the frame-rate based on always changed object speed for minimizing the total energy while taking into account the trade-off. Simulation results show a maximum energy reduction of 50.0%, and an average reduction of 35.9% without serious tracking accuracy degradation.*

Keywords: Object tracking, Low energy, Frame-rate optimization

1. Introduction

The importance of on-line object tracking (OLOT) to pursue target objects in captured image frames is rapidly increasing due to the spreading use of emerging real-time applications such as driving assist systems and augmented reality. Because OLOT applications tend to be applied in battery-operated systems, e.g., electric automobiles and smartphones, energy consumption must be minimized without degrading the tracking accuracy.

OLOT systems mainly have two major energy consuming processes: image capturing and object tracking. Employing low power devices such as image sensors [1], [5], [7], [8] and processing units [11], [12] is a well-known approach to improve energy efficiency. However, such local optimization has a potential limitation; it cannot take into account an energy trade-off existing in the two processes. Although applying a high frame-rate configuration dissipates a large amount of energy in image inputs, for instance, it reduces the number of required pattern match operations due to the shorter moving distance of the target object in consecutive frames, resulting in energy reduction in the tracking process.

This paper proposes a novel system-wide energy reduction method for OLOT systems. Unlike conventional implementations, our approach attempts to tune the frame-rate dynamically by considering the energy trade-off based on the behavior of the tracked object. The key challenge is to determine the optimum frame-rate that minimizes the total energy without degrading tracking accuracy. Researchers have proposed a fast tracking algorithm with a reduction in computational cost [3], [9]. In contrast, our approach takes into account not only the computational cost but also the cost required to obtain frames. A few studies have so far discussed frame-rate optimization for reducing energy consumption. LiKamWa et al. reported the development of an image sensing device that supports static frame-rate configuration [7]. Although the static optimization works well for fixed systems such as surveillance cameras installed in buildings, it cannot exploit the dynamic behavior of tracking objects, resulting in energy inefficiency in mobile OLOT applications. Han et al. proposed a dynamic technique that lowers the frame-rate in a smart-phone's display [4] based on scroll actions. Their scope covers only the output devices, so it is orthogonal to our adaptive optimization. The contributions of this paper are as follows.

- We theologically analyze the impact of frame-rate on total energy in OLOT systems. Our reveal there is an optimal frame-rate that minimizes the total energy consumption when an object speed is given.
- The concept of dynamic frame-rate optimization for energy efficient OLOT systems is proposed. We also support sensitivity management of frame-rate control, and two implementations are introduced: energy-aware and accuracy-aware frame-rate optimization methods.
- Our experimental results show a maximum energy reduction of 50.0%, and an average reduction of 35.9% across all benchmarks without causing degradation in the serious tracking accuracy compared with a conventional fixed frame-rate method.

The rest of this paper is organized as follows. Section 2 explains the object tracking algorithm and introduces an energy model for an OLOT system. Section 3 analyzes the energy characteristics of the OLOT system and proposes our dynamic frame-rate optimization scheme. Section 4 reports the evaluation results, and Section 5 concludes this paper.

2. Object Tracking System

2.1 Object Tracking Algorithm

We assume that a template-based object tracking algorithm with an adaptive search-area selection is utilized.

Image frames are captured as a stream, and two tasks are executed on each frame after a target object is detected.

The first task is to define a *search-area*. After the target object is detected at $frame_i$, which is a frame captured at time i, the OLOT system attempts to find the position of the object in $frame_{i+1}$. Although searching the entire frame potentially can achieve higher tracking accuracy, it is quite energy inefficient because a number of pattern match operations are required. An approach to avoid this negative effect is to define a search-area that is much smaller than the full frame size. If the object moves slowly, the adaptive method attempts to make the search-area smaller so as to reduce the required number of pattern match operations. In contrast, it expands the search-area when the moving speed of the object is high. We assume that the shape of the search-area is square and that its size S in pixels is defined by Equation (1).

$$S = (2r + 1)^2, \tag{1}$$

where r is the predicted moving distance of the object in pixels, and it can be defined with the object speed v (pixels per second) and frame-rate FR (frames per second) as follows.

$$r = \frac{v}{FR}. \tag{2}$$

From Equation (1) and (2), we can see that the search-area depends on both the object speed and the frame-rate.

The second task is to find the object in the search-area by using pre-defined templates. If we assume that the template has N width and M height shape, a N×M pixel block in the defined search-area is selected. Then the result of the normalized cross-correlation (NCC) presented by Equation (3) is calculated as a similarity score.

$$R_{NCC}(x, y) = \frac{\sum_j \sum_i SA(x+i, y+j) \cdot T(i,j)}{\sqrt{\sum_j \sum_i SA(x+i, y+j)^2 \cdot \sum_j \sum_i T(i,j)^2}}. \tag{3}$$

Here, $SA(x + i, y + j)$ and $T(i, j)$ are the pixel-values for each position in the search-area and the template, respectively. The NCCs for all candidates in the search-area are calculated. Finally, the pixel block that has the highest NCC score is selected as the result of the tracking process. Each NCC calculation requires 1) three *sum-of-products* with $N \times M$ *multiply-and-add* operations, 2) one *multiply* operation in the denominator, 3) one *square-root* operation, and 4) one *div* operation. Thus, we can consider the required number of operations for obtaining an NCC value, $n_{matching}$, as follows.

$$n_{matching} = 6 \cdot N \cdot M + 3. \tag{4}$$

2.2 Energy Modeling

In this section, we present an energy model for an OLOT system. The energy consumed for object tracking consists of two sources: the energy for frame input E_f and that for object tracking E_t. Thus, the total energy consumed per second, E, can be expressed as follows.

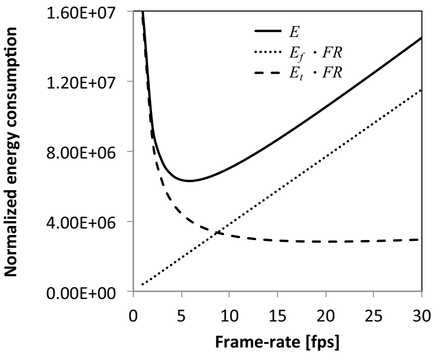


Fig. 1: Relationship between energy consumption and frame-rate.

$$E = (E_f + E_t) \cdot FR. \tag{5}$$

E_f can be defined with the energy for obtaining a pixel from frame memory E_{mem}, frame width W, and frame height H.

$$E_f = E_{mem} \cdot W \cdot H. \tag{6}$$

Generally, E_t is proportional to the number of computations required for tracking processes. Because the computing complexity of arithmetic operations, *add, multiply, square-root, div*, is implementation dependent, we assume for simplicity that they consume the same amount of energy E_{op}[1]. Therefore, E_t can be defined as follows.

$$E_t = E_{op} \cdot n_{matching} \cdot S. \tag{7}$$

In summary, total the energy consumed in a second can be presented by Equation (8).

$$E = (\alpha \cdot W \cdot H + (6 \cdot N \cdot M) + 3)(\frac{2 \cdot v}{FR} + 1)^2) \cdot E_{op} \cdot FR, \tag{8}$$

where α is the energy ratio, E_{mem} divided by E_{op}. The exact values of E_{op} and E_{mem} depend on the specification of devices, e.g., the performance capability and memory capacity. The object speed *v* used in the two tasks on $frame_i$ is calculated based on the object moving distance from $frame_i$ to $frame_{i+1}$, so obtaining its correct value is impossible. To solve this issue, real implementations predict *v* based on the moving distance from $frame_{i-1}$ to $frame_i$, expecting that the object maintains the moving speed from $frame_{i-1}$ to $frame_{i+1}$.

3. Adaptive Frame-rate Optimization

3.1 Motivation

On the basis of the energy model introduced in Section 2.2, we analyze the fundamental impact of the frame-

[1] Our dynamic optimization algorithm does not depend on the difference in the operation complexity, so it can easily be applied to any execution platform by following the difference in the computing complexity.

rate on the total energy by assuming a constant speed of the tracked object. Fig. 1 shows the correlations between the frame-rate and energy consumption, expressed in units of E_{op}. Based on some implementations of DRAM and multimedia processor chips [6], [11], the energy ratio between an arithmetic operation and a memory access, α, is assume to be 5.0. We also assumed that the frame size $W \cdot H$ is 76,800 (320 $\times$ 240) pixels, that the template size $N \cdot M$ is $5,914.2$ pixels, and that the object speed is 10 pixels per second. We have decided to assume the frame size based on *Dog1* which is a video stream included in Tracker Benchmark [13]. Because Tracker Benchmark does not provide templates for object tracking, we have prepared template-data for each video stream. *Dog1* has seven templates and their average size is 5,914.2 pixels. An optimum frame-rate that minimizes the total energy consumption E can be found. This observation comes from the fact that against the frame-rate 1) the energy for frame inputs $E_f \cdot FR$ increases proportionally and 2) that for object tracking $E_t \cdot FR$ decreases in inverse proportion to the square of the frame-rate. We can see similar observations in Fig. 1 even if other benchmarks are assumed.

Next, we discuss the impact of object speed on energy consumption. Fig. 2 presents the total energy with varied object speed. The x-axis and y-axis are the same as those in Fig. 1. The dot marker in the figure shows the minimum point on each energy curve. As we can see from Fig. 2, the optimum frame-rate depends precisely on the object speed. Unfortunately, the conventional OLOT systems cannot exploit this feature due to the fixed frame-rate. If the target object speed is 50 pixels/s, the minimum energy can be achieved with a frame-rate of 29 fps in Fig. 2. When the object changes its speed to 10 pixels/sec, the conventional method can reduce the energy consumption by 55.3% due to the effect of reducing the size of the search-area. However, it misses a chance at reducing energy by 24.7% because the optimum frame-rate is 6 fps for the object with 10 pixels/sec moving speed.

In summary, our analysis results motivate us to exploit the frame-rate as an *energy-control knob* and dynamically tune the knob based on the object speed in order to implement energy-efficient OLOT systems.

3.2 Concept

As explained in Section 3.1, the conventional OLOT system cannot reduce wasted energy due to its fixed frame-rate. To reduce the wasted energy, we propose adaptive frame-rate optimization to reduce the energy consumption. Fig. 3 shows walkthroughs of the conventional and proposed method. The conventional method decides the search-area by using the estimated object speed. In Fig. 3(a), the conventional method sets the search-area widely at $frame_3$ because of the high object speed at $frame_2$. In addition, it reduces the search-area size at $frame_4$ and $frame_5$.

However, the proposed method adaptively optimizes not only the search-area but also the frame-rate. As explained in Section 3.1, the proposed method attempts to minimize the energy consumption by taking into account the energy trade-off between the number of the processing frames and the search-area. Until $frame_3$, the behavior of proposed method is the same as that of the conventional method, because the proposed method processes high frame-rate due to its high object speed. If the estimated object speed is low, the proposed method reduces the frame-rate, i.e., it decreases the number of frames that are used for object tracking. For instance, the proposed method does not track the object at $frame_4$ in Fig. 3(b). Instead, the search-area is expanded to continue tracking object at $frame_5$.

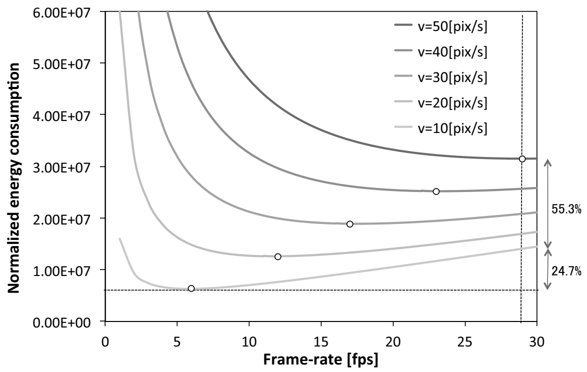
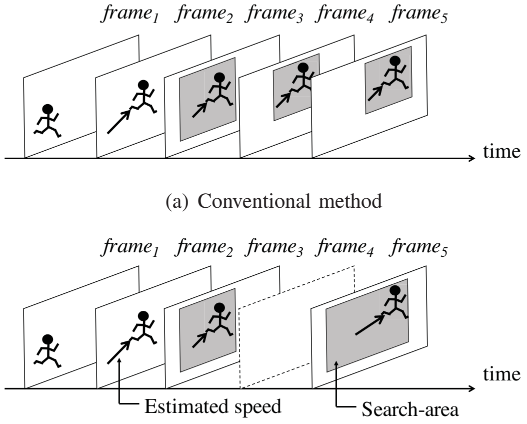


Fig. 2: Relationship between energy consumption, frame-rate, and object speed.


frame1 frame2 frame3 frame4 frame5
time
frame1 frame2 frame3 frame4 frame5
time
Estimated speed
Search-area

(a) Conventional method

(b) Proposed method

Fig. 3: Walkthroughs of conventional and proposed method.

3.3 Implementation

The object speed prediction is expected to be always accurate. Unfortunately, this assumption is not practical because lowering the frame-rate potentially can cause miss-predictions when the object quickly changes its moving speed. Predicting the object speed to be higher than the ac-

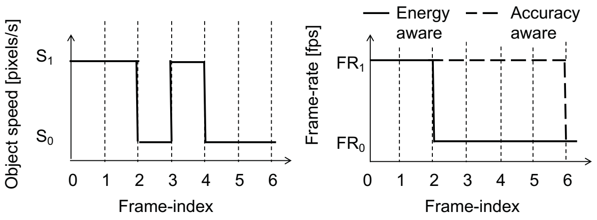


Fig. 4: Frame-rate control in energy and accuracy aware optimization.

tual speed causes a false positive, so we dissipate the energy consumption but maintain the tracking accuracy. However, a false negative, i.e., miss-predictions at a lower speed, causes a serious accuracy problem and is unacceptable for especially mission-critical applications such as driving assists. Here, we focus on alleviating the deterioration in tracking accuracy and deal with false negative cases.

If the object speed variation is continuously significant, miss-prediction at low speed is caused. Fig. 4 on the left shows an example scenario of an object moving, and the object speed varies suddenly after $frame_2$. The figure on the right describes a comparison of frame-rate transitions in two optimizations, as explained below. As shown in the solid-line of Fig. 4 on the right, the frame-rate is reduced at $frame_2$ in the optimization that updates the frame-rate with each processed frame. Because it skips some frames and cannot notice speed variation when the object speed quickly increases after $frame_2$, the optimization miss-predicts as a lower speed. This energy-aware optimization is expected to reduce more energy consumption due to the aggressive frame-rate update.

To prevent miss-predicting, the accuracy-aware optimization reduces the frame-rate only when the object speed is stable. Here, we define NF and S_{th} in accuracy-aware optimization. NF is the number of most recent compared frames, and this parameter is utilized to take into account stability of the object speed. Also, S_{th} is defined as the threshold of the speed variation. The amount in which the object speed decreases remains within S_{th}, as shown in the dashed line of Fig. 4 on the right.

Algorithm 1 shows the pseudo code of the proposed adaptive frame-rate optimization method. In lines 1-5, the proposed method executes the following processes, 1) capturing a current frame, 2) defining a search-area based on the object speed obtained in the previous iteration, 3) finding the object in the search-area, 4) and calculating the object speed, the same as in the conventional OLOT system with a fixed frame-rate, as explained in Section 2.1. After these processes, 5) the proposed method optimizes the frame-rate based on the estimated object speed to minimize the energy consumption (lines 6-16). Here, if multiple target objects exist, the fastest object speed is utilized. The frame-rate is maintained if the difference in object speed is more than S_{th} between the current frame and past them until NF (lines 6-13). In contrast, when the decline in speed is lower than S_{th}, the frame-rate is decided based on the current object speed (lines 14-16). Because the frame-rate is set to high as NF increases or S_{th} decreases, it is expected to improve tracking accuracy and increase energy consumption.

Algorithm 1 Adaptive frame-rate optimization

```
Require: $t, NF, S_{th}$
Ensure: $coordinate[i]$
 1: while $SystemIsRunning()$ do
 2:     $frame[i] \leftarrow TakeFrame(t)$
 3:     $search\_area \leftarrow DecisionSearchArea(coordinate[i-1], speed[i-1])$
 4:     $coordinate[i] \leftarrow Tracking(frame[i], search\_area)$
 5:     $speed[i] \leftarrow Prediction(coordinate)$
 6:     $flag \leftarrow false$
 7:     for $j \leftarrow 1$ to $NF$ do
 8:         if $speed[i-j] - speed[i] > S_{th}$ then
 9:             $flag \leftarrow true$
10:         end if
11:     end for
12:     if $flag$ is true then
13:         $FR_{next} \leftarrow FR_{current}$
14:     else
15:         $FR_{next} \leftarrow DecisionFR_{min}(speed[i])$
16:     end if
17:     $i \leftarrow i + 1$
18: end while
```

The optimized frame-rate FR_{min} can be decided by solving the derivative of Equation (8) with respect to FR.

$$FR_{min} = 2 \cdot \sqrt{\frac{6 \cdot N \cdot M + 3}{\alpha \cdot W \cdot H + 6 \cdot N \cdot M + 3}} \cdot v. \tag{9}$$

Here, the four parameters, α, W, H, and $n_{matching}$ are known because they are decided at design time. Thus, only the object speed v is required for the dynamic optimization, and it is predicted in line 5 the same as in search-area selection.

4. Evaluation

4.1 Experimental Setup

We evaluate the tracking accuracy and energy consumption for the following OLOT models.

- FIX : We use a conventional OLOT system with a 30 fps fixed input frame-rate. The size of the search-area is decided based on the predicted speed of the object as explained in Section 2.1.
- $ADAPT_{EN}$: This is an energy-aware implementation of our dynamic frame-rate optimization explained in Section 3.3. The parameter NF is set to zero. The model can choose the input frame-rate out of 1, 2, 3, 4, 5, 6, 10, 15 and 30 fps.
- $ADAPT_{AC}$: We use an accuracy-aware implementation of the proposed method explained in Section 3.3. The model has the same specifications with $ADAPT_{EN}$ except that the two parameters to avoid

lowering the frame-rate excessively, NF and S_{th}, are set to 5 frames and 30 pixels/s, respectively. We have decided these values empirically.

- FIX_{ideal} : This is the same as FIX except that the object speed is perfectly predicted on each frame. This ideal model can always apply the minimum sizes of search-area without degrading tracking accuracy.
- $ADAPT_{ideal}$: We use an ideal model of $ADAPT_{EN}$ which assumes perfect object-speed prediction. Both the search-area size and frame-rate are minimized without degrading tracking accuracy.

To implement the OLOT models as a simulator, we use the OpenCV library [10]. A 9×9 Gaussian filter is applied against each search-area to enhance the tracking accuracy. Because the energy impact of such filter operations is very small compared to that of NCC calculations, we do not take this energy cost into account. The first 600 frames of each video stream from Tracker Benchmark v1.0 [13] are used, and we assume that the video frame-rate is 30 fps. We prepare templates by clipping the target object from some frames in each benchmark. Here, we focus on our evaluation of the single object tracking for verification and analysis of our method. We will evaluate multiple object tracking in future work.

The simulator calculates the tracking accuracy and energy consumption. The evaluation index of the tracking accuracy on each frame, called *frame-level accuracy (FLA)*, is an overlap ratio between correct answer rectangle A provided by the benchmark and output rectangle generated by the OLOT models Q, and is defined as follows:

$$FLA = \frac{area(A) \cap area(Q)}{area(A) \cup area(Q)} \times 100\ [\%], \qquad (10)$$

where $area(A)$ and $area(Q)$ are the area of rectangle A and Q, respectively. For each benchmark, we define the average of *FLA* across all frames as *application-level accuracy (ALA)*. Equation (8) is used to obtain the energy consumption for each OLOT model. In this equation, 30 fps on FIX, FIX_{ideal}, and FR_{min} calculated by Equation (9) on $ADAPT_{EN}$, $ADAPT_{AC}$, and $ADAPT_{ideal}$, are used. We also assume that the energy ratio α is 5 as discussed in Section 3.1.

4.2 Tracking Accuracy

Fig. 5 shows the tracking accuracy of three realistic models, FIX, $ADAPT_{EN}$, and $ADAPT_{AC}$. The x-axis shows the benchmark videos, and the y-axis is the application-level accuracy *ALA*. *ALA* should be greater than or equal to 50% as a criterion [2]. From the results, we can see that FIX has as higher accuracy for all benchmarks, more than 80% on average, than the required threshold. Although our energy-aware frame-rate optimization $ADAPT_{EN}$ can almost maintain the accuracy of FIX for *Dudek* and *FaceOcc1*, it causes poor results in many benchmarks, 6.7% in the worst case for *Dog1* and 47.2% on average. As we expect, $ADAPT_{AC}$ achieves comparable accuracy to FIX for many benchmarks, *Dog1*, *Dudek*, *FaceOcc2*, and *FleetFace*, and produces results that are sufficiently higher than the threshold of 50%. Detailed analysis of the results is discussed in Section 4.4.

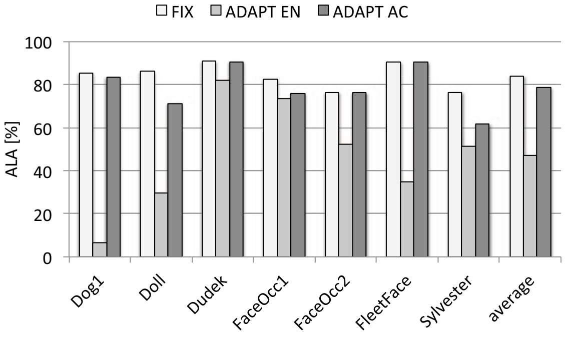


Fig. 5: Tracking accuracy results.

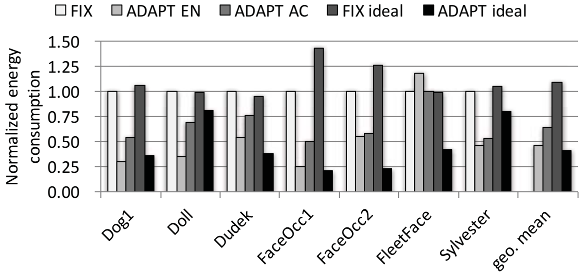
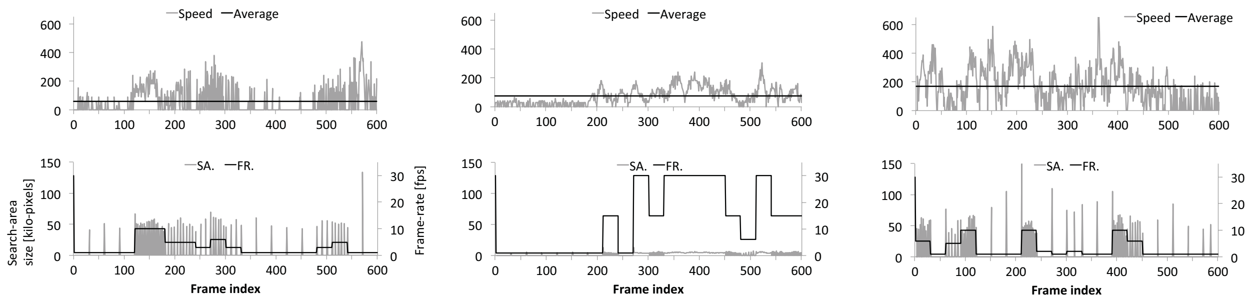


Fig. 6: Energy consumption results.

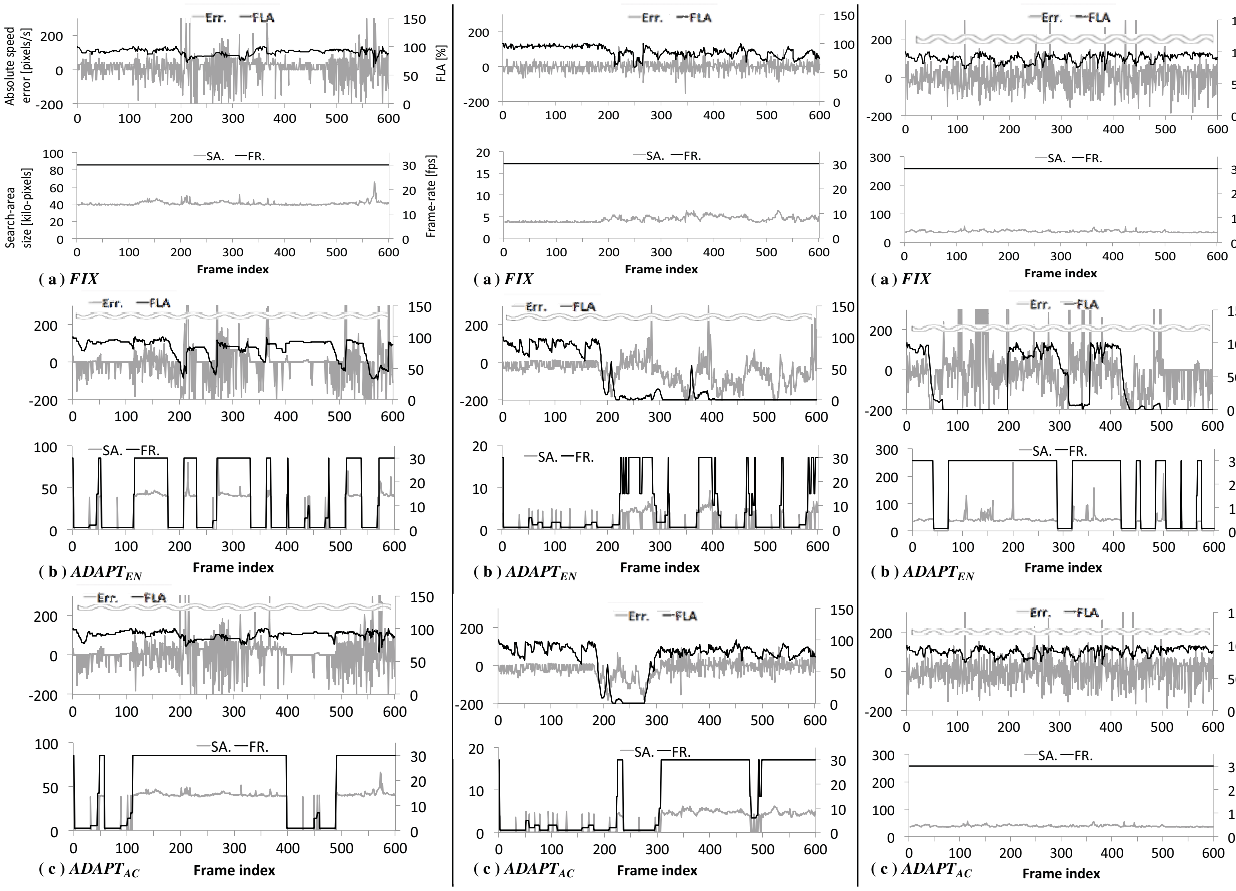
4.3 Energy Consumption

Fig. 6 shows the results of the total energy consumption. The x-axis shows the benchmarks and the y-axis is the energy consumption normalized to FIX. First, we focus on the ideal models, FIX_{ideal} and $ADAPT_{ideal}$. Compared to FIX, the ideal model with 30 fps frame-rate consumes slightly more energy due to the 100% accurate tracking. This energy increase does not appear in $ADAPT_{ideal}$, because the effect of energy reduction achieved by lowering the frame-rate is much larger, resulting in 50.2% of energy saving on average. These results demonstrate that optimizing the frame-rate has considerable potential to reduce energy consumption. Next, we show comparison of the proposed methods, $ADAPT_{EN}$ and $ADAPT_{AC}$ with FIX. We can see that $ADAPT_{EN}$ reduces the energy consumption by 74.8% in the best case (*FaceOcc1*) and more than 53.7% on average. An interesting observation is that for *FleetFace* it consumes more energy than FIX. Although the accuracy-aware method is more energy consuming than the energy-aware method, $ADAPT_{AC}$ has an sufficient advantage,



(A) *Dudek* (B) *Doll* (C) *FleetFace*

Fig. 7: Transitions of object speed (upper), search-area size (SA), and frame-rate (FR) on $ADAPT_{ideal}$ (lower).



(a) *FIX* (a) *FIX* (a) *FIX*

(b) $ADAPT_{EN}$ (b) $ADAPT_{EN}$ (b) $ADAPT_{EN}$

(c) $ADAPT_{AC}$ (c) $ADAPT_{AC}$ (c) $ADAPT_{AC}$

Dudek **Doll** **FleetFace**

Fig. 8: Transitions of FLA, absolute speed error (Err.), search-area size (SA), and frame-rate (FR) for *Dudek* (left), *Doll* (center), *FleetFace* (right) on FIX, $ADAPT_{EN}$ and $ADAPT_{AC}$.

50.0% of energy reduction at maximum (*FaceOcc1*) and 35.9% on average.

4.4 Discussion

In this section, we analyze the effects of frame-rate optimization in detail from the viewpoint of accuracy and energy. Due to a lack of space, we limit the discussions to three representative benchmark videos, *Dudek*, *Doll*, and *FleetFace*. In Fig. 7, the upper graph shows the object speed on each frame and its average in pixels/s, and the lower graph shows the transition of search-area size in kilo-pixels and frame-rate on $ADAPT_{ideal}$. By referring to the figures, we can understand the benchmark characteristics and its ideal frame-rate control. For the three benchmarks, Fig. 8 gives us the detailed behavior of FIX, $ADAPT_{EN}$, and $ADAPT_{AC}$. We can see two graphs associated with each benchmark (in column) and each OLOT model (in row). The upper one can be used to discuss the tracking accuracy.

Err. is the absolute difference in pixels/s calculating as subtraction in the real speed from its estimation. Because negative *Err.* means that the estimated speed is less than real moving speed, it potentially can cause the low accuracy explained in Section 3.3. *FLA* is the frame-level accuracy defined in Equation (10). The lower one is useful to consider the energy trade-off because it shows the transition of the search-area size and frame-rate. The following subsections refer to the figures.

4.4.1 *Dudek*

Here, we focus on Fig. 7(A) and the first column in Fig. 8(a)(b)(c). In this benchmark, both $ADAPT_{EN}$ and $ADAPT_{AC}$ achieve high accuracy and energy efficient object tracking. A disadvantage of $ADAPT_{EN}$ is to quickly response to object speed change. As we see in Fig. 7(A), the frame-rate around $frame_{200}$ can be reduced, but should be maintained at least 5 fps. However, as reported in Fig. 8(b), $ADAPT_{EN}$ excessively lowers the frame-rate before $frame_{200}$ and around $frame_{250}$, resulting in accuracy degradation. Because it returns quickly to a higher frame-rate, the object is accurately tracked. However, in these frame points, $ADAPT_{AC}$ does not apply a low frame-rate (Fig. 8(c)), resulting in slightly higher energy but accurate tracking compared to $ADAPT_{EN}$.

4.4.2 *Doll*

We look at Fig. 7(B) and the second column in Fig. 8(a)(b)(c). $ADAPT_{EN}$ causes poor accuracy, failing to track as shown in Fig. 8(b). In this benchmark, the object tends to increase the speed after $frame_{200}$, and the system should not aggressively lower the frame-rate as shown in Fig. 7(B). However, as we can see in Fig. 8(b), *Err.* goes in a negative direction largely, causing a serious accuracy reduction, and cannot recover until the end. This is the main reason for the accuracy of $ADAPT_{EN}$ being low. Although $ADAPT_{AC}$ also degrades the FLA for the same reason, it can track the target after $frame_{300}$ because it maintain a high frame-rate due to significant speed variation.

4.4.3 *FleetFace*

Next, we look at Fig. 7(C) and the third column in Fig. 8(a)(b)(c). In this benchmark, $ADAPT_{AC}$ cannot achieve energy reduction. This is because as shown in Fig. 7(C), the moving speed variation of the object is significant. This means that $ADAPT_{AC}$ attempts to maintain a high frame-rate as presented in Fig. 8(c) which is similar to (a). However, $ADAPT_{EN}$ has an accuracy problem, as you can see in Fig. 8(b) due to the significant speed variation. After degradation in the FLA, $ADAPT_{EN}$ expands the search-area size repeatedly around $frame_{150}$, $frame_{200}$, and so on. This is because the energy consumption of $ADAPT_{EN}$ is increased compared to that of FIX.

5. Conclusions

We propose a new energy reduction method based on adaptive frame-rate optimization for an OLOT system with adaptive search-area selection. Lowering the frame-rate potentially can cause miss-predictions to miss-predict when the object quickly changes its moving speed. To alleviate this issue, the proposed method attempts accuracy-aware tracking by decreasing the sensitivity of the frame-rate to the variations in speed.

We evaluate the proposed method using the energy model in terms of the tracking accuracy and the energy consumption. These results show that the accuracy-aware method reduces the energy consumption by up to 50.0% and 35.9% on average without causing serious degradation in the average tracking accuracy. We will evaluate the proposed method in terms of energy consumption in actual equipment in future work.

Acknowledgment

This research was supported in part by JSPS KAKENHI Grant Number 16H02796 and 16K16027.

References

[1] N. Cottini *et al.* A 33μw 42 gops/w 64x64 pixel vision sensor with dynamic background subtraction for scene interpretation. In *Proc. of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 315–320, July 2012.

[2] M. Everingham *et al.* The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, pages 98–136, June 2014.

[3] J. Gonzalez-Mora *et al.* Efficient image alignment using linear appearance models. In *Proc. of the 2009 Computer Vision and Pattern Recognition*, pages 2230–2237, June 2009.

[4] H. Han *et al.* E3: Energy-efficient Engine for Frame Rate Adaptation on Smartphones. In *Proc. of the 11th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14, Nov. 2013.

[5] S. Hanson and D. Sylvester. A 0.45 0.7V sub-microwatt CMOS image sensor for ultra-low power applications. In *Proc. of the 2009 Symposium on VLSI Circuits*, pages 176–177, June 2009.

[6] Y. Kikuchi *et al.* A 40 nm 222 mW H.264 Full-HD Decoding, 25 Power Domains, 14-Core Application Processor With x512b Stacked DRAM. *IEEE Journal of Solid-State Circuits*, 46(1):32–41, Jan. 2011.

[7] R. LiKamWa *et al.* Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proc. of the 11th annual international conference on Mobile systems, applications, and services*, pages 69–82, June 2013.

[8] L. McIlrath. A low-power low-noise ultrawide-dynamic-range CMOS imager with pixel-parallel A/D conversion. *IEEE Journal of Solid-State Circuits*, 36(5):846–853, May 2001.

[9] X. Mei *et al.* Minimum error bounded efficient $\ell 1$ tracker with occlusion detection. In *Proc. of the 2011 Computer Vision and Pattern Recognition*, pages 1257–1264, June 2011.

[10] OpenCV. http://opencv.org/.

[11] Y. Tanabe *et al.* A 464GOPS 620GOPS/W heterogeneous multi-core SoC for image-recognition applications. In *Proc. of the 2012 IEEE International Solid-State Circuits Conference*, number 11, pages 222–223, Feb. 2012.

[12] H. Usui *et al.* An evaluation of an energy efficient many-core SoC with parallelized face detection. In *Proc. of the 2014 19th Asia and South Pacific Design Automation Conference*, pages 311–316, Jan. 2014.

[13] Y. Wu *et al.* Online Object Tracking: A Benchmark. In *Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, June 2013.

Proposition of ESMM for Moving Object Detection

Hyojin Lim[1], Yeongyu Choi[1], S. M. Lee[2], and Ho-Youl Jung[1,3]
[1,3]Department of Information and Communication Engineering, Yeungnam University,
Gyeongsan, Republic of Korea
[2]School of Electronics Engineering, Kyungpook National University,
Daegu, Republic of Korea
[3]Correspondence: Prof. Ph. D. Ho-Youl Jung (hoyoul@yu.ac.kr)

Abstract – *In this paper, a new method for detecting moving objects in static scenes is proposed. The method is based on pyramidal Lucas-Kanade optical flow algorithm to estimate the motions of the moving objects. Extended Salient Motion Map (ESMM) is introduced to obtain more information of motion vectors. Gaussian Mixture Model (GMM) is applied to ESMM for discriminating of foregrounds and background. Through the simulations tested on the real videos of BMC2012 dataset, we show that the proposed method is competitive to RGB based GMM segmentation, in terms of moving object detection performances.*

Keywords: Background subtraction, Gaussian Mixture Model, Moving object detection, Optical flow, ESMM, BMC 2012 dataset

1 Introduction

Moving object detection technique is useful for object recognition. In detail, the results of detection step are the input data for classification step. The background subtraction technique is one of popular techniques for detection of moving objects. Many researchers have developed background modeling techniques for static scenes that are captured by fixed-camera, such as surveillance systems. One basic technique for moving object detection is frame differencing. In order to model the background, consecutive frames are accumulated. For detection of moving objects, modeled background is subtracted from the new frame. Finally, we can extract moving objects. However, frame differencing technique has a big problem that it cannot update the background. One of popular adaptive background modeling techniques is GMM [1], [2]. In [3], GMM shows the best performance for moving object detection in the static scenes. However, in order to implement GMM for moving object detection, the temporal information should be considered. The other technique is using optical flow extraction. In [4], the modified GMM is used for background modeling, and optical flow is used for detection of moving objects. Other techniques for background modeling are median filter [5], approximated median filtering technique [6], and Kalman filter technique [7].

In this paper, we propose a new method for moving object detection. Our approach is based on motion flows. By using motion flows, we introduced a map which is called Extended Salient Motion Map (ESMM). Firstly, to estimate feature points, we applied Shi-Tomasi corner detection [8]. Based on feature points, pyramidal Lucas-Kanade optical flow technique is applied to estimate motion flows. After ESMM mapping, median filter is applied. Next, GMM is used to segment the moving objects from ESMM. For simulations, we used the real videos of BMC2012 dataset [13] and compared to the moving object detections of RGB based GMM and ESMM based GMM. The results of both methods are evaluated by BMC Wizard [14].

The rest of the paper is organized as follows. In the next section, we describe the related work. In section 3, we explain the proposed method which is ESMM. In section 4, experimental results are described. Finally, the conclusion and future work conclude in section 5.

2 Related works

In this paper, the motion flows of moving objects are estimated by Lucas-Kanade optical flow algorithm. The ESMM is stablished based on estimated motion flows. ESMM is applied to GMM for detection of moving object. Therefore, this section aims to summarize briefly Lucas-Kanade optical flow technique and Gaussian Mixture Model.

2.1 Lucas-Kanade Optical flow [9]

In order to estimate motion flows, we applied Lucas-Kanade optical flow technique [9]. The intensity variation over continuous frames is analyzed to estimate motion flow. One of other optical flow techniques was introduced by Horn-Schunck [11]. The main difference between Horn-Schunck and Lucas-Kanade optical flow is that optical flow is estimated by global or local locations. The performance evaluation of these optical flow techniques can be found in [12]. Lucas-Kanade optical flow technique assumes that optical flows are the same within a window region $N(x,y)$ at center point $f(x,y)$. One more assumption is that objects do not move far between frame to frame. Optical flow constraint equation with representing window is given by

$$\frac{\partial f(x_i,y_i)}{\partial y}v+\frac{\partial f(x_i,y_i)}{\partial x}u+\frac{\partial f(x_i,y_i)}{\partial t}=0 \qquad (1)$$

$$(x_i,y_i)\in N(x,y)$$

Equation (1) can represent to matrix format (2). Using the least square method, unknown values u, v can be found.

$$\mathbf{A}\mathbf{v}^{\mathrm{T}} = \boldsymbol{b}$$

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f(x_1,y_1)}{\partial y} & \frac{\partial f(x_1,y_1)}{\partial x} \\ \vdots & \vdots \\ \frac{\partial f(x_n,y_n)}{\partial y} & \frac{\partial f(x_n,y_n)}{\partial x} \end{pmatrix}, \mathbf{v} = (u \quad v), \tag{2}$$

$$\boldsymbol{b} = \begin{pmatrix} -\frac{\partial f(x_1,y_1)}{\partial t} \\ \vdots \\ -\frac{\partial f(x_n,y_n)}{\partial t} \end{pmatrix}$$

Equation (2) is solved to $\mathbf{v}^{\mathbf{T}} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}\boldsymbol{b}$ in order to find unknown values u, v that are motion information.

$$\mathrm{v}^{\mathrm{T}} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{3}$$

$$= \begin{pmatrix} \sum_{i=1}^{n} \left(\frac{\partial f(x_i,y_i)}{\partial x}\right)^2 & \sum_{i=1}^{n} \frac{\partial f(x_i,y_i)}{\partial x}\frac{\partial f(x_i,y_i)}{\partial y} \\ \sum_{i=1}^{n} \frac{\partial f(x_i,y_i)}{\partial x}\frac{\partial f(x_i,y_i)}{\partial y} & \sum_{i=1}^{n} \left(\frac{\partial f(x_i,y_i)}{\partial y}\right)^2 \end{pmatrix} \begin{pmatrix} -\sum_{i=1}^{n} \frac{\partial f(x_i,y_i)}{\partial y}\frac{\partial f(x_i,y_i)}{\partial t} \\ -\sum_{i=1}^{n} \frac{\partial f(x_i,y_i)}{\partial y}\frac{\partial f(x_i,y_i)}{\partial t} \end{pmatrix}$$

Lucas-Kanade optical flow shows good performance. However, Lucas-Kanade optical flow has some disadvantages. One of disadvantage is that the motion flow is described within a small window, so the fast moving object cannot be detected. In order to overcome this weak point, pyramidal technique was introduced in [10]. The other disadvantage is that optical flow cannot be estimated at the low illumination conditions.

2.2 Gaussian Mixture Model

The GMM is a mixture of K Gaussian distributions that representing the distribution of pixel intensities in current frame. In [3], GMM shows the best performance of moving object detection among for four techniques. The probabilities of intensities of a frame at time t is modeled as:

$$P(x_t) = \sum_{k=1}^{K} w_{k,t} \times N(x_t, \mu_{k,t}, \Sigma_{k,t}) \tag{4}$$

where, $w_{k,t}, \mu_{k,t}, \Sigma_{k,t}$ are weight estimation, mean, and covariance matrix of Gaussian *k*-th, respectively. In this paper, we assumed that GMM components are independent. Thus, the covariance matrix of Gaussian *k*-th is referred from standard deviation as (5)

$$\Sigma_{k,t} = \sigma_{k,t} \times I \tag{5}$$

The new pixel intensity x_t is estimated with respect to each Gaussian component to find the nearest distribution where it should belong to. Then, the new parameters $w_{k,t}, \mu_{k,t}, \Sigma_{k,t}$ are updated. By using the GMM, the set of background pixels is characterized adaptively follow temporal domain. Therefore, it is useful to use GMM to eliminate all of possible background pixels to obtain the foregrounds. In another word, GMM based moving object detection is robust against such illumination changing.

3 Extended Salient Motion Map

The distribution of motion vectors from corner point are very sparse. The motion vectors are not good enough applying to background subtraction method. In order to enrich information, we propose Extended Salient Motion Map (ESMM), on which motion vectors are extended to neighbors region. For given vector $\mathbf{v} = (u, v)$ on (*x, y*) point, its neighbor within the extended window have the same vector features of $\mathbf{v}$. If a location is overlapped with more than two other windows, it has average feature of the overlapping vectors. We use the amplitudes of motion vector for the vector features selection. Flow chart of ESMM is shown in Fig. 1. Firstly, the input image is converted to gray image to estimate corner points. Corner extraction step is applied by using Shi-Tomasi corner detection. Using corner points, we applied pyramidal Lucas-Kanade optical flow technique is used to estimate motion flows. And then, we calculate feature of motion vectors, such as amplitudes and orientations. Then, using extended window, ESMM is obtained by using the amplitudes of the estimated motion vector that are inside the extended window.

Figure 2 shows the ESMM mapping. Red arrows are motion vectors, the coordinated of the head points of the red arrows are represented as $(x_i + u_i, y_i + v_i)$, and the red points (x_i, y_i) are the corner points of the previous frame. The extended window can be established by using the head and tail coordinates. The height of the extended window in Fig. 2 is equal to the amplitude value of motion vectors.

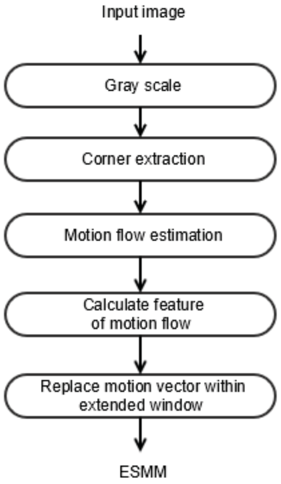

Input image
Gray scale
Corner extraction
Motion flow estimation
Calculate feature of motion flow
Replace motion vector within extended window
ESMM

Fig. 1 flow chart of ESMM

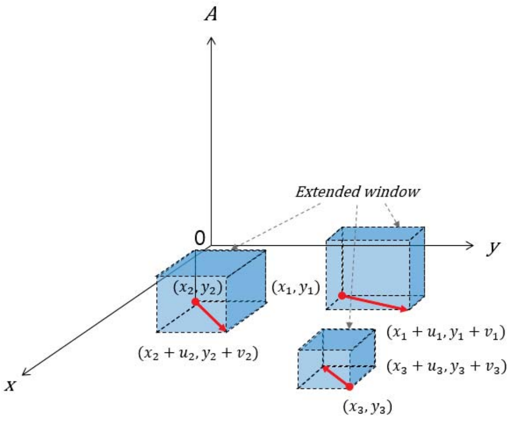


Fig 2. ESMM mapping

The size of width and length of the extended windows are determined by using the head and tail points of motion vectors. After estimating the extended window, the amplitudes of motion vectors are normalized to [0 . . 255]. And then, the normalized values are assigned to ESMM within the window. Fig. 3 is an example of ESMM result. Compared to the original frame, the ESMM has rich information. It is useful to segment the moving objects.

4 Experimental results

In this paper, 9 real videos of BMC 2012 dataset [13] are used. The median filter is used for pre-processing of tested videos. GMM algorithm is used for detection of moving objects. We tested 2 kinds of methods. One is RGB based GMM, the other is ESMM based GMM for detection of moving objects. For evaluation, we used the BMC Wizard. Table 1 shows descriptions for every video. Table 2 shows the results of the GMMs. In case of ghost, not much noises, small waving rope, and abrupt changing lights, ESMM based GMM shows better performance compared to RGB based GMM, However, ESMM based GMM gets disadvantages. Since ESMM is computed from motion information, so ESMM has no information in the region without motion vectors. The other disadvantage is that ESMM cannot extract exactly the shapes of moving objects. In case of video 3, the performance of ESMM based GMM is not very high, because pedestrians have not much motion vectors. In video 4, ESMM based GMM shows gets not very high performance, because the consecutive two frames are the same. In video 7, the result shows similar performance between ESMM and RGB. Fig. 4 shows some result of each type with GMM.

5 Conclusion and future work

In this paper, we proposed ESMM for detection of moving objects. ESMM is based on motion flows. ESMM has positions and characteristics of motion flows. ESMM shows motion spread effect, so even though motion flows could not be detected, it can estimate motion information, using neighbor motion information. We tested BMC 2012 dataset, and BMC Wizard was used for evaluation. The results show that the proposed method is competitive to RGB based GMM, in terms of moving object detection performances. In case of ghost, less noise, and sudden changing light, ESMM shows better performance. However, it could not extract exact shape of moving objects. Then, if the image could not estimate motion information, ESMM cannot be implemented.

For the future work, we will compare the computational cost between motion flows at all locations of the scene and ESMM.

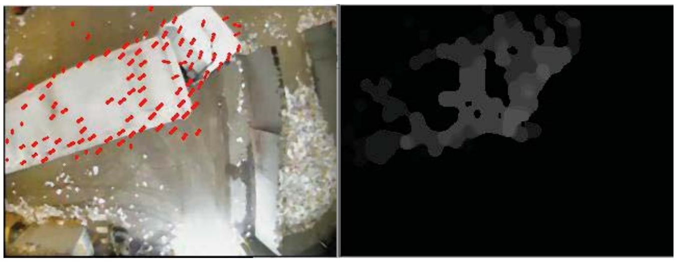


Fig. 3 Example of ESMM: original image with motion flows (left), and ESMM (right)

Table 1. Description of test video.

Video number	Number of frames	Description
1	32965	Trees are waving, small object at parking lot
2	1498	Big tracks and small humans with noise and illumination changing.
3	794	Pedestrians at three-way
4	1895	Camera noise, human and rabbit are passing in the night.
5	117149	Snowing
6	1064	Train and vehicles are passing at a railroad crossing
7	1726	Train and human with shadows at tunnel
8	792	Waving trees, vehicles are passing at highway
9	107817	Parking lot

Table 2. Experimental results with ESMM based and RGB based GMM background subtraction methods

Video	Type	Recall	Precision	F-measure
1	ESMM	0.684	0.595	0.637
	RGB	0.614	0.584	0.599
2	ESMM	0.737	0.753	0.745
	RGB	0.649	0.786	0.711
3	ESMM	0.632	0.617	0.625
	RGB	0.768	0.746	0.757
4	ESMM	0.658	0.642	0.650
	RGB	0.735	0.728	0.731
5	ESMM	0.791	0.517	0.625
	RGB	0.698	0.548	0.614
6	ESMM	0.836	0.724	0.776
	RGB	0.720	0.750	0.734
7	ESMM	0.620	0.644	0.632
	RGB	0.630	0.652	0.641
8	ESMM	0.574	0.520	0.546
	RGB	0.544	0.519	0.531
9	ESMM	0.622	0.539	0.578
	RGB	0.545	0.539	0.542
Average	ESMM	0.684	0.617	0.646
	RGB	0.656	0.650	0.651

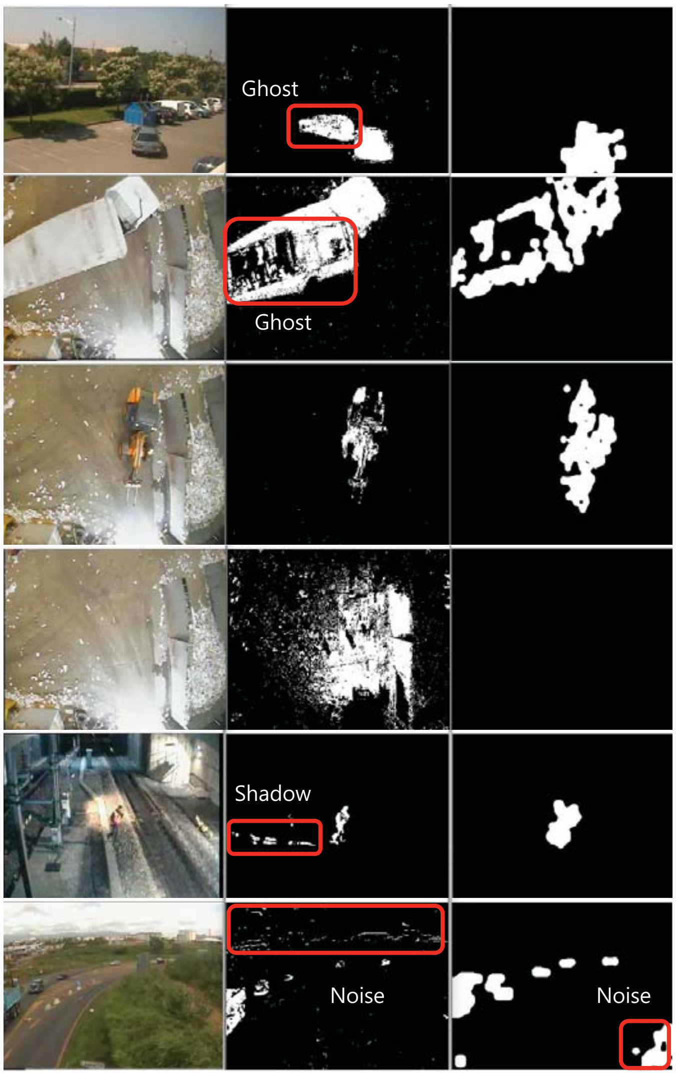


Fig. 4 Detection results of GMMs. Original frame (left column), RGB based GMM (middle column), and ESMM based GMM (right column)

6 Acknowledgment

This work was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2014H1C1A1066380)

7 References

[1] Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Vol. 2, IEEE, 1999.

[2] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 2, pp. 28-31, 2004.

[3] Sen-Ching, S. Cheung; Kamath, Chandrika. “Robust techniques for background subtraction in urban traffic video.” Electronic Imaging 2004. International Society for Optics and Photonics, pp. 811-892, 2004.

[4] Zhou, Dongxiang, and Hong Zhang. "Modified GMM background modeling and optical flow for detection of moving objects." Systems, Man and Cybernetics, 2005 IEEE International Conference on. Vol. 3. pp. 2224-2229, 2005.

[5] Cucchiara, Rita, et al. "Detecting moving objects, ghosts, and shadows in video streams." Pattern Analysis and Machine Intelligence, IEEE Transactions on. Vol. 25, Issue 10, pp. 1337-1342, 2003.

[6] McFarlane, Nigel JB, and C. Paddy Schofield. "Segmentation and tracking of piglets in images." Machine vision and applications, Vol. 8, Issue 3, 187-193, 1995.

[7] Koller, Dieter, Joseph Weber, and Jitendra Malik. “Robust multiple car tracking with occlusion reasoning.” Springer Berlin Heidelberg, pp.189-196, 1994.

[8] Shi, Jianbo, and Carlo Tomasi. "Good features to track." Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94. 1994 IEEE Computer Society Conference on. pp. 593-600, 1994.

[9] Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." IJCAI. Vol. 81. pp. 674-679, 1981.

[10] Bouguet, Jean-Yves. "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm." Intel Corporation, Vol. 5, pp. 1-10, 2001.

[11] Horn, Berthold K., and Brian G. Schunck. "Determining optical flow." 1981 Technical symposium east. International Society for Optics and Photonics, pp. 319-331, 1981.

[12] Barron, John L., David J. Fleet, and Steven S. Beauchemin. "Performance of optical flow techniques." International journal of computer vision. Vol. 12, Issue 1, pp. 43-77, 1994.

[13] Background Models Challenge (BMC) 2012 dataset is available at http://bmc.iut-auvergne.com/?page_id=24

[14] BMC Wizard is available at http://bmc.iut-auvergne.com/?page_id=63

SESSION

NOISE REDUCTION, DATA AND SIGNAL QUALITY STUDIES, RESOLUTION ENHANCEMENT, AND RELATED ISSUES

Chair(s)

TBA

Symmetrical recursive median filter for region smoothing without edge distortion

A. Raji
Laboratory of Images, Signals and Intelligent Systems
Paris Est Creteil University, F94010, France
raji@u-pec.fr

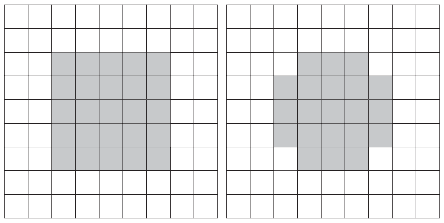
***Abstract*— In this paper, we present a new median-based operator which associates in a multidirectional filtering scheme the noise suppression ability of the recursive median filter with the edge conservation property of the standard median filter. We show in a preliminary comparative study the superiority of the recursive median filter for smoothing regions corrupted by additive white noise, while the standard median filter is showed to be better in preserving the edges with less distortion. A new operator called the symmetrical recursive median filter is then introduced aiming to improve noise suppression properties as well as to conserve accurately object contours in the filtered images. The properties of the proposed operator are discussed. The operator effectiveness for region smoothing without edge distortion is illustrated on real and synthetic image processing examples.**

***Index Terms*— median filters, multidirectional filtering, noise suppression, edge conservation, shape distortion**

I. Introduction

In image processing applications, it is wished to remove noise as well as to preserve the shape and position of edges. Linear filters are generally not very satisfactory because they blur unavoidably sharp edges. Among nonlinear techniques, order statistic have long been studied and used in different disciplines [1], [2], [3]. In particular, many signal and image processing applications are based on order statistics filtering [4], [5]. For example, median [6], [7], [8] and rank-order [9] based filters can show a great efficiency for impulsive noise suppression. The median filter is known to be increasingly effective when the distribution of noise tends towards an impulsive one. During the last decades, a multitude of median-based operators have been proposed to improve certain aspects of median filtering. For example, the FIR-median hybrid filters [10], [11] associate noise smoothing properties of FIR linear filters with the edge preservation ability of the median filter. In the same scope, the weighted median filters [12], [13] affect a big weight to the median and non null weights to the other order statistics in the filter window. The switched median filter [14], [15], [16] applies a median filter only to the pixels that are designated as impulses by a prior impulse detector, using the assumption that an effective removal of noise impulses is often accomplished at the expense of distorted features if the median filter is implemented uniformly across the image.

In fact, the median filter presents an efficient alternative to linear filtering when the preservation of contours is of primary importance and when the noise has rather an impulsive nature. However, an important remaining drawback of median-based filters when applied to images is the loss or distortion of certain geometrical features such as thin lines and sharp corners, because the 2D structure of the filter window does not allow to preserve such features as illustrated by the example of fig.1. The multidirectional filtering technique [17], [18] propose to combine the outputs of several basic 1D subfilters along different directions in the image in order to conserve such features. In this article, we propose a new median-based operator which associates the noise suppression ability of the recursive median filter with the edge conservation property of the standard median filter and geometrical features preservation by multidirectional filtering principle. We show that the proposed operator leads to enhanced performances compared to the classical standard and recursive median filters.



(a) region with sharp corners (b) result of 2D 3x3 median

Fig. 1. Distortion of the angular zones by 2D median filtering

II. Order Statistic Operators and Median Filters

By sorting a set $W = \{X_1, X_2, \cdots, X_M\}$ of M independent and identically distributed (i.i.d) random variables, we obtain an ordered sequence $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(M)}$ where $X_{(r)}$ is called the r^{th} order statistic in W. Let us consider an infinite length 1D signal $\{X(i), i = ..., -1, 0, 1, ...\}$ to be filtered by a sliding operator window $W(i) = \{X(i-m), ..., X(i), ..., X(i+m)\}$ of size $M = 2m+1$ centered at the sample $X(i)$. The output of the order statistic filter of size M is given by a linear combination of the order statistics in $W(i)$:

$$Y(i) = \sum_{j=1}^{M} a_j X_{(j)}(i) \tag{1}$$

where the real coefficients a_j verify $\sum_{j=1}^{M} a_j = 1$ for an unbiased estimation. Assuming that $\{X(i)\}$ is a constant signal corrupted by additive white noise $\{n(i)\}$, the optimal order statistic filter in the mean square error criterion is given by:

$$a = \frac{R^{-1}e}{e^t R^{-1} e} \tag{2}$$

with $a^t = (a_1 \cdots a_M)$, $a_i \; i = 1, ..., M$ representing the filter coefficients, $R = (r_{kl}) = \left(E\{n_{(k)} n_{(l)}\}\right)$, $k, l = 1, \cdots, M$ representing the M-correlation matrix of the noise order statistics vector and $e^t = (1 \cdots 1)$ the M component unitary vector. From (2) it results that the optimal quadratic order statistic filter for uniform distribution noise is the middle filter $Y(i) = \frac{1}{2}[X_{(1)}(i) + X_{(M)}(i)]$, while it is the averaging filter $Y(i) = \frac{1}{M}\sum_{j=1}^{M} X_{(j)}(i)$ for gaussian noise. Furthermore, it has been shown that the more impulsive the noise is, i.e. with less concentrated and more heavy tailed distribution, the more the optimal quadratic order statistic filter tends towards the standard median (SM) filter, which is a particular case of order statistic filters with $a_{m+1} = 1$ and $a_j = 0$ for $j \neq m + 1$:

$$Y(i) = med\{X(i-m), ..., X(i), ..., X(i+m)\} = X_{(m+1)}(i) \tag{3}$$

The median is an efficient non parametric estimator. For example, an important offset in the input sequence, i.e. a noise impulsion, has little effect on the output of the median filter, while it produces an important bias in the output of the averaging filter. Moreover, edges and monotonic changes are left invariant by median filtering while they are smoothed by linear filters. The median is also the best scale estimator in the absolute mean error criterion, and the maximum likelihood estimator for an exponential i.i.d input [19].

If we replace the point being processed by the output of the median operator before shifting the filter window to the next position, we obtain the recursive median (RM) filter defined by:

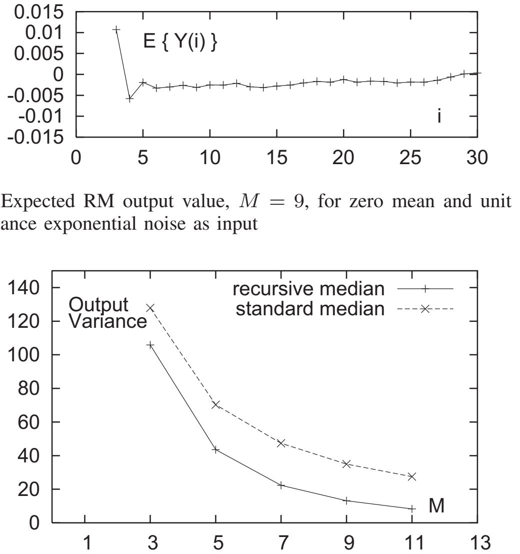
$$Y(i) \quad = \quad med\{\textbf{Y(i-m),...,Y(i-1)}, X(i), ..., X(i+m)\} \tag{4}$$

III. Noise suppression and edge conservation properties

In order to characterize the effect of the standard and recursive median filters on noisy signals, deterministic and statistical properties [20] are used instead of spectral analysis because of the nonlinearity of the median operator. Given the input probability density function f_X and distribution function F_X, the output probability density function of the standard median filter of size $M = 2m + 1$ is given by [19]:

$$f_Y = M.\frac{(M-1)!}{(m!)^2}.(F_X)^m.(1 - F_X)^m.f_X$$

In the case of the recursive median filter, there is no analytical expression of the output probability density function in the literature, and it is likely difficult to find out one because of the high correlation between the output samples, due to recursivity, especially for important filter sizes. We propose hereafter an experimental comparative study of the standard and recursive median filters properties. We will consider in this study the two main primitives in images which are regions and edges, and will carry out our analysis in 1D case for simplicity.



(a) Expected RM output value, $M = 9$, for zero mean and unit variance exponential noise as input

(b) Output variance versus filter size - exponential input noise: $\mu = 0$, $\sigma^2 = 400$

Fig. 2. Noise suppression properties of RM and SM filters

So, let us first consider, as a region model, a finite length constant signal of magnitude c corrupted by additive white noise $n(i)$: $\{X(i) = c + n(i), i = 0, 1, ..., L-1\}$. Since it is well known that the median filter is efficient for impulsive noise, we will use throughout this paper an impulsive noise model characterized by the exponential probability density function: $f_n(x) = \frac{\alpha}{2}\exp(-\alpha\,|x|), \quad -\infty < x < +\infty, \quad \alpha \in \Re$. Fig.2-a represents experimental expected output value of the recursive median filter of size $M = 9$ for zero mean and unit variance exponential noise as input. In this figure, we can see that the output of the recursive median filter shows a non stationary behavior at the beginning of the filtering operation. This is due to the progressive introduction of recursivity starting from the signal border:

$$\begin{aligned}
Y(m) \quad &= \quad med\{X(0), \cdots, X(m), \cdots, X(2m)\} \\
Y(m+1) \quad &= \quad med\{X(1), \cdots, X(m-1), \\
&\qquad \mathbf{Y(m)}, \\
&\qquad X(m+1), \cdots, X(2m+1)\} \\
Y(m+2) \quad &= \quad med\{X(2), \cdots, X(m-2), \\
&\qquad \mathbf{Y(m)}, \mathbf{Y(m+1)}, \\
&\qquad X(m+2), \cdots, X(2m+2)\} \\
&\quad \cdots \\
\text{for } i \geq 2m+1 & \\
Y(i) \quad &= \quad med\{\mathbf{Y(i-m)}, \cdots, \mathbf{Y(i-1)}, \\
&\qquad X(i), \cdots, X(i+m)\}
\end{aligned}$$

This phenomenon decreases for next samples such that the RM output tends towards the signal value and becomes stationary after a certain number of iterations. By examining the expression of the outputs of the standard and recursive median filters, we see in (3) that the output of the standard median filter is computed at every point i as the median of the M input samples around i, while in the case of the recursive median filter m previous output samples $Y(i-m), ..., Y(i-1)$ are used in addition to the $(m+1)$ input samples $X(i), X(i+1), ..., X(i+m)$ to compute the current output $Y(i)$, as expressed in (4). From a probabilistic point of view, the samples $Y(i-m), ..., Y(i-1)$ representing previous median estimations at different points are closer to the signal value c than the input samples $X(i), X(i+1), ..., X(i+m)$. Furthermore, the changes that occur when sliding the filtering window from the previous position $i-1$ to the current position i are: a) replacement of $X(i-1)$ by $Y(i-1)$, b) suppression of $Y(i-1-m)$ from the window and c) introduction of $X(i+m)$ in the window. Then, only one new input sample $X(i+m)$ is taken into account at each iteration i, so that:

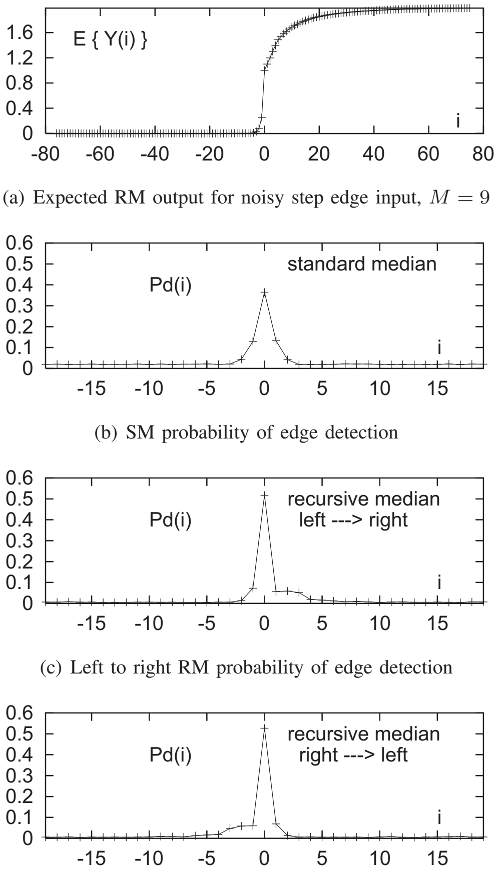
$$Y(i) = med \left\{ \begin{array}{ll} m & \text{previous median estimations:} \\ & Y(i-m), ..., Y(i-1), \\ m & \text{input samples having served in} \\ & \text{previous median estimations:} \\ & X(i), ..., X(i+m-1), \\ 1 & \text{new input sample: } X(i+m) \end{array} \right\}$$

Thanks to the impulsive noise rejection property of the median, the less the magnitude of the new sample is close to the signal value c (i.e. $X(i+m)$ represents a noise sample), the more the probability of its rejection is high. The sample $X(i+m)$ has a chance to be preserved only if its magnitude is within the dynamics of the previous median estimations $Y(i-m), ..., Y(i-1)$. Hence, as illustrated in fig.2-a, the successive recursive median estimations are closer and closer to the signal value c with more and more concentrated distribution around c as i increases, while the successive standard median estimations are all made from the input samples and then have the same distribution at any position i. This means that the recursive median filter has less output variance than the standard median filter which is confirmed in fig.2-b representing experimental output variance of the standard and recursive median filters in the case of an exponential input noise. We can then conclude that the recursive median filter is better than the standard median filter for noise suppression from regions.

In order to compare the edge preservation performances of the standard and recursive median filters, let us consider now an input containing two adjacent noisy regions:

$$Xi) = \left\{ \begin{array}{lll} n(i) & if & i < 0 \\ n(i) + A & if & i \geq 0 \end{array} \right.$$

where $n(i)$ is an i.i.d zero mean white noise, and A the magnitude of the step edge. From the previous study on a region model we can expect that when the recursive median filter reaches its stationary behavior, i.e. far from the signal borders and far from the edge, it removes noise better than the standard median filter. However, when the frontier between the two regions is encountered, the stationarity of the filter output will be lost before being progressively restored several iterations later when the filter has adapted itself to the new region value.



(a) Expected RM output for noisy step edge input, $M = 9$

(b) SM probability of edge detection

(c) Left to right RM probability of edge detection

(d) Right to left RM probability of edge detection

Fig. 3. Asymmetrical and non stationary behavior of the RM filter output near edges and Probability of edge detection with $SNR = 2$, $M = 5$, $th = 1$

This is illustrated in fig.3-a representing experimental expected output value of the recursive median filter. Indeed, this result shows an asymmetrical behavior of the recursive median filter and a deterioration of its performances near the edge. The performance deterioration affects particularly the second edge side in this example where the RM filter is applied from left to right.

In order to characterize the filter edge preservation ability, let us define an experimental parameter representing the probability of edge detection at position i by:

$$P_d(i) = P\{|Y(i) - Y(i-1)| > th\} \qquad (5)$$

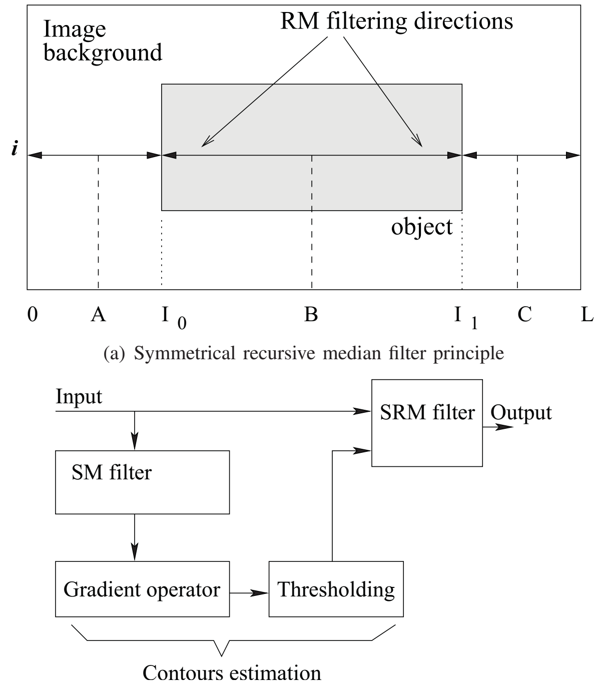
where $Y(i)$ is the filter output and th is a threshold level. Ideally, $P_d(i)$ should be equal to 1 at $i = 0$ and nil elsewhere. Fig.3-b,c,d show the experimental curves of $P_d(i)$ for the standard median, right to left and left to right recursive median filters. In this figure, we can see clearly the asymmetrical behavior of the recursive median filter around the edge point. As explained above, this is due to the loss of the stationarity of the filter output when it encounters the edge, which implies a deterioration of the filter performances during several iterations. However, we can see that the recursive median filter running from left to right performs better than the standard median filter on the left side of the edge, with higher detection probability at the edge point ($i = 0$) and lower false detection probability for $i < 0$, while it is less efficient than the standard median filter on the right side of the edge with higher false detection probability at several positions ($i > 0$). Similarly, the recursive median filter running from right to left performs better than the standard median filter on the right side of the edge with higher detection probability at the edge point ($i = 0$) and lower false detection probability for $i > 0$, while it is less efficient than the standard median filter on the left side of the edge with higher false detection probability at several positions ($i < 0$).

The main idea in the present article is to combine in an efficient way recursive median filters running in different scanning directions in order to improve noise suppression and edge preservation performances.

IV. Symmetrical recursive median filter

It results from the previous study on region and edge models that the recursive median filter is better than the standard median filter for noise suppression inside regions, while this superiority is progressively lost when edges are encountered. In this section, we propose a new operator called the symmetrical recursive median (SRM) filter, which is implemented as a multidirectional recursive median filter based on a prior estimation of the edge points in the image. The idea is that if we can do a good estimation of the edge points map in the image, then we can apply near each edge point the recursive median filter running in the direction that preserves in the best way the contour. Fig.4-a illustrates this filtering principle. Along the row number i, the object is delimited by its contours at coordinates I_0 and I_1. The proposed filtering scheme is to apply in each region the directional recursive median filter that has reached its stationary behavior. We achieve this by applying the recursive median filter running from left to right to the segments $[A, I_0]$, $[B, I_1]$ and $[C, L]$, while the recursive median filter running from right to left is applied to the segments $[O, A]$, $[I_0, B]$ and $[I_1, C]$. A, B and C are the middles of the segments $[O, I_0]$, $[I_0, I_1]$ and $[I_1, L]$ respectively.

Since the contour location constitutes the information that allows to select the optimal filtering direction, an edge points map must be estimated prior to the multidirectional filtering. Classical gradient operators are known to work well on high contrasted and noiseless images, but they are very weak in the presence of noise. We propose to achieve this preprocessing phase by the separable standard median filter since this operator has the interesting property to not displace the contours while it removes sufficiently the noise such that an acceptable estimation of the contour points can be made by means of a subsequent gradient operator. Since the directional recursive median filters have the same performance inside regions, i.e. far from the contours, where they are supposed to have reached a stationary behavior, it does not matter if the pre-processing step detects some false edge points. At such points, whichever recursive median filter is used, it will give a good result. Conversely, it is very important to detect all of the real edge points by the pre-processing step because at each of these points only one directional recursive median filter gives a good result. This means that in the gradient thresholding step, we should use rather a low threshold value to ensure that all of the real edge points are well detected. The synoptic diagram of the proposed method is shown in fig.4-b. The symmetrical recursive median filter is applied to images in a separable way, along rows and columns of the image.

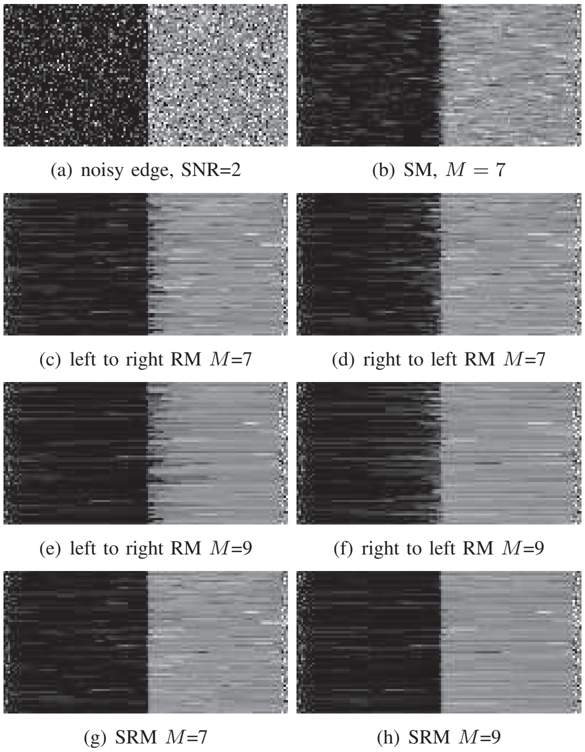


(a) Symmetrical recursive median filter principle

(b) Synoptic diagram

Fig. 4. Principle and synoptic diagram of the proposed method

However, the following limitation should be noticed for the proposed operator. We have shown that noise suppression from a region by the recursive median filter becomes optimal when the filter output reaches a stationary behavior. The filter output stationarity is broken and the filter performance is deteriorated when a contour is encountered. Consequently, if the image contains contours which are very close to each other, i.e. very small size regions, the recursive median filter will not be able to reach a stationary behavior at all, such that the symmetrical recursive median filter which aims to select the optimal directional recursive median filter will not be really useful. So, the proposed technique is not aimed for images containing fine details. In order to overcome this limitation, we propose to compute the filter output when two contour points are very close to each other (separated by a distance $\leq 3M$) by: $Y(i) = med(X(i), Y_1(i), Y_2(i))$, which is at least as efficient as the two directional recursive median estimations $Y_1(i)$ and $Y_2(i)$.


(a) noisy edge, SNR=2 (b) SM, $M = 7$

(c) left to right RM M=7 (d) right to left RM M=7

(e) left to right RM M=9 (f) right to left RM M=9

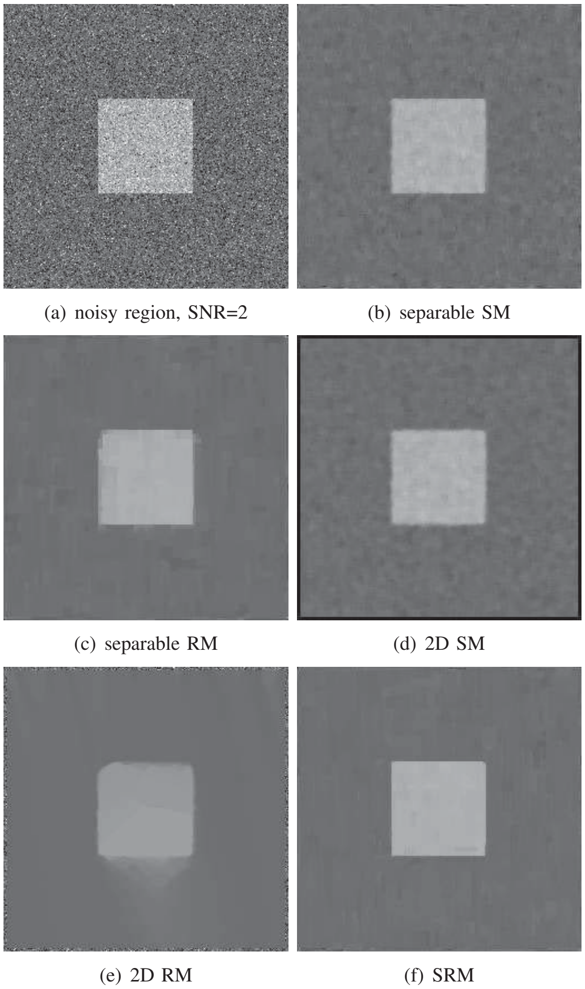
(g) SRM M=7 (h) SRM M=9

Fig. 5. Effect of SM, RM and SRM filters on a noisy step edge

V. Results

The asymmetrical behavior of the recursive median filter near edges is illustrated in fig.5. The image is filtered along rows only. The recursive median filter (fig.5-c,d) removes noise from the two regions of the image better than the standard median filter (fig.5-b). However, many rows in the filtered image show a displacement of the contour by several pixels in the processing direction, i.e. to the right for the 'left to right' recursive median filter and to the left for the 'right to left' recursive median filter. If we apply the proposed technique, assuming that the contour position (the middle column) is known, we obtain the result given in fig.5-g. In this image, the symmetrical recursive median filter removes noise from the regions and at the same time preserves well the edge with very lower distortion than in the other filters cases. Fig.5 shows also that the performance deterioration of the recursive median filter near edges is more important when the filter size increases (fig.5-c,e and d,f), while the symmetrical recursive median filter leads to satisfactory edge preservation performances with different filter sizes (fig.5-g,h).

Fig.6-a shows a noisy region with horizontal and vertical edges. The separable recursive median filter (fig.6-c) shows higher noise smoothing ability than the separable standard median filter (fig.6-b). However, it shows also some displacement of the contours in the processing directions which is particularly visible in the top left corner of the 2D area. Using 2D filtering windows allows higher noise smoothing but at the expence of distorted geometrical features. The angular zones of the object are distorted in fig.6-d,e and the resulting object in fig.6-e does not present an homogeneous gray level due to the anisotropy of the recursive median filter. In fig.6-f, the symmetrical recursive median filter preserves the shape and position of the object edges more accurately compared to the standard and recursive median operators, and at the same time, smoothes the noise efficiently from the object and the image background regions. The contour points map used as the contour information entry for the SRM filter in this example has been obtained by applying Roberts operator to the separable SM output image (fig.6-b) and thresholding the resulting gradient at level 29.

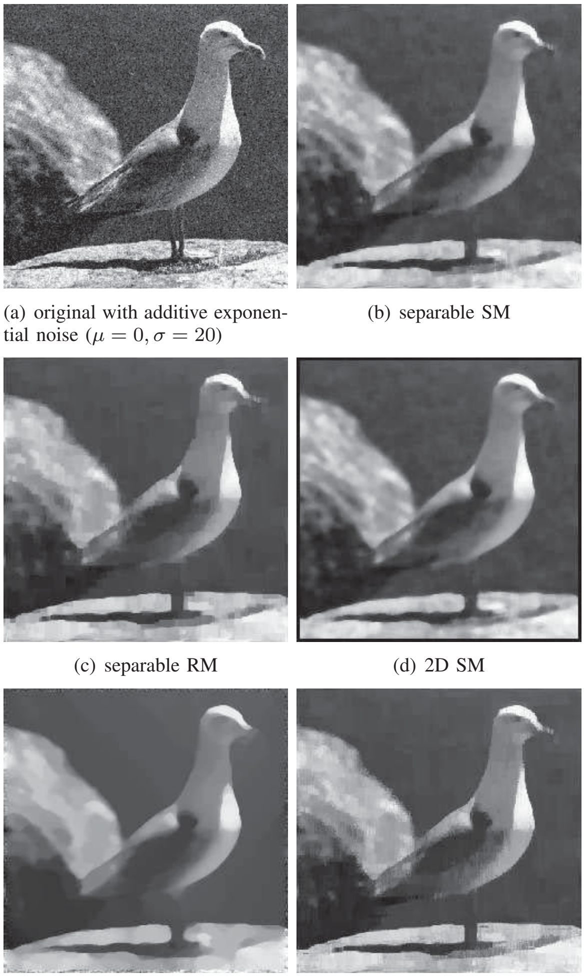

(a) noisy region, SNR=2 (b) separable SM

(c) separable RM (d) 2D SM

(e) 2D RM (f) SRM

Fig. 6. Comparison of SM, RM and SRM filters (of size 7) on a synthetic image

In the real image example of fig.7, we can see some discontinuities along the gull contours in the filtering directions in the separable RM filter output. 2D RM filter output presents higher smoothing effect but it also implies some distortions, for example the gull eye is removed and the angular zones are rounded. The symmetrical recursive median filter produces better noise suppression and edge preservation compromise than SM and RM filters.



(a) original with additive exponential noise ($\mu = 0, \sigma = 20$)

(b) separable SM

(c) separable RM

(d) 2D SM

(e) 2D RM

(f) SRM

Fig. 7. Comparison of SM, RM and SRM filters (of size 7) on a real image

VI. Conclusion

In this paper, we show that the standard median filter produces less edge distortion in images while the recursive median filter is more efficient for noise suppression inside regions but loses its performance near edges. Then, we propose a new local median based operator called the symmetrical recursive median filter which associates the advantages of the two filters in a multidirectional filtering scheme. The standard median filter is used in a pre-processing step such that a contour points map can be obtained by a subsequent gradient operator. The image is then filtered by a multidirectional recursive median filter where the contour information is used to apply near each edge point the best directional recursive median filter. The proposed method has been succesfully tested on synthetic and real images with noisy contours. The obtained results showed significant enhancement in noise suppression and edge conservation performances in comparison to the classical standard and recursive median filters. We noticed however that the proposed method is not applicable for images containing fine details with very small size objects.

References

[1] N. Balakrishnan and A. C. Cohen, *Order statistics & inference: estimation methods*. Elsevier, 2014.

[2] R.-D. Reiss, *Approximate distributions of order statistics: with applications to nonparametric statistics*. Springer Science & Business Media, 2012.

[3] H. A. David and H. N. Nagaraja, *Order Statistics*. John Wiley & Sons, Inc., 2005.

[4] A. Raji, “Detection of signal transitions by order statistics filtering,” in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV'11, Las Vegas, Nevada, USA*, pp. 757–761, CSREA Press, July 18-21 2011.

[5] C.-H. Hsieh and P.-C. Huang, “Adaptive rank order filter for image noise removal,” in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 7, pp. 90–94, March 2009.

[6] S. Akkoul, R. Ledee, R. Leconge, and R. Harba, “A new adaptive switching median filter,” *Signal Processing Letters, IEEE*, vol. 17, pp. 587–590, June 2010.

[7] K. Toh, H. Ibrahim, and M. Mahyuddin, “Salt-and-pepper noise detection and reduction using fuzzy switching median filter,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1956 – 1961, 2008.

[8] A. Toprak and I. Gler, “Impulse noise reduction in medical images with the use of switch mode fuzzy adaptive median filter,” *Digital Signal Processing*, vol. 17, no. 4, pp. 711 – 723, 2007.

[9] I. Aizenberg and C. Butakoff, “Effective impulse detector based on rank-order criteria,” *Signal Processing Letters, IEEE*, vol. 11, pp. 363–366, March 2004.

[10] V. Lyandres and S. Primak, “The fir-median suppression of impulsive interference,” *Signal Processing*, vol. 80, no. 5, pp. 883 – 887, 2000.

[11] A. Flaig, G. R. Arce, and K. E. Barner, “Affine order-statistic filters: medianization of linear fir filters,” *IEEE Transactions on Signal Processing*, vol. 46, pp. 2101–2112, Aug 1998.

[12] T. Chen and H. R. Wu, “Adaptive impulse detection using center-weighted median filters,” *IEEE Signal Processing Letters*, vol. 8, pp. 1–3, Jan 2001.

[13] G. R. Arce and J. L. Paredes, “Recursive weighted median filters admitting negative weights and their optimization,” *IEEE Transactions on Signal Processing*, vol. 48, pp. 768–779, Mar 2000.

[14] Z. Wang and D. Zhang, “Progressive switching median filter for the removal of impulse noise from highly corrupted images,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, pp. 78–80, Jan 1999.

[15] H.-L. Eng and K.-K. Ma, “Noise adaptive soft-switching median filter,” *IEEE Transactions on Image Processing*, vol. 10, pp. 242–251, Feb 2001.

[16] S. Zhang and M. A. Karim, “A new impulse detector for switching median filters,” *IEEE Signal Processing Letters*, vol. 9, pp. 360–363, Nov 2002.

[17] X. Wang, “Generalized multistage median filters,” *IEEE Transactions on Image Processing*, vol. 1, pp. 543–545, Oct 1992.

[18] V. R. Vijaykumar, D. Ebenezer, and P. T. Vanathi, “Detail preserving median based filter for impulse noise removal in digital images,” in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pp. 793–796, Oct 2008.

[19] I. Pitas and A. N. Venetsanopoulos, “Order statistics in digital image processing,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1893–1921, 1992.

[20] U. Eckhardt, “Root images of median filters,” *Journal of Mathematical Imaging and Vision*, vol. 19, no. 1, pp. 63–70, 2003.

Super-resolution Reconstruction for Diffusion-weighted Images using High Order SVD

Ying Fu[1], Yan Wang[2], Zhipeng Yang[3], and Xi Wu[1]
[1]Department of Computer Science, Chengdu University of Information Technology, Chengdu, China
[2]Department of Computer Science, Sichuan University, China
[3]Department of Electronic Engineering, Chengdu University of Information Technology, Chengdu, China

"Regular Research Paper"

Abstract **-** *The spatial resolution of diffusion-weighted imaging (DWI) is limited because of the loss of high-frequency information such as edges during data acquisition process. In this paper, method based on the patch-based super-resolution framework is proposed for single image super-resolution reconstruction of DWI dataset and the high order SVD was introduced to achieve more accurate image reconstruction and reduce the computational complexity. Experimental results demonstrate that the proposed method outperformed currently methods in both DWI reconstruction and its further applications.*

Keywords: Diffusion weighted imaging; high-order SVD; super-resolution reconstruction

1 Introduction

Diffusion Weight Imaging (DWI) is a non-invasive magnetic resonance technology and can be used to infer features of the local tissue anatomy, composition and microstructure from water displacement measurements [1]. Water does not diffuse equally and this property has been applied widely for in vivo analysis of white matter architecture and neuronal diseases [2]. Despite the rapid development and interesting property, DWI is an inherent low signal-to-noise ratio (SNR) imaging technology. Besides this, since DWI implements EPI scanning in multiple directions, the spatial resolution is relatively poor in the clinical conditions.

It has been shown that the limited resolution of DWI will introduce partial volume effect (PVE) which results bias in DWI imaging analysis [3]. The improvement of DWI spatial resolution with high SNR will provide a better sensitivity for the analysis of brain structure and clinical disease [4]. Moreover, the high resolution DWI could improve the estimation accuracy of diffusion tensor imaging, thus beneficial the fiber tractography and finer bundle analysis [5].

Several methods were proposed to enhance the spatial resolution of the DWI. In the acquisition stage, long acquisition time remains a main obstacle to be of real interest in a clinical perspective. For example, Miller et al. [6] implement five days of scanning to obtain post-mortem high resolution DWI with high SNR. To avoid the long scanning time, super-resolution (SR) acquisition emerged as an effective technology been initially proposed for MRI and then adapted into DWI soon. Sub-pixel shifting in the in-plane dimension was proposed to obtain multiple low resolution images to reconstruct the high resolution images [7]. Anisotropic scanning was another strategy to obtain low resolution images for reconstruction. Sherrer et al. [8] employed a maximum of a posteriori estimation from anisotropic orthogonal acquisition to reconstruct the isotropic high resolution DWI.

Compared with the SR acquisition implement specific scanning protocol, SR algorithm in the post-processing stage have been involved from the scene image SR reconstruction. This category method is independent of the acquisition protocol and previously implements in MRI. Manjon et al [9] implement non-local estimator to reconstruct high resolution MRI using the single low resolution dataset. Rousseau et al. [10] involved multimodality MRI to improve the SR quality. Coupe et al. [5] implemented the non-local estimator in DWI and also involved b0 information to enhance the reconstruction results. Sparse representation as a recent trend in signal and image processing were also implemented effectively in MRI. Trinh et al. [11] extended the sparse representation with nonnegative one to remove noise and super resolve together.

Single value decomposition (SVD) plays a central role in reducing high dimensional data into lower dimensional data and was involved as one of the classical method for inverse problem such as denoising [12] and restoration [13]. Recently, high order single value decomposition (HOSVD) generalized SVD of matrix into high order matrix and offered a simple and elegant method for handling similar patches [14]. Besides this, the HOSVD bases were adapted from image content and may achieve more sparse representation than the fixed bases. Have this in mind, we introduced the HOSVD into DWI super resolution. Based on the patch-based SR approaches implemented successfully in both MRI and DWI [5][9][10], HOSVD was involved to construct the regularization framework. The merit of the HOSVD SR method falls into the adaptive HOSVD bases which produce a more accurate reconstruct results. Besides this, HOSVD only implemented over a similar patches stack which effectively decreased the computation complexity. This is especially useful for DWI, since the involvement of joint information from adjacent directions DWI datasets causing extra computation burden dramatically [15].

In the remainder of this paper, we first describe the proposed method in detail, and then both synthetic and in vivo DWI datasets are involved for experimental evaluation. Experimental results and computational efficiency are demonstrated in section 4, finally the concluding remarks are given in section 5.

2 Methods

Image SR leads to an ill-posed inverse problem which is related to LR image y and HR image x, the general model is:

$$\mathbf{y} = DH\mathbf{x} + \mathbf{n} \qquad (1)$$

where **n** represents acquisition noise, *D* is decimator operator and *H* is degradation function [5][9][10].

Based on this model, the SR image can be estimated by minimizing a least-square cost function as follow:

$$\hat{\mathbf{x}} = \arg\min_{x} \| \mathbf{y} - DH\mathbf{x} \|^2 \qquad (2)$$

For such inverse problem, regularization term should be added to stabilize it [16], thus, the HR image **x** can be estimated from LR observation **y** by the following equation:

$$\hat{\mathbf{x}} = \arg\min_{x} \{ \| \mathbf{y} - DH\mathbf{x} \| + \lambda R(\mathbf{x}) \} \qquad (3)$$

where $R(\mathbf{x})$ is regulation term, $\| \mathbf{y} - DH\mathbf{x} \|$ is fidelity term, and λ is the parameter to balance them. As shown in Coupe et al. [5], an efficient way to define the regulation term is to use non-local patches methods. Instead of nonlocal mean estimator, we proposed to implement high order SVD as the estimator in this work owing to its simple application and promising performance [14].

The HOSVD estimator clusters similar patches into a stack in a similar manner as other patche-based methods [5][9][10] and then perform HOSVD transform on it to obtain the HOSVD base and coefficients. After the truncation of the coefficient, the patches were then reconstructed by inverse HOSVD transform.

Have this in mind, the regulation term of the super resolution process in equation 3 can be defined as follow:

$$R(\mathbf{x}) = \sum_{i} \left\| \mathbf{x}(i) - \psi_{HOSVD}(\mathbf{x}(i)) \right\| \qquad (4)$$

where ψ_{HOSVD} is the HOSVD based estimator.

Given a $n \times n$ patch $\mathbf{P}_i$ centered in *i*, define *K* such similar patches (including $\mathbf{P}_i$) as $\{\mathbf{P}_n\}$, where $1<n<K$, and the K-1 similar patches are obtained as follow:

Let us denote $\{\mathbf{P}_n\}$ as stack $\mathbf{Z} \in \mathbf{L}^{n \times n \times K}$, the HOSVD of the stack can be defined as [17] :

$$\mathbf{L} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \qquad (5)$$

where **S** is coefficient matrices of three order tensor with $p \times p \times K$, $\times_j$ stands for the *j*th mode tensor product defined in [17] and $\mathbf{U}^{(1)} \in \mathbf{L}^{n \times n}, \mathbf{U}^{(2)} \in \mathbf{L}^{n \times n}, \mathbf{U}^{(3)} \in \mathbf{L}^{K \times K}$ are orthonormal unitary matrix.

After HOSVD transform, the patches can be estimated by nullifying the coefficients under the assumption that the coefficients of the clean image have sparse distributions. As indicated in [14], the coefficients can be truncated using the hard thresholding as below:

$$\mathbf{S}' = H_\tau(\mathbf{S}) \qquad (6)$$

where H_τ denotes the hard threshold defined with $\tau = \sigma\sqrt{2\log(p^2 K)}$ for the stack have *K* patches with size of $\text{n} \times n$. As pointed out in [17], the coefficients in tensor **S** are not necessarily positive, and the hard thresholding is defined on the absolute value of the coefficient array:

$$H_\tau(\mathbf{S}) = \begin{cases} S_i \ \ if \ \ abs(\mathbf{S}_i) \geq \tau \\ 0 \ \ if \ \ abs(\mathbf{S}_i) \leq \tau \end{cases} \qquad (7)$$

Where $\mathbf{S}_i$ denotes the *i*th element of tensor **S**.

After truncation, the stack **Z** is reconstructed with the inverting transform with truncated coefficients to obtain the final HOSVD estimator ψ_{HOSVD} :

$$\psi_{HOSVD} = \mathbf{S}' \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \qquad (8)$$

Since the DWI datasets are three dimensions, the above methods should be extend to four-order HOSVD transform of the stack with 3D similarity patches. Besides this, the threshold should be modified as $\tau = \sigma\sqrt{2\log(p^3 K)}$ for a stack of size $\text{n} \times n \times n \times K$.

In Manjon et al (2010), mean consistency correction was followed the estimator to ensure coherence with the physical model of acquisition. This was implemented on the fidelity term:

$$\mathbf{Y}(i) - \frac{1}{L}\sum_{i=1}^{L} \hat{\mathbf{X}}(i) = 0, \ \forall p \in \mathbf{Y} \qquad (9)$$

This is done for all the location *p* in the LR dataset and imposes sub sampling consistency to the reconstructed patches. At last, the iteration process can be summed up with equation 8 and 9, and is applied until convergence:

$$\mathbf{x}^{t+1}(i) = \psi_{HOSVD}(\mathbf{x}^t(i)) \qquad (10)$$

$$\mathbf{x}^{t+1} = \mathbf{x}^{t+1} - NN(DH\mathbf{x}^{t+1} - \mathbf{y}) \qquad (11)$$

where NN is the nearest neighbor interpolation, and t is the iteration number.

To improve further the SR performance, the proposed HOSVD SR method can be augmented using joint information from adjacent directions of DWI dataset [15]. For each patch $\mathbf{P}_i$, the corresponding stack $\mathbf{Z}$ was constructed with K similar patches and the choice of these K similar patches was implemented as follow: the distance threshold selected all patches that $\left\|\mathbf{P}_i - \mathbf{P}_n\right\| < \tau_d$ was chosen to $\tau_d = 3\sigma^2 n^2$ where σ is the variance of the noise. This threshold balanced between the estimation accuracy and the computational speed as indicated in [14]. The joint information was introduced through enlarging the searching window into the M adjacent DWI datasets, where M was defined as $M = 2m + 1$ and m denotes the m directions before and after it. In this paper, the HOSVD super-resolution method using joint information with multiply directions is referred to as HOSVD-M.

3 Experiments

To evaluate the quality of reconstruction, B-spline interpolation, which have been introduced for DWI resolution enhancement in literature [18], are used for comparison. Besides this, non-local approach for image SR [9] as an effective non-local patch-based SR method is also involved for comparison. In this section, both synthetic and in vivo dataset were implemented for evaluation.

The simulation dataset consists of the 3D structure field presented at the 2012 HARDI Reconstruction Challenge [19] and has 16×16×5 volume attempting to simulate a realistic 3-D configuration of tracts occurring. As shown in Fig. 1a, this dataset is comprised of five different fiber bundles, which gave rise to the nonplanar configurations of bending, crossing, and kissing tracts. All fiber tracts were characterized with a fractional anisotropy between 0.75 and 0.90. To better explore the proposed method, this synthetic dataset was also corrupted by Rician noise (SNR = 30) and demonstrated in Fig. 1e. Both the original dataset and the noisy one were down-sampled by a factor 2 using the nearest neighbor interpolation along each axis. Next, the LR datasets were super resolved using the B-spline method, the non-local method, and the proposed method, respectively. In addition to the visual comparison demonstrated in Fig. 1, the angular accuracy was also involved for quantitative evaluation [19]. The angular accuracy in the orientation of the estimated fiber compartments was assessed by means of the average error (in degree) between the estimated fiber directions and the true ones present in a voxel:

$$\overline{\theta} = \frac{180}{\pi}\arccos(|\,\mathbf{d}_{true}\cdot\mathbf{d}_{estimated}\,|) \qquad (12)$$

where the unitary vector $\mathbf{d}_{true}$ and $\mathbf{d}_{estimated}$ are a true fiber population in the voxel and the closest of the estimated directions.

The in vivo DWI dataset was acquired using a 7T Philips Achieva whole body scanner (Philips Healthcare, Cleveland, OH) equipped with a volume head coil for transmission and 32-channels. A DW dual spin-echo, SENSE accelerated msh-EPI was used to acquire the DWI data (b-value: 700 s/mm^2; 15 diffusion directions); FOV = 210 × 30 × 21 mm^3; matrix size = 300 × 300 with 15 slices and a spatial resolution of 0.7 × 0.7 × 2 mm^3. In order to validate the proposed approach quantitatively and qualitatively, a gold standard image was constructed based on this in vivo HR DWI dataset. To do that, we averaged 10 acquisitions of high-resolution DW images in the image space (0.7 × 0.7 × 2 mm^3). Then the LR images used for the experiment were simulated by down-sampling our gold standard by a factor of 2 using the nearest neighbor interpolation along each axis (i.e., [2 2 2]), which resulted in simulated LR images of 1.4 × 1.4 × 4 mm^3..

Tensor estimation of the in vivo DWI dataset are evaluated quantitatively between the super-resolved dataset and the gold standard. First, the diffusion tensor field and principal eigenvector were computed using CAMINO [20] and are demonstrated in Fig. 1; Tab. 1 contains the mean and standard deviation (std) of the angular error estimated by equation (12). Fractional Anisotropy (FA) map and colormap of the estimated DTI are calculated for comparison. At last, the main direction of the tensor was also demonstrated for visualized comparison.

4 Results

Fig. 1 demonstrates the principle eigenvector of the tensor model in the synthetic phantom and reconstructed results using the B-spline interpolation, non-local upsampling, proposed HOSVD and proposed HOSVD-M. It can be observed that all the results of super-resolved methods were outperform the interpolated results dramatically. The proposed method achieved best results in both noisy and no-noise situations. This may probably due to the adaptive HOSVD bases derived from the stacked patches which is more suitable for the reconstruction.

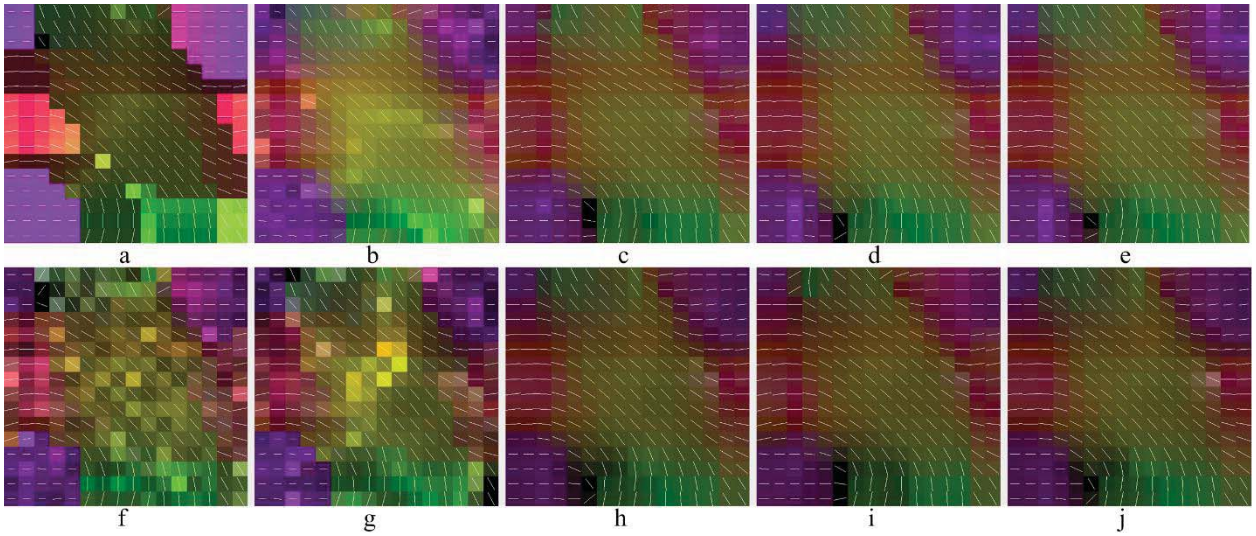


Fig. 1 Principle eigenvector of tensor model in the synthetic phantom (a-e); original dataset, reconstructed phantom datasets using B-spline, non-local upsampling, proposed NL-SVD, proposed HO-SVD; (f-j) noisy phantom (SNR=30), reconstructed phantom datasets using B-spline, non-local method, proposed HOSVD, proposed HOSVD-M

The reconstructed results of in vivo DWI data were demonstrated in Fig. 2 quantitatively and qualitatively. Fig. 2 shows the visual comparison of the reconstructed DWI images. Interpolated results (Fig. 2b) achieved blurriest results. The proposed method reconstructed the most similar results with the original images. The enlarged region (Fig.2j) demonstrated that the proposed HO-SVD reconstructed the clear structure of the crack area compared with the same area constructed by other methods were blur and hard to distinguish the edges.

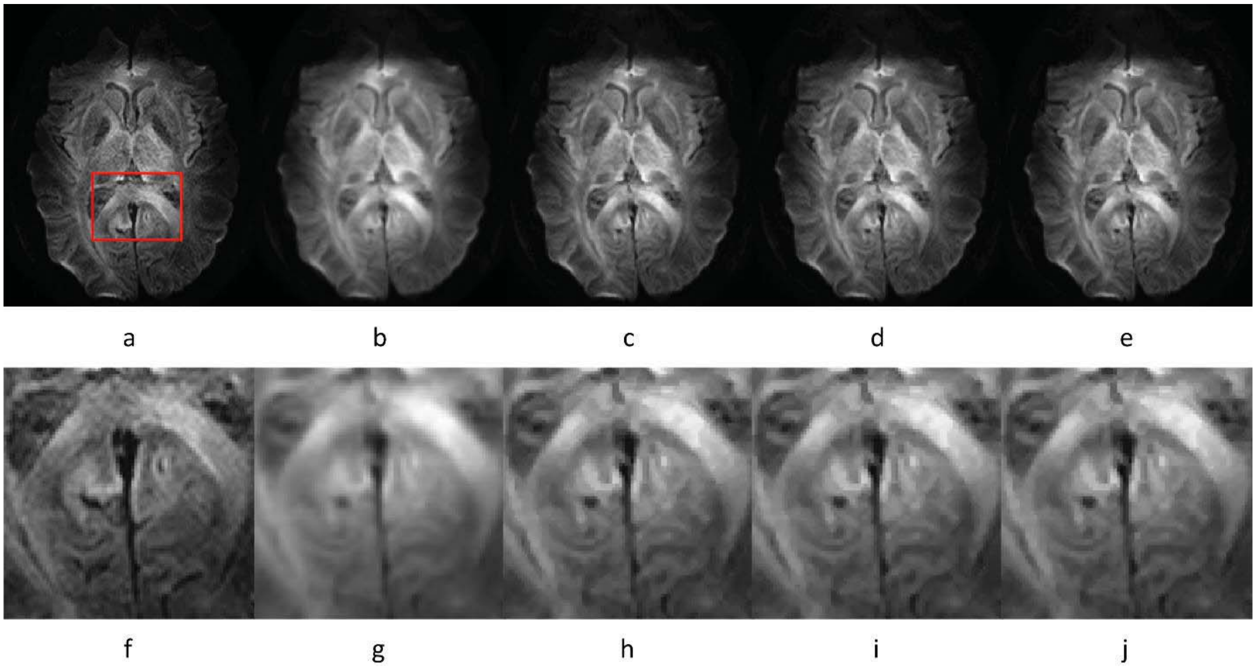


Fig. 2. Tests of diffusion weighted image reconstruction obtained for different methods. Top: (a) The gold standard. (b-e) Result of B-spline reconstruction, result of non-local method, proposed HOSVD, proposed HOSVD-M. Bottom: (f-j) the enlarged details of the gold standard, B-spline reconstruction, the non-local method, proposed HOSVD, proposed HOSVD-M.

Fig. 3 and Fig. 4 demonstrate the tensor estimation results using the super-resolved DWI datasets. Fig. 3 shows the FA map of the estimated DTI datasets. The proposed HOSVD-M method achieved the best results in the enlarged areas and remained most of the structure and tissues in the original images. The fiber direction indicated with the FA colormap was demonstrated in Fig. 4. It can be seen that, in Fig. 4e, the proposed HOSVD-M obtained the robust direction reconstruction results. For example, in the bundle of corpus callosum, the color of most of the voxels remained the same.

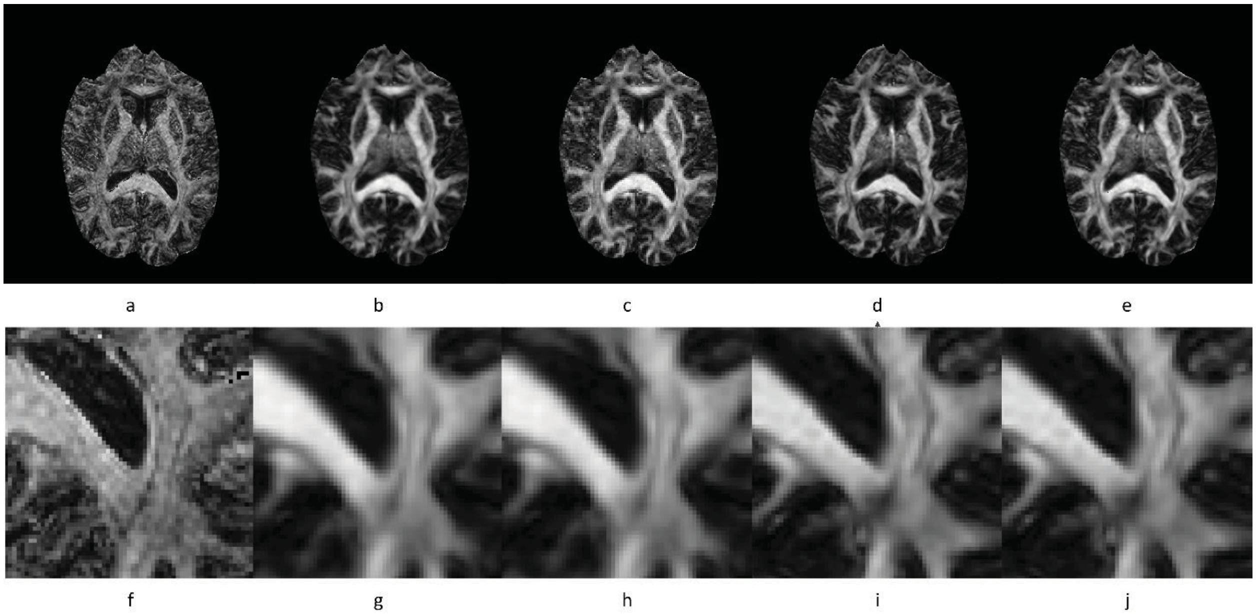


Fig. 3. FA maps estimated on the gold standard and several methods. (a) FA maps estimated on the gold standard; (b-e) FA maps obtained on the reconstructed dataset using B-spline, the non-local method, proposed HOSVD, proposed HOSVD-M. (f-j) The enlarged details of the B-spline reconstruction, the non-local method, proposed HOSVD, proposed HOSVD-M. The red ROIs indicate the detailed reconstruction. Visually, the FA map obtained using the proposed method is closer to the FA of the gold standard.

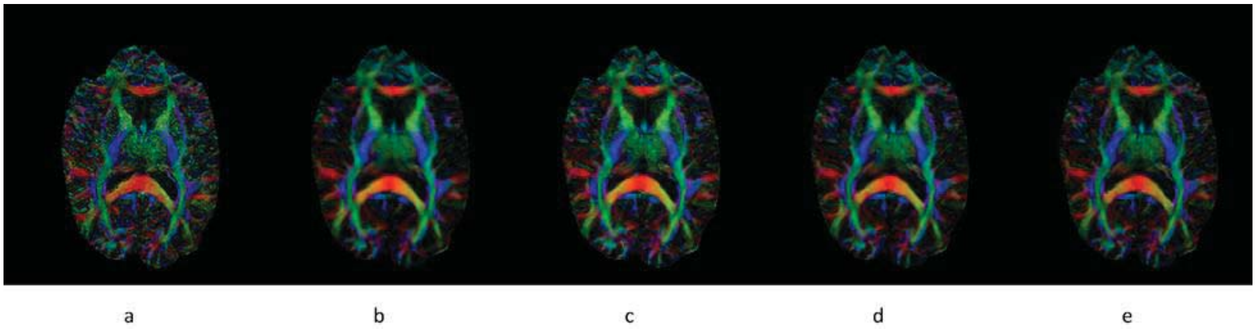


Fig. 4 (a) FA colormap on the gold standard; (b-e) FA colormap on reconstructed dataset using B-spline, the non-local method , proposed HOSVD, proposed HOSVD-M

5 Conclusions

In this paper, we proposed a patch-base single image super-resolution method which involved high order SVD for DWI dataset. The adaptive HOSVD bases learned from image ensured a more accurate image reconstruction and manipulation on similar patches stack leaded to a reduction on computational complexity. Quantitative and qualitative comparisons of the traditional interpolation method and non-local patch-based method demonstrate the competitive results in both DWI reconstruction and DTI estimation.

Compared with the currently used interpolation method and patch-based SR method, the improvement of the proposed HOSVD based method can be contributed to two features. The first one is the adapted HOSVD bases learned from a stack of similar patches. This method obtained the bases adaptively according to the image content and achieved more effective reconstruction results. The second feather is the introduction of the joint information from the adjacent direction of the DWI datasets. As pointed out in (Tristán-Vega and Aja-Fernández, 2010)[16], the adjacent directions contained a great many of image redundancy, the encapsulation of both processed direction and its adjacent ones benefits the reconstruction effectively.

Computational complexity is another important issue for pattch-based method as well as DWI processing. All experiments were performed on the Windows 7 computer equipment with an Intel(R) core i7-4600U AND 8g RAM, MATLAB R2013b. For a commonly used DWI datasets with 128×128 matrix size, 60 slices and 32 directions, the running time for single direction of non-local upsampling, proposed HOSVD and the proposed HOSVD-M were around 8 minutes, 3 minutes and 5 minutes respectively. This speed-up is probably due to the inherent dimension decreasing property of the SVD approaches and is specifically useful to HOSVD-M. method which introduce many times of extra computational burden.

6 Acknowledgements

This work is supported by National Basic Research Program of China (973 Program, No. 2014CB360506), National Science Foundation of China (No. 61303126), Ministry of Education, China: Chunhui Project (No. Z2015108), Sichuan Province Science and Technology Support Program (No. 2016RZ0051) ,the Open Research Fund from Province Key Laboratory of Xihua University (No. szjj2013-022), Sichuan Province Science and Technology Support Program (No.2015RZ0008) and Educational Commission of Sichuan Province of China (NO. 15ZA0186).

7 References

[1] Heidi Johansen-Berg and Timothy Behrens. Diffusion MRI, second edition, 2013, Academic Press.

[2] Pareyson D, Fancellu R C, Romano S, et al. Adult-onset Alexander disease: a series of eleven unrelated cases with review of the literature.[J]. Brain A Journal of Neurology, 2008, 131(Pt 9):2321-2331.

[3]Alexander D C, Pierpaoli C , Basser P J, et al. Spatial transformations of diffusion tensor magnetic resonance images[J]. IEEE Transactions on Medical Imaging, 2001, 20(11):1131-9.

[4] Turkal M ,, Tan E ,, Uzgur R ,, et al. Incidence and distribution of pulp stones found in radiographic dental examination of adult Turkish dental patients.[J]. Annals of Medical & Health Sciences Research, 2013, 3(3):572-6.

[5] Béné J, Saïd W, Rannou M, et al. Rectal bleeding and hemostatic disorders induced by dabigatran etexilate in 2 elderly patients.[J]. Annals of Pharmacotherapy, 2012, 46(6):e14.

[6]Miller K L, Stagg C J, Gwenalle D, et al. Diffusion imaging of whole, postmortem human brains on a clinical MRI scanner.[J]. Neuroimage, 2011, 57(1):167-81.

[7]Peled S, Yeshurun Y . Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging.[J]. Magnetic Resonance in Medicine, 2001, 45(1):29-35.

[8]Scherrer B, Gholipour A, Warfield S K. Super-resolution reconstruction to increase the spatial resolution of diffusion weighted images from orthogonal anisotropic acquisitions [J]. Medical Image Analysis, 2012, 16(7):1465-1476.

[9]Manjon J V, Coupe P A, Fonov V, et al. Non-local MRI upsampling.[J]. Medical Image Analysis, 2010, 14(6):784-792.

[10]Rousseau F, Kim K, Studholme C. A groupwise super-resolution approach: application to brain MRI[J]. Proceedings, 2010, 58(10):860-863.

[11] Turkal M , Tan E , Uzgur R , et al. Incidence and distribution of pulp stones found in radiographic dental examination of adult Turkish dental patients.[J]. Annals of Medical & Health Sciences Research, 2013, 3(3):572-6.

[12] Dinh Hoan Trinh, Marie Luong, Jean-Marie Rocchisani, et al. An Optimal Weight Method for CT Image Denoising[J]. Journal of Electronic Science & Technology of China, 2012, 10(2):124-129.

[13] Letexier D. and Bourennane S., “Adaptive Flattening for Multidimensional Image Restoration,” IEEE Signal Processing Letters, vol. 15, pp. 229-232, 2008.

[14]Ajit R, Anand R, Arunava B. Image denoising using the higher order singular value decomposition.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(4):849-862.

[15]Tristán-Vega A. and Aja-Fernández S. DWI filtering using joint information for DTI and HARDI. Medical Image Analysis, 14(2), 205-218 (2010).

[16]Makni S, Idier J, Vincent T, et al. A fully Bayesian approach to the parcel-based detection
estimation of brain activity in fMRI[J]. Neuroimage, 2008, 41(3):941–969.

[17] Lathauwer L D, Moor B D, Vandewalle J. 38–The application of higher order singular value decomposition to independent component analysis [J]. Svd & Signal Processing III, 1995:383-390.

[18] David R, J-Donald T, Stephen R, et al. Apparent Fibre Density: a novel measure for the analysis of diffusion-weighted magnetic resonance images [J]. Neuroimage, 2012, 59(4):3976-3994.

[19]Daducci A., E.J. Canales-Rodriguez, M. Descoteaux, E. Garyfallidis, Y. Gur, Y. C. Lin and J.P. Thiran, “Quantitative comparison of reconstruction methods for intra-voxel fiber recovery from diffusion MRI”. IEEE Trans. on Med. Imaging, 33(2), 384-399 (2014).

[20] Cook P. A., Bai Y., S. Nedjati-Gilani, K. K. Seunarine, M. G. Hall, G. J. Parker, D. C. Alexander, Camino: Open-Source Diffusion-MRI Reconstruction and Processing, 14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine, Seattle, WA, USA, p. 2759, May 2006.

Exponential Interpolation Technique for Scanning Electron Microscope Signal-to-Noise Ratio Estimation.

Z.X.Yeap1, K.S.Sim[1]
[1]Faculty of Engineering and Technology, Multimedia University, Ayer Keroh, Melacca, Malaysia

Abstract - *This paper introduces a new technique to estimate the SNR value of the focused SEM images from a defocus images. Basically SEM user took hours to make adjustment on the SEM and produced a focused image. Therefore, in order to solve the time consuming problem, a solution is proposed. Based on the experiments on 100 images, a method used an exponential equation is developed to estimate the noise-free zero offset point of a focused image by using a defocused image. This method uses less than a minute to process and it can overcome the time consuming problem while taking image repeatedly to get the most focus one.*

Keywords: SNR estimation, focus, defocus, SEM

1 Introduction

Scanning Electron Microscopy (SEM) is a device to capture high-resolution imaging of surfaces. SEM is particularly used for understanding of nano materials based on topographic and composition elements [1]. It is widely used in science, such as metallurgy, geology, biology and IC failure analysis. A high resolution, noiseless and focused SEM images are helpful in the analysis process. However, it is time consuming to capture such high quality images [2]. Besides, filtering process might make the image blur [3]. Therefore, a method to estimate the original noise free image is needed. Basically, signal-to-noise ratio (SNR) is used to quantify the image quality.

In this paper, a method to estimate the SNR of the SEM image using an out of focus image is proposed. This method is based on the focus detection which proposed by Ong in 1997 and single image SNR estimation technique which proposed by Sim in 2002 [3][4][5]. This method works by performing the Fast Fourier transform (FFT) of the image, obtains the autocorrelation curve (ACF) and determines the relationship of focus points and the center peak of the FFT. From the peak estimation, the SNR can be calculated by using the single image approach.

2 Problem Formulation

SNR is an important parameter to characterize the quality of images taken by using SEM. However, it is a difficult process to get a high resolution and focus SEM image. Manual focusing is very time-consuming even for the experienced SEM operator especially in low-dose and high-resolution task. Although there is an auto focus feature in the SEM itself, it does not perform well in the samples which containing highly directional features.

In order to estimate the SNR of the SEM images, Frank and L.Al Ali developed two images SNR estimation based on the cross correlation function (CCF) of two images acquisition of the same object [6][7][8]. The equation is shown in Eq.1.

$$\rho_{12} = \frac{r_{12}(0,0) - \mu_1\mu_2}{\sigma_1\sigma_2} \tag{1}$$

where, $r_{12}(0,0)$ is peak of the CCF function, μ is the mean and σ is the variance of the respective two aligned images. Then equation of SNR is shown in Eq.2.

$$SNR = \frac{\rho_{12}}{1-\rho_{12}} \tag{2}$$

However, this method needs to capture the images of the same sample. The perfect alignment of the two images becomes critical issue. Besides, this method is not applicable on the existed images stored in the data base. In 2002, Sim proposed a single image SNR estimation technique [5]. In Sim's single image approach, he assumed that the two images were identical. Therefore from Eq.1, the mean and variance of two images are the same which is,

$$\mu_1 = \mu_2 \tag{3}$$

$$\sigma_1 = \sigma_2 \tag{4}$$

Eq.2 can be simplified into Eq.5

$$SNR = \frac{\rho_{12}}{1-\rho_{12}} = \frac{r_{11}(0,0) - \mu^2}{r_{11}(0,0) - \bar{r}_{11}(0,0)} \tag{5}$$

where, $r_{11}(0,0) - \mu^2$ is signal component and $r_{11}(0,0) - \bar{r}_{11}(0,0)$ is noise component. Fig.1 represents the equation in graphical way.

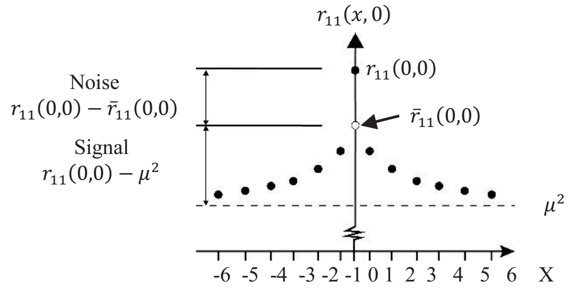


Figure 1: ACF of image with white noise [9]

Since noise-free zero offset point $\bar{r}_{11}(0,0)$ is an unknown in this single image approach, an estimation technique to estimate the point is needed [10].

3 Estimation of peak point in ACF curve from defocus image

In 1979, Tee proposed auto-focus method by determining the point of the best focus from the derivative or gradient of the signal [11]. This method has a major disadvantage. It is sensitive to noise as the method is based on the differentiation of the signal. Therefore in 1982, Erasmus and Smith proposed another method which was based on the power spectrum [12].

The power spectrum of an image and the Fourier transform of the covariance function (CF) of the same image are the same [5]. So the CF can be used directly in the method where the CF contains also the focus information. By observing the peak center point on the CF without concerning the shape, the focusing correction can be done. However, this method requires certain number of points of image variance versus the focus currents to maintain the accuracy of the algorithm. If the number of points are too less, the accuracy will decrease. However, it is time consuming to take more points in order to increase the accuracy of the algorithm. Besides, this method requires three curves. So more time is needed to compare to other algorithm using only one curve. Furthermore, this method is proved that it has a best result when using 3,000 x magnifications. But, in order to get a high-resolution SEM image, Ong suggested to have magnification more than 3,000x [4].

In 1996, Postek verified that FFT can be used in detecting the defocus of images [13]. Later in 1997, Ong proposed an algorithm which can detect the focus or defocus of the images by observing the change in the FFT while changing the focus [4]. The algorithm changes the FFT of an image into the range of 0 to 255 for display purpose so that important information does not lost while display. Then the intensity of every pixel is transformed using Eq.6.

$$s = c * log(1 + |r|) \qquad (6)$$

where c is a scaling constant and r is the intensity of each pixel.

Fig.2 shows the FFT displayed and its respective image. Top left image is a focused SEM image of IC and top right image is its processed FFT image. In the other hand, the bottom left image is a defocused SEM image of IC and bottom right image is its processed FFT image. A focus image has large amount of white dots and defocus image has lesser white dots.

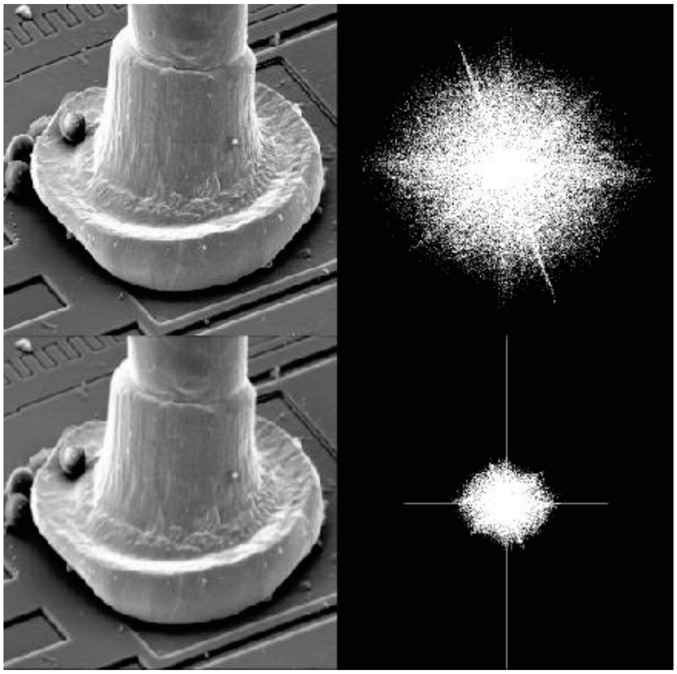


Figure 2: Comparison of focus and defocus image and its FFT.

However, this method might not perform well while the images are too much out of focus as the information contains in FFT is not enough. Even though Ong has proposed a solution which is to repeat the algorithm from low magnification until the desired magnification is reached, this solution is time consuming [2][5].

Based on the theory proposed by Ong, this paper proposes a method which can determine the relationship of focus points and SNR of the images and create an equation to estimate the noise-free zero off set point of the noise-free focused original image. From the observation, we can see that the ACF curve of focus image has sharper trend line on the center of ACF. Fig.3 shows the comparison of focus and defocus image in term of ACF curve.

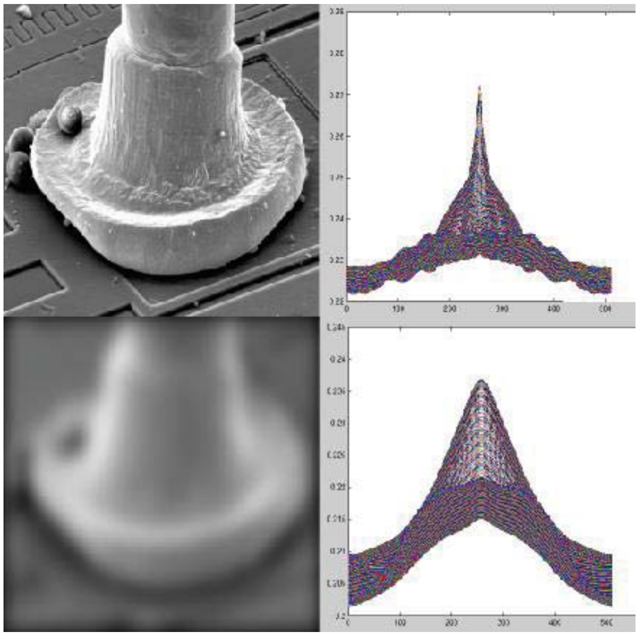


Figure 3: Comparison of ACF curve of focus and defocus SEM image. Top left: focus SEM image, top right: ACF curve of focus image, bottom left: defocus SEM image, bottom right: ACF of defocus SEM image.

There are two versions of the proposed method. These methods are developed based on 100 images with different level of focus sets. One set of image contains of many images of the same specimen area with different focus level.

First version is done by observing the relationship between the focus points and the difference between noisy and noise-free zero offset point. The relationship information is then plotted on a graph. The first method give poor accuracy due to the non-standardize number of focus points. Eq.7 shows the equation which best fitted the graph of focus points and the difference between noisy and noise-free zero offset point.

$$d = 0.05 \times 10^{-1.391f} - 0.03203 \tag{7}$$

where f is the non-normalized focus point of the image. *d* is the difference between focus noise-free zero offset point and defocus noise-free zero offset peak point $\bar{r}_{11(focus)}(0{,}0) - \bar{r}_{11(defocus)}(0{,}0)$

From the analyzation and observation of the first version, the error percentage is high for some images. It is because the number of focus points of the most focus image in a set is different due to the nature of image. Therefore, we normalized the focus points by dividing all the number of focus points of images with the largest number of focus points among set. After the normalization, the most focus image among the set will have focus point of "1".

From the data collected using 100 images with different level of focus, the relationship between the normalized focus points and the difference between noisy and noise-free zero offset points are plotted on graph. From the graph plotted, an equation which best fitted the line on the graph is shown in Eq.8.

$$d = 0.01748 \times 10\text{^}(-10.06f_n) \tag{8}$$

where f_n is the normalized focus point of the image. *d* is the difference between focus noise-free zero offset point and defocus noise-free zero offset peak point $\bar{r}_{11(focus)}(0{,}0) - \bar{r}_{11(defocus)}(0{,}0)$

f_n is calculated the FFT of that particular image. The image is first resized into 512x512 for the ease of the process. Then it will be converted into double format and performs the FFT. From the FFT, the image data is normalized into range of 1 to 255. Then the threshold is set to 0.5 according to Ong and the data is converted into white as 1 and black as 0 [4]. The focus points are the white dots inside the 262144 points. In order to normalize the focus points of different images into the same standard, the focus points are divided by 262144 to get the normalized f_n. 262144 is the total number of pixels in the image with size of 512x512. Next, a graph of the difference of peak value of focused and defocused image versus normalized f_n is plotted. Estimation trend line is plotted on the graph and the equation to estimate the noise-free zero offset point of the focused image is defined in Eq.7.

Eq.7 is applied on a set of SEM images with different level of focus and gets the *d* value respectively. The difference *d* is added into the defocus noise-free zero offset point respectively and average is calculated to get the focus noise-free zero offset point. SNR can be calculated using Eq.5.

4 Results and discussion

4 sets of images are chosen to test this proposed SNR estimation technique. The 4 selected images are shown in Fig.4.

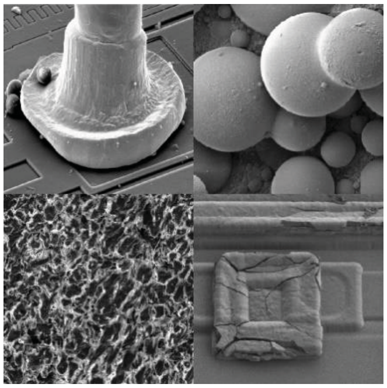


Figure 4: 4 selected images used to test the estimation method. Top left: set 1, top right: set 2, bottom left: set 3, bottom right: set 4.

Tables 1 to 4 show the estimated d and estimated zero-offset noise free using 3 sets of images. Eq.9 is used to calculate the percentage error by comparing to no.0 image (focused image).

$$percentage\ error = \frac{|estimated\ peak - real\ peak|}{real\ peak} \times 100 \qquad (9)$$

Table 1: Results of estimated d using set 1 images. 0 to 8 indicates image no.1 to image no.8. Labeling Q indicates the sequence number of the images.

Q	Peak	Focus points	f_n	Estimated d	Estimated peak
0	0.2724	60549	-	-	-
1	0.2727	54396	0.2075	0.002167	0.274867
2	0.2718	53852	0.2054	0.002213	0.274013
3	0.2718	53844	0.2054	0.002214	0.274014
4	0.2717	53124	0.2027	0.002276	0.273976
5	0.2715	50860	0.1940	0.002482	0.273982
6	0.2701	35994	0.1373	0.004392	0.274492
7	0.2686	25294	0.0965	0.006622	0.275222
8	0.2672	18670	0.0712	0.008538	0.275738
Average estimated peak point= 0.274538					

By comparing the average of estimated peak point and the peak point of image no.0 (focused image) peak point, the difference of the estimated peak and real zero-offset noise free point is equal to$0.274538 - 0.2724 = 0.002138$. Percentage difference is $\left(\frac{0.002829}{0.2724}\right) * 100 \approx 0.78\%$

Table 2: Results of estimated d using set 2 images. 0 to 8 indicates image no.1 to image no.8. Labeling Q indicates the sequence number of the images.

Q	Peak	Focus points	f_n	Estimated d	Estimated peak
0	0.2212	37837	-	-	-
1	0.2194	20763	0.0792	0.007879	0.227279
2	0.2194	21175	0.0808	0.007756	0.227156
3	0.2194	21181	0.0808	0.007754	0.227154
4	0.2191	19397	0.0740	0.008304	0.227404
5	0.2185	15749	0.0601	0.009551	0.228051
6	0.2182	14317	0.0546	0.010091	0.228291
7	0.2175	11998	0.0458	0.01103	0.22853
8	0.2171	11152	0.0425	0.011394	0.228494
Average estimated peak point= 0.227795					

By comparing the average of estimated peak point and the peak point of no.0 (focused image) peak point, the difference of the estimated peak and real zero-offset noise free point is equal to $0.227795 - 0.2212 = 0.006595$. Percentage difference is $\left(\frac{0.006595}{0.2212}\right) * 100 \approx 2.98\%$

By applying Eq.8, the percentage error of set 3 is: $\frac{|0.160887 - 0.1708|}{0.1708} \times 100 \approx 5.8\%$

From the results shown in Table 1 to Table 3, the error percentage is less than 10% which is acceptable result.

By applying Eq.8, the percentage error of set 4 is: $\frac{|0.17404 - 0.1660|}{0.1660} \times 100 \approx 4.83\%$

From the results shown in Table 1 to Table 3, the error percentage is less than 10% which is acceptable results.

By observing the results of Table 1 to Table 4, the estimation method is applicable in both high focus images and low focus images. From Table 2 and Table 4, a focus image has 114083 white points and a defocus image has only 29473 white points, the percentage error is less than 10% and this means that this method can estimate the zero offset point accurately.

Using the results get from 4 sets of images, the SNR of the focus image can be estimated. Table 5 shows the comparison of real SNR value and estimated SNR value. In this test, noise variance is set to 0.005.

Table 3: Results of estimated *d* using set 3 images. 0 to 8 indicates image no.1 to image no.8. Labeling Q indicates the sequence number of the images.

Q	Peak	Focus points	f_n	Estimated d	Estimated peak
0	0.1708	114083	-	-	-
1	0.1671	70895	0.2704	0.003257	0.165657
2	0.1624	43781	0.1670	0.005943	0.163543
3	0.1576	28111	0.1072	0.008260	0.16166
4	0.1534	19534	0.0745	0.009969	0.159669
5	0.1497	14634	0.0558	0.011246	0.157846
6	0.1466	11493	0.0438	0.012148	0.156048
7	0.1439	9483	0.0362	0.012822	0.154422
8	0.1416	8076	0.0308	0.001151	0.168251
Average estimated peak point= 0.160887					

Table 4: Results of estimated d using set 4 images. 0 to 8 indicates image no.1 to image no.8. Labeling Q indicates the sequence number of the images.

Q	Peak	Focus points	f_n	Estimated d	Estimated peak
0	0.1660	29473	-	-	-
1	0.1643	16811	0.0641	0.00917	0.17347
2	0.1637	14109	0.0538	0.010172	0.173872
3	0.1627	10229	0.0390	0.011805	0.174505
4	0.1621	8431	0.0322	0.012648	0.174748
5	0.1611	6607	0.0252	0.013565	0.174665
6	0.1596	4785	0.0183	0.014548	0.174148
7	0.1582	3504	0.0134	0.015281	0.173481
8	0.1577	3092	0.0118	0.015524	0.173224
Average estimated peak point= 0.174014					

Table 5: Comparison of real SNR and estimated SNR.

Image	Real SNR	Real SNR in dB	Estimated SNR	Estimated SNR in dB
set 1	68.5362	36.7184	102.9800	40.2500
set 2	84.8782	38.5759	46.1533	33.2841
set 3	32.7706	30.3097	9.9339	19.9424
set 4	29.0340	29.2581	48.1361	33.6494

From the result in Table 5, the estimated SNR in dB are similar to the real SNR in dB. The estimated SNR of set 3 has bigger difference compare to real SNR due to the noise existed in the image is more than other images. Table 6 shows the comparison results of two versions of the proposed method.

Table 6: Comparison of results of using normalized focus points and non-normalized focus points

Image	Estimated SNR without normalize d	Estimated SNR without normalize d in dB	Estimated SNR with normalize d	Estimated SNR with normalize d in dB
set 1	2.8045	8.9572	102.9800	40.2500
set 2	1.7099	4.6596	46.1533	33.2841
set 3	0.15364	-16.2697	9.9339	19.9424
set 4	0.0693	-23.1806	48.1361	33.6494

From the result in Table 6, we can see the application using normalized focus points give a better result compare to the version which use the non-normalized focus points. Table 7 shows the percentage error using the non-normalized focus points to predict the zero offset noise-free point.

Table 7: Percentage error of using the non-normalized focus points.

Image	Real zero offset noise-free point	Estimated using non-normalized	Percentage error (%)
Set 1	0.2724	0.3005	10.34%
Set 2	0.2212	0.1932	16.39%
Set 3	0.1708	0.1208	29.30%
Set 4	0.1660	0.1865	15.67%

From the result of Table 7 and the calculation in Table 1 to Table 4, the percentage error reduce for average of 14% overall.

5 Conclusion

In conclusion, this proposed method can estimate the focus SEM images' SNR using defocus image. Besides, this method only used up to few seconds to process. It saves more time as compared to time taken for repeating adjusting on SEM machine to capture a focused image. From the results and discussion part, the normalization part plays an important role to make the solution more accurate. This method can estimate the noise-free zero offset peak point more accurately if using more images with different focus level as the final result is from the average result of images with different focus level.

6 References

[1] K.S. Sim, M.A. Kiani, M.E.Nia and C.P. Tso, "Signal-to-noise ratio enhancement on SEM images using a cubic spline interpolation with Savitzky–Golay filters and weighted least squares error," in *Journal or Microscopy*, 2014, vol.00(0), pp1-11.

[2] E. Oho and K. Suzuki, "Highly accurate SNR measurement using the covariance of two SEM images with the identical view," in *Scanning*, vol.34, pp.43-50, Feb 2012.

[3] K.S. Sim, V. Teh and M.E.Nia. Adaptive noise "Wiener filter for scanning electron microscope imaging system" in *Scanning* vol.38, pp.148-163, 2015.

[4] K.H. Ong, J.C.H.Phang and J.T.L.Thong, "A Robust Focusing and Astigmatism Correction Method for the Scanning Electron Microscope," in Scanning, 1997, vol.19, pp 553-563.

[5] K.S.Sim, "Signal-to-noise ratio estimation in scanning electron microscope imaging system,"M.S.thesis, National University if Singapore, Singapore, (2002).

[6] J. Frank, "Three dimensional electron microscopy of macromolecular assemblies," San Diago: Academic.

[7] J.Frank and L.Al-Ali, "Signal-to-noise ratio of electron micrograph obtained by cross correlation" in Nature. Vol.256, pp376-379, 1975.

[8] J.Frank, "The role if correlation technique in computer image processing in computer processing of electron microscope images," Berlin: Springer-Verlag, pp.214-215, 1980.

[9] K.S.Sim, M.Y.Wee and W.K.Lim. "Image signal-to-noise ratio estimation using shape-preserving piecewise cubic Hermite autoregressive moving average model," in Journal of Microscopy Research and Technique, vol. 71(10), pp.710-720, 2008.

[10] V. Teh & K.S. Sim, "Image signal-to-noise ratio estimation using adaptive slope nearest-neighborhood model." in *Journal of Microscopy* vol.260, pp.352-362, 2015.

[11] W.J. Tee, K.C.A. Smith and D.M.Holburn, "Automatic focusing and stigmatizing system for the SEM," in Journal of Physics E: Scientific Instrument, vol.12, pp.35-38, 1979.

[12] S. J. Erasmus and K. C. A. Smith, "An automatic focusing and astigmatism correction system for the SEM and CTEM," in *Journal of Microscopy*, vol.127, pp.185-199

[13] M.T. Postek and A.E.Vladar, "SEM performance evaluation using the sharpness criterion," in Standards and calibration method for critical dimension metrology, Santa Clara, CA, 1996, vol.2725, pp.169-172.

SESSION

FEATURE EXTRACTION, CLASSIFICATION, AND SEGMENTATION METHODS

Chair(s)

TBA

On Selecting the Best Unsupervised Evaluation Techniques for Image Segmentation

Trung H. Duong
Center for Advanced Infrastructure and Transportation
Rutgers the State University of New Jersey
New Brunswick, NJ, US
trung.duong@rutgers.edu

Lawrence L. Hoberock
Mechanical & Aerospace Engineering Dept.
Oklahoma State University
Stillwater, OK, US
larry.hoberock@okstate.edu

Abstract— **One fundamental difficulty with evaluation of segmentation is that there is no objective, clear definition of good or bad segmentation. Even worse, different observers often do not agree on how to segment the same image. In this paper, we present six unsupervised metrics in the literature that are commonly used to evaluate segmentation results. Then we propose a framework to find the best comparison metric in the sense that this metric is the most consistent with the ground-truth provided by manual segmentation and, at the same time, is the most sensitive to random segmentation results. We believe a "good" metric should produce a high score on the ground-truth segmentation, as well as produce a low score on random segmentation. We employ the best unsupervised metric to compare results from different image segmentation methods on the Berkeley Segmentation Dataset and Benchmark, which consists of 300 color images of natural scenes.**

Keywords— ***image segmentation, unsupervised evaluation, quantitative evaluation***

I. Introduction (*Heading 1*)

Image segmentation is a process in computer vision that partitions a digital image into multiple segments of non-overlapping regions. It can be viewed as the process of labeling all pixels of the input image such that pixels with the same label are connected and share certain visual properties. The result of this process is a set of non-overlapping segments whose union forms the entire input image. Segmentation is a first step to simplify and represent an input image in a form that is more meaningful and easier to analyze. Then, properties of objects resulting from the segmentation process can be determined (such as size, shape, color distribution) for purposes of recognition, classification, and forming higher knowledge. Therefore, segmentation serves as a fundamental step in extracting knowledge from the image, and can be widely applied in many fields, such as classification, object recognition, object tracking, content-based image retrieval, surveillance, and medical imaging, among others [1-4].

The main challenge in segmentation problems is how to quantitatively evaluate a given image segmentation method, of which there are many approaches [5-10].What constitutes good segmentation is a problem similar to what constitutes good clustering. Since there is no precise definition of "good" clustering results or segmentation results, it is difficult to compare two given segmentation techniques. Normally, results of a segmentation method are compared with manually segmented results by humans on a set of test images (ground truth segments). However, it is very time consuming and tedious to construct such a ground truth database. Even worse, different persons often provide significantly different segmentation results for the same image.

The reminder of this paper is organized as follows. Six unsupervised metrics in the literature that are commonly used to evaluate segmentation results are presented in the Section 2. Then, in the Section 3, we propose a framework to find the best comparison metric in the sense that this metric is the most consistent with the ground-truth provided by manual segmentation and, at the same time, is the most sensitive to random segmentation results. A "good" metric should produce a high score on the ground-truth segmentation, as well as produce a low score on random segmentation. Section 4 shows the comparison of different image segmentation methods using the best unsupervised metric, and Section 5 presents conclusions.

II. Unsupervised Metrics

A. *Discrepancy evaluation technique (D-metric)*

This technique, introduced by Weska and Rosenfeld [5], is a simple technique based on the discrepancy measure (D-metric) between the original and the segmented images. Precisely, discrepancy is computed by the sum of the squares of specified differences between the original image and the segmented image. This measure D is given by:

$$D = \sum_{i=1}^{Ih} \sum_{j=1}^{Iw} \left(G(i,j) - L(i,j)\right)^2 \qquad (2.1)$$

where I_h and I_w are, respectively, the height and width of the image in pixels; $G(i,j)$ and $L(i,j)$ are the grayscale of the pixel (i,j) of the input image and the segmented image, respectively. Note that in the segmented image, pixels of the same segment have the same value (grayscale or color) that is the average of pixel values (grayscale or color) that belongs to this segment in the original image. In other words, the D-metric is related to the total variation of grayscale in the original image corresponding to all segments. For a good segmentation result, the metric *D* should be close to zero.

B. The intra/inter-region visual error ($\boldsymbol{E}$ − metric)

An unsupervised evaluation technique based on the visible color difference [6] is employed to evaluate image segmentation algorithms. Define "intra-region visual error" E_{intra} as:

$$E_{intra} = \frac{1}{N}\sum_{k=1}^{n} \mu(e_k^2 - th) \tag{2.2}$$

where $\mu(a)$ is a step function, given by

$$\mu(a) = \begin{cases} 1 & a > 0 \\ 0 & otherwise \end{cases} \tag{2.3}$$

and e_k^2 is the square of the color error in the R_k image region computed in **L*a*b** color space ($1 \leq i \leq n$; where n is number of segments), given by

$$e_k^2 = \sum_{p \in R_k} \|\boldsymbol{p} - \overline{\boldsymbol{R}_k}\|^2 \tag{2.4}$$

where $\boldsymbol{p} = [l, a, b]^T$ is the 3D vector corresponding to the pixel $\boldsymbol{p}$; $\overline{\boldsymbol{R}_k} = [\bar{l}_k, \bar{a}_k, \bar{b}_k]^T$ is the mean vector of the region $\boldsymbol{R}_k$; $\|\ \|$ is the standard Euclidean norm; N is number of pixels in the input image; th denotes the threshold for visible color difference, with $th = 0.36$ according to [6].

The intra-region visual error is designed to measure the visible color difference within the segmented regions. This measure can be used to estimate the degree of under-segmentation. Intuitively, a properly segmented region should contain as few visible color errors as possible. In other words, the smaller the value of E_{intra}, the better is the segmentation.

On the other hand, another measurement named inter-region visual error is designed to measure the invisible color difference between every adjacent pair of segmented regions. This measure can be used to estimate the degree of over-segmentation. Define "inter-region visual error" E_{inter} as:

$$E_{inter} = \frac{1}{N}\sum_{k=1}^{n}\sum_{j=1, j\neq k}^{n} \frac{L_{kj}}{L_k L_j}\, \mu\left(th - \|\overline{\boldsymbol{R}_k} - \overline{\boldsymbol{R}_j}\|^2\right) \tag{2.5}$$

where L_k and L_j are the boundary length (in pixels) of regions $\boldsymbol{R}_k$ and $\boldsymbol{R}_j$, respectively. L_{kj} is the "joined length" (or number of pixels in the "shared" boundaries) between the image regions $\boldsymbol{R}_k$ and $\boldsymbol{R}_j$. $L_{ij} = 0$ if $\boldsymbol{R}_k$ and $\boldsymbol{R}_j$ are disjoined. Given a segmentation result, we take into account these boundary pixels with "invisible" color difference (no difference in color) across the boundary. Intuitively, these pixels should not be treated as boundaries. Hence, the smaller the value of $\mathrm{E}_{\mathrm{inter}}$, the better is the segmentation.

Based on these two measures, a score or metric E to measure how good is a given segmentation, may be defined by:

$$E = \frac{1}{2}(E_{intra} + E_{inter}) \tag{2.6}$$

Note that, for a segmented image, a large value of intra-region visual error means numerous pixels may be mistakenly merged, such that this image could have been under-segmented. On the other hand, a large value of inter-region visual error means numerous boundary pixels may be mistakenly generated, such that the image could have been over-segmented. Moreover, there is a reciprocal relationship between intra-region error and inter-region error. As we adjust the controlling parameters of a segmentation algorithm to merge more regions together, the inter-region error decreases, while the intra-region error increases. On the contrary, as we segment an image into more regions, the intra-region error decreases while the inter-region error increases. Also note that all pixel color values are normalized to range of [0,1]. Normally n (number of regions) $\ll N$ (number of image pixels), such that E_{intra}, E_{inter} and E will all lie in the range [0,1]. For a good segmentation result, the metric E should be close to zero.

C. Average squared color error ($\boldsymbol{Q}$ − metric)

This metric is empirically defined by Borsotti at el. [7] as:

$$Q = \frac{1}{kN}\sqrt{n}\sum_{k=1}^{n}\left[\frac{e_{R_k}^2}{1 + \log A_k} + \left(\frac{n_k}{A_k}\right)^2\right] \tag{2.7}$$

where $k = 10^4$ is an empirical number and a normalization factor that takes the size of the image into account, N is the total number of pixels in the image, n is the number of segments, e_{Rk}^2 is the square of the color error in the segment $\boldsymbol{R}_k$ as in (2.4), A_k is the area in square pixels of the segment $\boldsymbol{R}_k$, and n_k is the number of segments that have the area in the range from 0.98A_k to 1.02A_k.

Note that the $\sqrt{n}$ term penalizes segmentation results having too many regions; the $e_{R_k}^2$ term penalizes results having non-homogeneous regions. The square of the color error will be significantly higher for a large region, such that the adjusted term ($1 + \log A_k$) is applied. Experiments show that the number of large regions that have a similar area is small, while the number of small regions that have a similar area may be large [42]. Therefore, the Q measure also penalizes the segmentation result having too many small regions that are similar in size. For a good segmentation result, the metric Q should be close to zero.

D. Entropy based metric ($\boldsymbol{H}$ − metric)

Zhang et al. [8] proposed another unsupervised evaluation metric based on the "region" entropy and the "layout" entropy. Define the entropy for each segment $\boldsymbol{R}_k$ by:

$$H(\boldsymbol{R}_k) = -\sum_{m \in V_k} \frac{L_k(m)}{A_k} \log \frac{L_k(m)}{A_k} \tag{2.8}$$

where V_k is the set of all possible grayscale values of pixels in the region $\boldsymbol{R}_k$ of the original image, and $L_k(m)$ is the number of pixels in this region $\boldsymbol{R}_k$ of the original image that have the grayscale value of m.

Define the region entropy H_r of entire image as the sum of entropy across all regions weighted by their areas, given by:

$$H_r = \sqrt{n}\sum_{k=1}^{n}\frac{A_k}{N}\, H(\boldsymbol{R}_k) \tag{2.9}$$

where n, N, A_k, and $H(\boldsymbol{R}_k)$ are defined as before. Note that N is the total number of pixels of the image or the "area" of the image. The first term $\sqrt{n}$ penalizes segmentation result having too many regions. Now define the layout entropy H_l by:

$$H_l = -\sum_{k=1}^{n} \frac{A_k}{N} \log \frac{A_k}{N} \qquad (2.10)$$

The H-metric measuring the effectiveness of a segmentation method is the addition of the two entropies, given by:

$$H = H_r + H_l \qquad (2.11)$$

For a given dataset, the H-metric can be normalized to a range [0,1], in which a small value of H (close to zero) indicates good segmentation.

E. *Spatial color contrast along the boundaries of segments ($\boldsymbol{C}$ − metric)*

This metric introduced in [9] considers the internal and external contrast of the neighbors of each pixel in all segments. Define $W(\boldsymbol{p})$ as the set of pixels that are the 8 neighbors of the pixel $\boldsymbol{p}$. For each segment $\boldsymbol{R}_k$, define the internal contrast $\boldsymbol{I}_k$ and external contrast $\boldsymbol{E}_k$ as:

$$I_k = \frac{1}{A_k} \sum_{\boldsymbol{p} \in \boldsymbol{R}_k} \max(\|\boldsymbol{p} - \boldsymbol{q}\|, \forall \boldsymbol{q} \in W(\boldsymbol{p}) \cap \boldsymbol{R}_k) \qquad (2.12)$$

$$E_k = \frac{1}{l_k} \sum_{\boldsymbol{p} \in \boldsymbol{R}_k} \max\left(\|\boldsymbol{p} - \boldsymbol{q}\|, \forall \left(\boldsymbol{q} \in W(\boldsymbol{p})\right) AND \left(\boldsymbol{q} \notin \boldsymbol{R}_k\right)\right) \quad (2.13)$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the 3D vectors corresponding to the pixel $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively; $\| \quad \|$ is the standard Euclidean norm; and $\boldsymbol{A}_k$ and $\boldsymbol{l}_k$ are the area and the boundary length of the segment $\boldsymbol{R}_k$, respectively.

The contrast $C(\boldsymbol{R}_k)$ of the segment $\boldsymbol{R}_k$ is given by:

$$C(\boldsymbol{R}_k) = \begin{cases} I_k/E_k & if\ I_k \le E_k \\ E_k/I_k & otherwise \end{cases} \qquad (2.14)$$

The global contrast, which is used as the measure the effectiveness of segmentation, is defined by:

$$C = \frac{1}{N} \sum_{k=1}^{n} A_k C(\boldsymbol{R}_k) \qquad (2.15)$$

For a given dataset, the C-metric can be normalized to a range [0,1], in which a small value of C (close to zero) indicates good segmentation

F. *Global intra-region homogeneity and inter-region disparity ($\boldsymbol{DD}$ −metric)*

Rosenberger and Chehdi [10] proposed a metric for segmentation evaluation based on the global intra-region homogeneity and the global inter-region disparity of segments. This metric employs only grayscale levels of image pixels. The global intra-region homogeneity $\boldsymbol{D}_1$ of segments is the weighted average of the pixel intensity variation of all segments:

$$D_1 = \frac{1}{N} \sum_{k=1}^{n} A_k \sum_{\boldsymbol{p} \in \boldsymbol{R}_k} \left(G(\boldsymbol{p}) - \overline{G(\boldsymbol{R}_k)}\right)^2 \qquad (2.16)$$

where $G(\boldsymbol{p})$ is the grayscale level or intensity of the pixel $\boldsymbol{p}$; $\overline{G(\boldsymbol{R}_k)}$ is the average grayscale level of all pixels belong to the segment $\boldsymbol{R}_k$; and other notations are as defined earlier.

Define the disparity of two segments $\boldsymbol{R}_k$ and $\boldsymbol{R}_m$ as:

$$d(\boldsymbol{R}_k, \boldsymbol{R}_m) = \frac{\left|G(\boldsymbol{R}_k) - G(\boldsymbol{R}_m)\right|}{N_G} \qquad (2.17)$$

where $\boldsymbol{N}_G$ is the number of gray levels of all pixels in the entire image. Then, the global inter-region disparity $\boldsymbol{D}_2$ is defined as the average of all the disparity between any two segments, given by:

$$D_2 = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{m=1}^{n} d(\boldsymbol{R}_k, \boldsymbol{R}_m) \qquad (2.18)$$

Notations in (2.18) are as defined earlier. The metric $\boldsymbol{DD}$ used as a quantitative measure of a segmentation is:

$$DD = \frac{D_1 - D_2}{2} \qquad (2.19)$$

For a given dataset, the DD-metric can be normalized to a range [0,1], in which a small value of DD (close to zero) indicates good segmentation.

III. Selecting the Best Unsupervised Metric

The key idea of selecting the best evaluation metric out of six metrics introduced above is that a better metric should produce a better score on the ground-truth segmentation (produced by human observers) and produce a worse score on random segmentations at the same time. The dataset used in our test is the public Berkeley Segmentation Dataset and Benchmark (BSDB) [11]. It consists of 300 color images of natural scenes and hand-implemented segmentations for each image serving as ground-truth as shown in Fig.1.

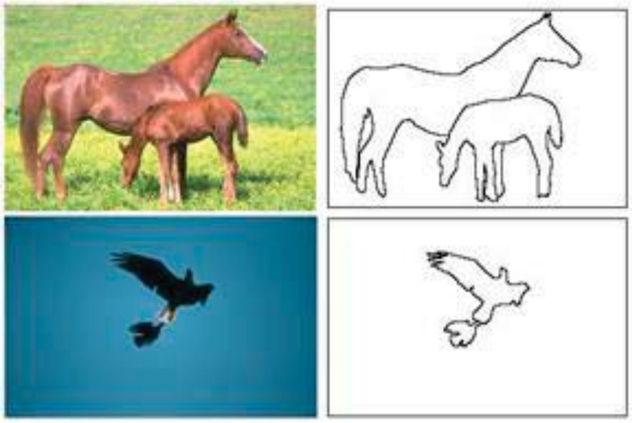


Fig. 1. Some pairs of original image (left) and ground truths (right)

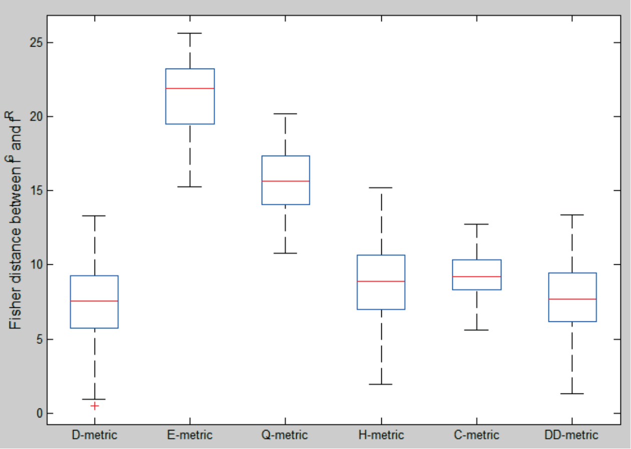
We generated 300 random segmentation results to help evaluate the various metrics. To create a random segmentation, we first generated a bitmap image having the same size as the image to be segmented (e.g. 480 pixels x320 pixels) and employed a random integer number n, $1 \le n \le 150$. Next, we randomly initialized n segment "centers" (n pairs of xy-coordinators for n points in the image plane). Then, we employed a K-means clustering technique to divide the bitmap image into n regions, which served as a random segmentation map consisted of random n segments with random size, shape, and location.

For the ith evaluation metric, $1 \le i \le 6$, we calculate its 300 metric values (normalized to be in the range from 0 to 1) for the 300 ground-truth segmentations. Call the distribution of these 300 values the f_i^G distribution, and compute its mean μ_i^G and standard deviation σ_i^G. Similarly, we calculated these metric values for 300 random segmentations to form the distribution f_i^R with μ_i^R mean and σ_i^R standard deviation. Since we expect for the "good" evaluation metric, the f^G distribution will be close to zero, and the f^R distribution will be close to one, the Fisher's distance [12], was used to measure the dissimilarity or the "distance" between two distributions. The Fisher distance F_i then become our quantitative measurement of the goodness of an evaluation metric, with F_i given by:

$$F_i = \frac{2(\mu_i^R - \mu_i^G)^2}{(\sigma_i^R)^2 + (\sigma_i^G)^2} \qquad (3.1)$$

The larger the value of the Fisher distance, the more separation there is between the two distributions (ground-truth and random segmentations), and therefore the better is the metric. Note that the distributions of metric values corresponding to ground-truth segmentations are fixed, while the distributions of metric values corresponding to random segmentations vary, due to a different set of 300 random segmentation is generated each time a metric evaluation is computed. Accordingly, we ran the Fisher distance procedure for each metric 50 times (each time with a different set of 300 random segmentations) to obtain reliable statistical measures. Table I shows a sample result of a single run, in which the E-metric provides the best separation between the two distributions.

Fig. 2. : Fisher's distances distribution corresponding to 6 metric measurements after 50 randomization runs



The boxplot of Fisher distance distributions corresponding to the 6 metrics after 50 runs is provided in Fig.2. In each box, the central mark (in red) is the median, the edges of the box (in blue) are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers, if present, are plotted individually as red crosses.

The larger the Fisher's distance, the better is the metric measurement. The results clearly show that E-metric is the best among the six metrics presented herein. Accordingly, it will be used in comparisons that follow.

IV. Comparison of Image Segmentation Methods Using the Best Metric E

In this section, the segmentation results from four segmentation algorithms using unsupervised color segmentation based on superpixels are compared, namely: PSEG [16], GSEG [17], JSEG [18], and DUHO [13]. The DUHO algorithm contains two main steps. First, a superpixel generating algorithm is applied to a given image to build *K* superpixels. Then a new region-growing algorithm iteratively groups these superpixels into appropriate regions and forms the final image segmentation result. This method is a type of unseeded region-based segmentation technique that preserves the spatial relationship between pixels in the image, and hence preserves the detailed edges and the image spatial structure. The DUHO algorithm has three main advantages compared with other region-growing-based segmentation techniques [14-17]. First, it operates at a "superpixel" level, rather than at the image pixel level, to reduce computational time and depress noise. Second, the proposed method works for color images rather than gray scale image as in [14-17]. Third, the decision of grouping an adjacent superpixel to an existing region is dynamically dependent upon the statistics, or "shape and size", of this region. The segmentation results show significant improvements when compared with results from existing methods using a fixed, global threshold [13,20].

The main idea of the PSEG [16] is to scan through a hierarchy of image partitions, from a highly over-segmentation to a highly under-segmentation partition, to find the best partition that maximizes a predefined goodness function. In PSEG, the pixel colors in RGB color space are used directly. The GSEG [17] is based on the unseeded region growing technique, in which the initial seeds are found using the color gradient information (in CIE L*a*b* color space). After the region growing process, regions with similar characteristics are blended by the multi-resolution region merging to form the final segmentation. During the merging process, new seeds might be added or old seeds discarded .The JSEG [18] includes 2 stages, quantization and spatial segmentation. First, pixel colors (in CIE L*u*v* color space), smoothness of the local area, and texture orientations are quantized into a small number of predefined values. Then, these values are formed into a representation vector of a local region that will be clustered into different groups.

The E-metric unsupervised evaluation was applied for 300 segmentation results on images from the DBSB dataset [11]. In addition, we employ supervised evaluation techniques, called the boundary recall and boundary precision [19] measurement to evaluate the segmentation results. Boundary recall is the percent of the ground-truth edge pixels that are within two pixels distance from a region boundary. We can express this as:

$$Boundary\ recall = \frac{L(\text{"hit" groundtruth edge})}{L(\text{groundtruth edge})} \qquad (3.2)$$

where L(.) is the length in pixels; "hit" means that the current pixel in the ground-truth edge is in the range of 2 pixels from a pixel in a region's boundary of the segmentation result. Boundary precision is the percent of the region edge pixels (resulted from a segmentation method) that are within two pixels distance from a ground-truth edge boundary. We can express boundary presision as:

$$Boundary\ precision = \frac{L(\text{"hit" segmentation edge})}{L(\text{segmentation edge})} \quad (3.3)$$

$L(\text{"hit" segmentation edge}) = L(\text{"hit" groundtruth edge})$ is the number of "mutual" pixels, or pixels within 2 pixels along the boundary of the ground-truth edge and the segmentation result edge. Boundary recall and boundary precision measure the degree of matching between the ground-truth and the segmentation results. A high recall value indicates that most of the "correct" boundaries are discovered in the segmentation results. A high precision value indicates that the segmentation results are more accurate, or most of the segmentation boundaries are the "correct" boundaries. For a good segmentation, both boundary recall and boundary precision values should be high (near 1). Table 3.2 summarizes the evaluation results.

Fig. 3 presents some segmentation results from the four methods. Visually, results from all four methods appear close to human segmentation. However, the DUHO produces finer details.

We see that the DUHO algorithm performs better than the JSEG, GSEG, and PSEG algorithms, based on both the unsupervised metric **E** and the supervised boundary recall measurement. The DUHO algorithm is the second best (and comparable with the best) among these four algorithms based on the boundary precision measurement. The values of boundary precision in Table 3.2 are not as high as desired because hand-label segmentation usually ignores details in the image, and hence produces coarser results compared with results from all four segmentation methods. Our DUHO algorithm performs significantly faster than the JSEG, GSEG, and PSEG algorithms.

TABLE I. EVALUATION OF FOUR SEGMENTATION ALGORITHMS ON THE DATASET

	Segmentation algorithm				
	Human hand-label	***PSEG [16]***	***GSEG [17]***	***JSEG [18]***	***DUHO [13]***
Metric E	***0.14***	0.31	0.29	0.31	**0.26**
Boundary recall	--	0.82	0.86	0.77	**0.89**
Boundary precision	--	**0.81**	0.76	0.72	0.79
Average computational time (sec)	--	232.4	162.7	145.1	**93.3**

[a.] using a PC with Intel dual core 2.2 GHz CPU, 2GB RAM, Matlab©2010b, and Image Processing Toolbox Version 7.1.

V. CONCLUSION

In this paper, we propose a framework to find the best unsupervised metric in the sense that this metric is the most consistent with the ground-truth provided by manual segmentation and, at the same time, is the most sensitive to random segmentation results. We believe a "good" metric should produce a high score on the ground-truth segmentation and produce a low score on random segmentations at the same time. This approach can be applied for any set of unsupervised metrics for image segmentation evaluation. We found the best metric out of six unsupervised metrics in the literature that are commonly used to evaluate segmentation results namely the E-metric, and then used this metric to compare four different segmentation methods, with the DUHO method scoring the best and having by far the lowest computation times.

REFERENCES

[1] Mori, G. Guiding model search using segmentation. in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. 2005.

[2] Felzenszwalb, P.F. and D.P. Huttenlocher, *Efficient Graph-Based Image Segmentation.* Int. J. Comput. Vision, 2004. **59**(2): p. 167-181.

[3] Stockman, G. and L.G. Shapiro, *Computer Vision*2001: Prentice Hall PTR. 608.

[4] Pham, D.L., C. Xu, and J.L. Prince, *Current methods in medical image segmentation.* Annual Review of Biomedical Engineering, 2000. **2**(1): p. 315-337.

[5] Weszka, J. S. Threshold Evaluation Techniques, University of Maryland, College Park, 1998.

[6] Hsin-Chia, C. and W. Sheng-Jyh (2004). *The use of visible color difference in the quantitative evaluation of color image segmentation*. Acoustics, Speech, and Signal Processing, (ICASSP '04). IEEE International Conference, 2004.

[7] Borsotti M., Campadelli P., and Schettini R., "*Quantitative evaluation of color image segmentation results*," Pattern Recognition Letters, vol. 19, pp. 741-747, 1998

[8] Zhang, H., Fritts, J. E., & Goldman, S. A. *An entropy-based objective evaluation method for image segmentation*. SPIE, 5307, 38-49. (2005)

[9] Chabrier, S., B. Emile, H. Laurent, C. Rosenberger, and P. Marche. "*Unsupervised evaluation of image segmentation application to multi-spectral images.*" In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, pp. 576-579. IEEE, 2004.

[10] Rosenberger, C., and K. Chehdi. "Genetic fusion: application to multi-components image segmentation." *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. Vol. 6. IEEE, 2000.

[11] Martin, D., et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. in Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. 2001.

[12] Johson, R., Wichem, D. Applied multivariate statistical analysis. Princeton Hall, 1992.

[13] Trung Duong. *New Methods for Data Clustering and Color Image Segmentation*. Ph.D Dissertation, Oklahoma State University, 2013

[14] Shah, B.N., S.K. Shah, and Y.P. Kosta. A seeded region growing algorithm for spot detection in medical image segmentation. in Image Information Processing (ICIIP), 2011 International Conference on. 2011.

[15] Shih, F.Y. and S. Cheng, *Automatic seeded region growing for color image segmentation.* Image and Vision Computing, 2005. **23**(10): p. 877-886.

[16] Garcia Ugarriza, L., et al., *Automatic Image Segmentation by Dynamic Region Growth and Multiresolution Merging.* Image Processing, IEEE Transactions on, 2009. **18**(10): p. 2275-2288.

[17] Martínez-Usó, Adolfo, Filiberto Pla, and Pedro García-Sevilla. "Unsupervised colour image segmentation by low-level perceptual grouping." *Pattern Analysis & Applications* (2011): 1-14.

[18] Deng, Yining, and B. S. Manjunath. "Unsupervised segmentation of color-texture regions in images and video." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.8 (2001): 800-810.

[19] Powers, David M W (2007/2011). "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies* **2** (1): 37–63.

[20] T. H. Duong, M. Emami and L. L. Hoberock, "Automatic Dishware Inspection: Applications and Comparisons of Two New Methods," Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, Honolulu, HI, 2011, pp. 25-30.

Fig. 3. : Some Comparison: row (a): original image; (b) segmentation results from human hand-label (ground truths), (c) PSEG, (d) GSEG, (e) JSEG, and (f) DUHO.

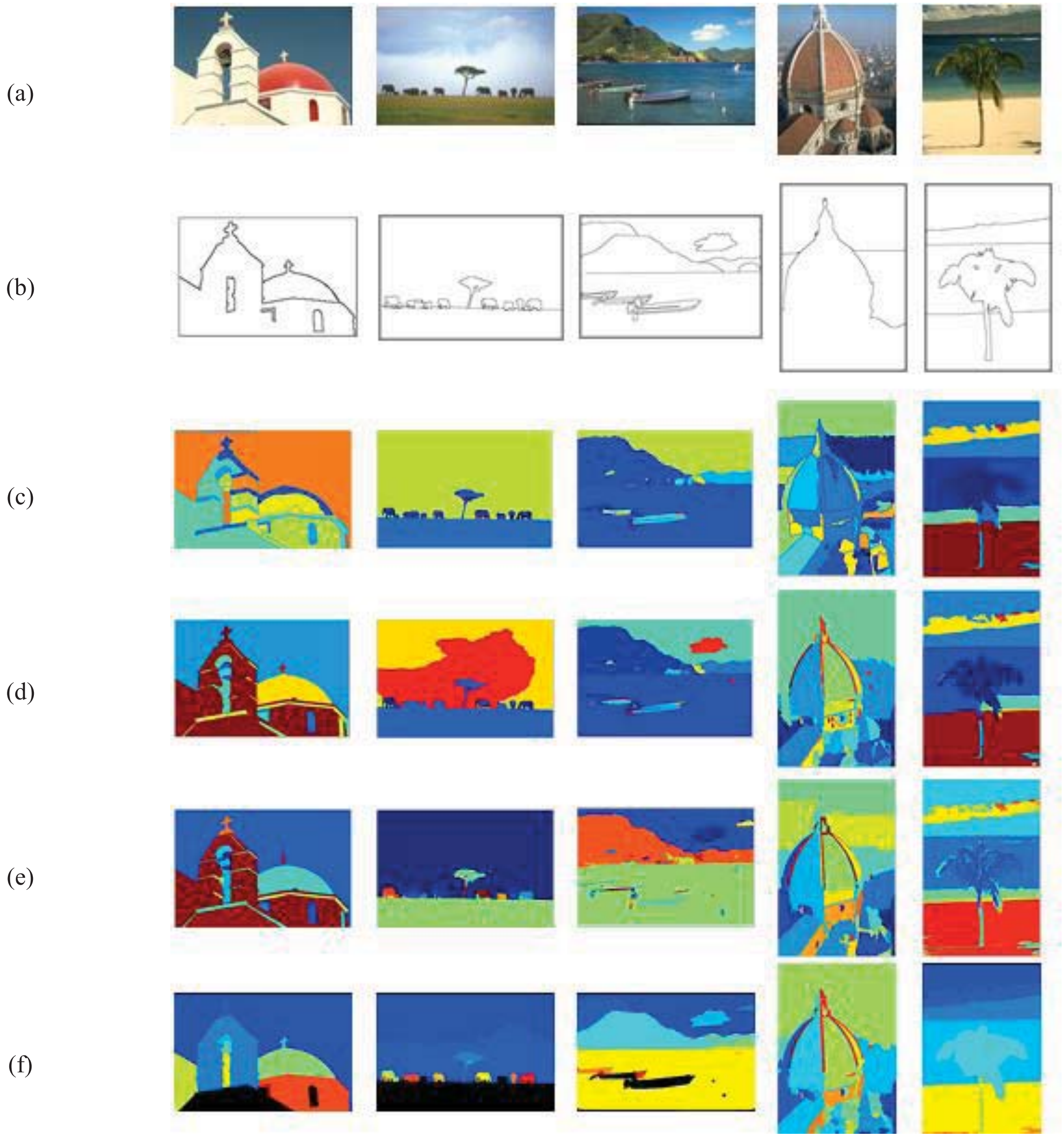


Image Processing for Data Acquisition and Machine Learning of Helicopter Flight Dynamics

Dan Tappan and Matt Hempleman
Department of Computer Science, Eastern Washington University, Cheney, WA, USA

Abstract - *Learning to fly a full-sized helicopter is a complex iterative process of mapping interdependent causes to effects via inputs to outputs in real time in a wildly dynamic, messy, and unforgiving environment. This work presents a prototype system for noninvasively acquiring otherwise inaccessible data from the controls, instruments, and flight dynamics of a Robinson R22 helicopter with an array of cameras and sensors and then processing these images with OpenCV-based solutions into corresponding numerical form for later use in a machine-learning project. It describes a hardware and software architecture for safely and successfully calibrating the system, running a breadth and depth of representative experiments, and qualitatively and quantitatively presenting and validating the results.*

Keywords: feature extraction, data acquisition, machine learning, aviation

1 Introduction

Autonomous aircraft, especially consumer drones, have become an $11 billion yearly industry [9]. Machines can learn to fly well for many mainstream purposes now, but most approaches are disconnected from the way human pilots learn to fly [6]. The computational models provide little insight into the learning processes of either group. A better understanding would advance the field of artificial intelligence and intelligent systems. It could also extend this capability to other environments where machine learning might be advantageous.

This paper addresses the first two objectives of a larger project: (1) to build an acquisition system for recording flight data from a full-sized helicopter; (2) to collect data from basic flight maneuvers as representative teaching examples of how to perform them; (3) to investigate data processing and fusion techniques to merge data from numerous repetitions of maneuvers done to account for variation and errors; (4) to build a rudimentary software flight-dynamics model based on the nature of the collected data; and (5) to investigate machine-learning techniques to allow the system to learn and explain how to perform the same actions as the human pilot (Tappan).

The key element is to acquire real-world data from a Robinson R22 two-place helicopter, which is the world's most popular trainer [15]. Its wide range of capabilities and relatively low operating cost make it convenient for such activities. However, its primitive instrumentation provides no capability to log flight data directly. This limitation is significant because the machine-learning project must have the same awareness that a human pilot has, namely visual perception of the outside world and an understanding of the internal state of the helicopter by visually observing its instrumentation. Outfitting the helicopter with a complex array of sensors, as is common in other work, would undoubtedly be more effective, but a human pilot does not learn to fly based on such unnatural stimuli [4].

Two aspects of this proof-of-concept solution are considered here: the architecture for visual data acquisition, and an OpenCV-based postprocessing system for converting these data into usable numerical form [13]. The primary requirements address safety and practicality (in no order): no interference (physical or electrical) with the helicopter; no attachments at all outside, and no substantive ones inside; ease of setup and tear down; minimal wiring; the fewest number of cameras in the least obtrusive places; and no distraction for the pilot. Section 3 covers the technical requirements.

2 Background

A helicopter exhibits six degrees of freedom in its physical state: it has a position in space on the *x*, *y*, and *z* axes and an orientation respectively in *roll*, *pitch*, and *yaw* (collectively known as attitude) about them. A seventh variable is *time*, which contributes to computing the speed (change in state) and acceleration (change in speed) of the other six. This work assumes the coordinate system in Figure 1a.

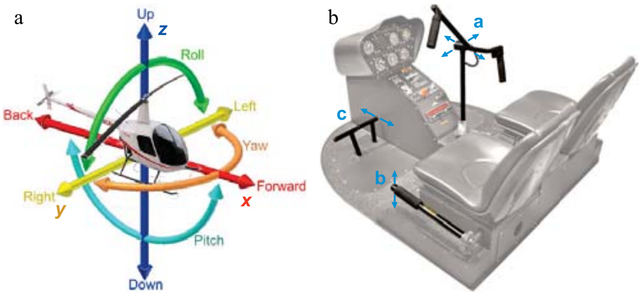


Figure 1: Degrees of freedom and cockpit controls [2,1]

2.1 Control inputs

In order to establish cause-and-effect relationships in flight, the machine-learning system must be able to connect changes in the inputs to their effects on the output state of the helicopter. To this end, the first part of image acquisition monitors the primary flight controls available to the pilot. Most aircraft have dual controls available to both pilots simultaneously, which is essential in a training environment. This work assumes only one pilot, sitting on the right.

The *cyclic pitch control* (usually called the "cyclic") is in principle a joystick for the right hand with two degrees of freedom (x and y) such that forward/backward movement affects pitch, and sideways movement affects roll. However, the actual arrangement in Figure 1b (a), known as a *T-bar*, places the pivot in the middle of the cockpit. The downside is that the position of the pilot's hand cannot be directly tracked to determine the corresponding inputs because the teetering nature of the bar allows vertical movement of roughly 30 centimeters without any actual changes to the input. Section 6.4 covers this issue further.

The *collective pitch control* (the "collective") in Figure 1b (b) is a lever with a vertical arc of travel that changes the amount of thrust from the main rotor to affect the z position (altitude), and through more complex interactions, the x and y positions. While its range of motion is more regular than that of the cyclic, it is mostly obscured by the seats and the pilot's left arm. On the end of the lever is the throttle, which is like a motorcycle twist grip. In some helicopters, the pilot manages this input manually, which would require corresponding data acquisition, but the R22 normally operates in automatic mode.

Finally, the *antitorque pedals* (the "pedals") in Figure 1b (c) travel in a forward/backward arc to change the amount of thrust from the tail rotor to affect yaw. The pedals are linked in opposition, so pushing one forward moves the other backward correspondingly. Only one needs to be tracked.

2.2 Augmented outputs

Through complex flight dynamics far beyond the scope of this paper, every input affects the state of the helicopter in multiple interdependent ways. Unlike an airplane, which is always in motion in flight and must generally face its direction of travel, a helicopter is practically unlimited in its maneuverability. This capability offers great flexibility in use, but it has a significant downside for automated data acquisition because most of the fine state awareness is acquired visually by the pilot looking out the window. At least in small helicopters, instrumentation is sparse.

To mitigate this limitation, this work provides quantitative instrumentation in the form of small, inexpensive sensors. The CHR-UM7 integrated inertial measurement unit (IMU) and attitude heading reference system (AHRS) in Figure 2a records all six degrees of freedom [12]. It operates within a local frame of reference, meaning that it is aware of the state of the helicopter relative to itself only, not of its relationship to the world it operates in. In other words, it records changes only; it cannot establish absolute state data like latitude and longitude or altitude. For this purpose, the Parallax LS20031 GPS receiver in Figure 2b supplies x, y, z coordinates and compass heading for yaw, as well as real-world time, for the global frame of reference [14]. This instrumentation is essential for data acquisition in the larger project, but its role in this paper is limited to crosschecking the results from the image acquisition and processing.

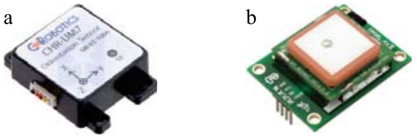


Figure 2: IMU/AHRS and GPS units [12,14]

2.3 Instrument outputs

The sensors do not interact with the helicopter beyond being simply attached to it internally. Despite the rich quantitative data they provide in native digital form, the overall picture is still incomplete. The visual data available to the pilot from the following cockpit instruments are also needed.

The *altimeter* in Figure 3a measures altitude above sea level in feet. It has three elements of interest: a long needle for hundreds, a short needle for thousands, and a triangle for tens of thousands. Converting them from individual needles into a single value for altitude is a straightforward equation, but it does require establishing their values separately.

The *vertical speed indicator* (VSI) in Figure 3b measures change in altitude in feet per minute. The GPS already provides the equivalent of the altimeter and VSI data. However, it is counter-intuitively *too good* in this role. The cockpit instruments have complex real-world behaviors and limitations that affect how the pilot interprets them, such as a lag in response time. For machine learning to function as a human does, it needs to deal with the same issues.

The *airspeed indicator* (ASI) in Figure 3c measures the speed of the helicopter through the air. The GPS receiver also appears to provide these data, but it actually measures the speed over the ground. Wind conditions almost always cause these two values to be different. The aircraft, and thus the pilot, react to airspeed, which the sensors inside the cockpit cannot measure.

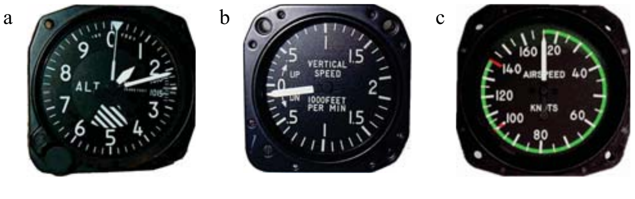


Figure 3: Altimeter, VSI, and ASI instruments

The *manifold pressure gauge* (MAP) in Figure 4a measures the amount of power being demanded from the engine, which varies according to the inputs from the pilot. The acceptable range is based on atmospheric conditions and determined from tables in the pilot's operating handbook.

The combined engine and main-rotor *RPM gauge* in Figure 4b measures the rotations per minute of each as a percent and indicates the acceptable operating range. This instrument is not considered here because of the automatic throttle management, but in other helicopters or more advanced experiments, it would be important. Section 7 covers its value for future work.

Finally, Figure 4c depicts the least high-tech instrument, the *yaw string*, which is a small piece of yarn attached to the front outside of the canopy. It indicates by wind deflection how the nose and tail of the helicopter (essentially the yaw) are aligned with respect to the direction of travel, known as coordinated flight. While this detail could actually be very useful in some contexts, for logistical reasons this output is not considered here. (And it can be derived reasonably well from the sensors.) Similarly, the compass, which also technically provides yaw, is not considered because in practice it is so unreliable as to be almost completely useless, even to a human.

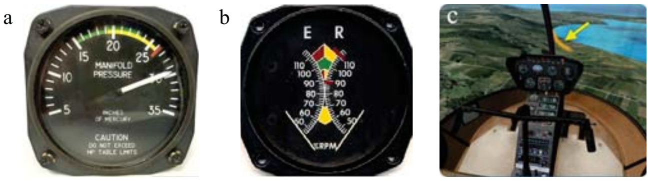


Figure 4: MAP and RPM gauges and yaw string [3]

3 Architecture

The hardware architecture needed to support up to six cameras in simultaneous operation for complete coverage. The requirements were (in no order) that they be inexpensive, small, lightweight, relatively easy to mount, externally powered, have reasonable video quality, store to flash memory cards, and permit remote operation. The FlyCamOne eco V2 in Figure 5a satisfied all these needs remarkably well [11]. Designed to provide a pilot's view in small radio-controlled aircraft, its compact 15-gram package records 24 frames per second of 24-bit color at 720×480 resolution with three megapixels. The image quality from its tiny lens is acceptable, but not great.

Figure 5: FlyCamOne and BeagleBone [11,10]

A critical safety requirement in this work was not to distract the pilot from flying the helicopter. Each test flight generally took an hour and involved several dozen small experiments. The pilot could not afford to be manipulating the system in any complex way to start, run, and end each experiment. (The acquisition system occupied the other seat, so having an assistant was not an option.) The large number of experiments combined with the large number of cameras and sensors required simple one-button coordinated operation to advance to the next experiment.

To this end, the compact BeagleBone Black single-board computer in Figure 5b mapped this button to the appropriate actions [10]. Through serial and I^2C interfaces, it controlled the sensors and recorded their data. Controlling the cameras was similarly convenient because their intended use in radio-controlled aircraft provided a communication interface through standard pulse-width-modulated (PWM) servo signals. The camera data, however, were stored on the 8GB microSD memory cards in the cameras themselves. Transferring so much data over such a distance on lightweight unshielded cables to a relatively weak computer was not an option, so the BeagleBone could not manage the files itself. (In earlier proof-of-concept tests, even a high-powered laptop was unable to keep up with six comparable cameras connected via USB.)

This solution introduced a major problem with synchronizing the files across all the cameras because each camera generates a different filename with no timestamp when started. Therefore, after a flight, it was almost impossible to determine which file referred to which experiment. Conveniently, however, these cameras also record audio. Each time the BeagleBone instructed the cameras to start recording, it played a Morse code-like preamble identifying the automatically generated test number. While not particularly human-friendly, this code provided enough information to change the filenames by hand to something meaningful later. The BeagleBone also generated a second tone sequence every five seconds to ensure that the timing across all videos could be synchronized when startup delays occurred or the recording rates were not exactly the same.

4 Image processing

Image processing is the core of this work, but this paper is primarily about the architecture that facilitated it. It plays the role of postprocessing the videos into a series of values that correspond to the state of the controls and instruments. The processing itself is relatively straightforward and uses traditional approaches in OpenCV as intended. Hempleman [5] provides a very detailed description to supplement the summary here.

4.1 Controls

The controls come in three forms with related types of linear or angular motion, so the same image-processing approach could be applied to each. The most important aspect was being able to track a known object affixed to key points on the controls. This step entailed significant *what if* experimentation to find a satisfactory (but never ideal) solution. The requirements limited the object to being something small and unobtrusive, like a sticker. Selecting the size and color alone could be its own paper because image acquisition operated under such a wide range of environmental conditions. (See Section 6.4.) This part investigated dozens of combinations of swatches made of every conceivable colored tape and paper, as well as 19 small LEDs. Similarly, camera placement entailed many experiments to find reasonable compromises within the tiny, cramped cockpit. This section summarizes the overall approach of color-based blob detection, the details of which often varied depending on the particular goals and actual conditions, etc.

Although the lighting, contrast, and other uncontrollable dynamic factors varied wildly in the cockpit, nothing else consistently resembled the roughly 8mm reflective orange tape squares on a black background in Figure 6. Color, hue, and saturation isolation were usually able to find this object within the expected region. The controls do not normally move quickly, so tracking the position of a known object at 24 frames per second was reliable. However, different positions of both the controls and the helicopter itself changed the target color threshold frequently. To mitigate this variation, the tracking algorithm started with the exact color to find (or its components) and then relaxed the requirements iteratively until it found a strong match. If it could not, it ignored these frames until it could again. Further postprocessing into machine-learning data interpolated any missing frames, assuming that the missed motion was linear.

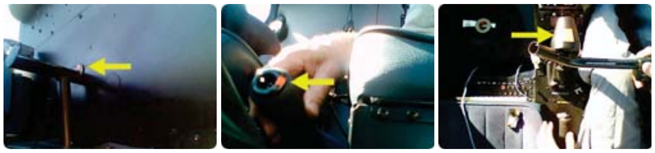


Figure 6: Pedal, collective, and cyclic tracking objects

With the target object isolated within the frame and the bounds of the calibrated range known (see Section 5), it was a straightforward algebra problem to translate the centroid of the object to its corresponding approximate coordinates.

4.2 Instruments

The instruments also share many commonalities in their presentation and behavior, so generally the same image-processing approach could be applied to each. However, the need for finer resolution combined with the presence of smaller features, more clutter, interference and distortion, and a lack of pilot-provided tracking objects, proved more challenging. Unlike the controls, attaching anything to the needles was not an option because they are in sealed glass cases. Even worse is that both the needles and the information on the instruments are usually presented in the same white on black. Due to space limitations, this section summarizes the general process that applied to all the instruments. Each instrument also had its own positive and negative aspects and idiosyncrasies to accommodate.

Interpreting the instruments first involved knowing where they were. The camera responsible for this perspective was mounted in the middle of the cockpit facing forward (see Figure 1b). The top and left edges of the instrument panel form a high-contrast fixed reference that helped automatically establish the exact scale and bounds of the instrument region, which then established the position of the instruments, as in the top row of Figure 7. Next came contrast normalization to improve the boundary between the needles and the background. This process involved redistributing the histogram representation of the colors in use over the entire range available, thereby spreading similar colors farther apart. The standard luminosity method then converted these new colors to grayscale. Tests showed that under normal conditions, the needles were (by design) almost always the most prominent feature. The primary color value of the needle thus became the binary threshold by which all pixels were finally converted into either pure black or white, as in the bottom row.

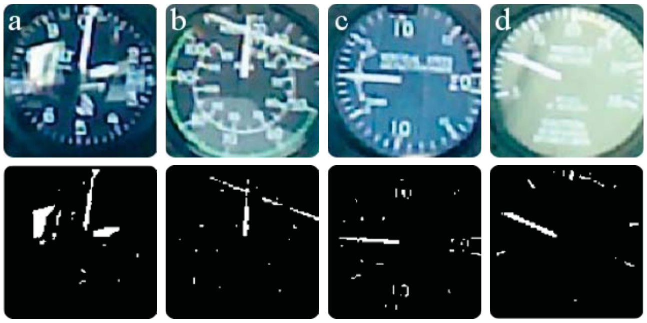


Figure 7: Original and binary-thresholded images

The needles are normally the most prominent linear features, called blobs. Glare can produce artifacts, but the shape does not normally lend itself to recognition as a line, as in Figure 7 (a). When it does, as in (b), the needle still tends to be larger, as well as in its expected position and a legal orientation. In the event that no line is found, two relaxation methods take over. The first incrementally erodes the image in an attempt to break up congealed features until the needle is present. If this attempt fails, then the opposite occurs to dilate the image to join separate features until they form a blob. If both fail, no instrument data are recorded for this frame.

Running a best-fit line approximation on the blob produces an angle, which maps to the predefined numerical scale on the instrument dial for the state value to record. In the case of the altimeter, there are two needles to isolate. (The triangle for tens of thousands of feet was not considered because no flight tests took place so high.) When the needles are far enough apart to differentiate, the process is identical. When they are fused, however, the centroid of this superblob still suffices. Later cleanup for machine learning could interpolate from the last known separate values, but in practice, this occlusion (which also happens to pilots) is not an issue. The altimeter is not precise enough anyway.

5 Calibration and experiments

For reliable, repeatable measurements of the controls and instruments, each in-flight testing session required an initial calibration stage to ensure that the same states mapped acceptably close to the same values. In fact, calibration was actually necessary before *and after* each session to verify that no changes occurred from vibration. This calibration qualified as experiments in its own right because it permitted comparison of the actual values to the expected.

5.1 Static experiments

It is extremely difficult to establish a set of ground-truth states during real flight maneuvers because the dynamic operating environment is so messy; i.e., the expected values are not precisely known. The sensors provided some capability for cross-checking, but their coverage was limited. To establish best-case baseline performance, the initial tests were static on a non-operating helicopter.

5.1.1 Controls

The pedals travel along a known arc with three natural calibration points: full forward, full backward, and half way, which is straightforward to determine because both pedals are adjacent. Similarly, the collective has full down and up positions. A vertical calibration jig with known angles printed on a poster board established intermediate points. The cyclic, however, was troublesome. Its range of two-dimensional motion is over a large horizontal plane whose limits exceeded the field of view of any single camera. Data collection in flight was not affected because the cyclic rarely reaches these limits; however, calibration did need them, or at least an equivalent. To this end, a similar jig with a rectangular internal cutout established the known limits for the center post, which also established the neutral center position. However, positioning the jig itself was tricky because there are few convenient fixed reference points in the cramped cockpit. This process looked ridiculous because it involved aligning small stick-on bubble levels and a lot of contortion, but it was actually effective.

5.1.2 Instruments

Static calibration of the instruments was far more limited because there is almost no access to their needles. Only the altimeter has a knob that changes the internal value, but its range is limited to a thousand feet or so. Instead, the simplest approach proved most effective: color printouts of the instruments to scale with known needle positions taped over the actual instruments. While this approach did not account for the visual disturbances covered in the next section, calibration would be inappropriate under such suboptimal conditions anyway.

5.2 Dynamic experiments

The dynamic experiments involved a breadth and depth of representative flight maneuvers. The purpose was to test the data acquisition system, not to collect actual data for machine learning. It was therefore not necessary to demonstrate more than one acceptable representative exemplar of each. Data for machine learning actually requires many such samples for filtering, smoothing, averaging, fusion, complex statistical analyses, etc. beyond the scope here.

The first set involved airborne maneuvers. They exhibited relatively large changes in the inputs and instruments:

- straight and level at a constant speed, accelerating, and decelerating
- straight with a shallow climb at a fixed climb rate
- straight with a steep climb
- straight with a shallow descent at a fixed speed
- straight with a steep descent
- straight with shallow sinusoidal climbs and descents
- right turn level
- right turn with a shallow climb
- right turn with a steep climb at a fixed speed
- right turn with a shallow descent
- right turn with steep descent at a fixed descent rate
- a left rectangular runway traffic pattern: taking off, climbing, leveling off, descending, and landing

The second set was near the ground with small changes:

- stationary hover
- pivot turn: stationary while rotating about the z axis
- square, circle, and figure 8: always facing forward, and always facing the center of the shape

6 Results and discussion

Evaluating results in terms of the agreement between actual and expected values is difficult when the former are messy and the latter are not definitively known. There was significant variation in the experiments caused by pilot error and uncontrollable conditions like wind, as well as measurement errors in the sensors themselves. This paper focuses on the raw image acquisition, not on their complex postprocessing into cleaner form, so the discussion of the results is mostly subjective. To mitigate biases, however, there were several complementary approaches.

6.1 Qualitative internal validation

Qualitative validation involved the pilot reviewing the results of each test in a form that consolidated the inputs and outputs into a meaningful representation. The instrument panel in Figure 8 shows the results derived from the image processing and the sensors [7]. Validation in this form is based on whether the instrument view internally from the pilot's local perspective in the virtual cockpit was consistent with performing the maneuvers correctly. Here the low-level data combine with the high-level knowledge and wisdom of the experienced human pilot to make informed interpretations and evaluations.

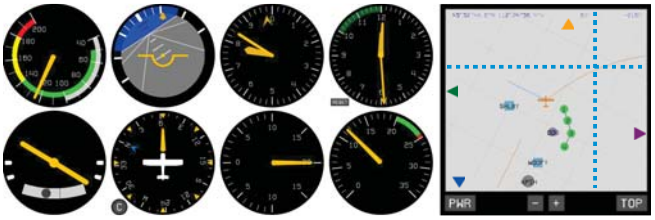


Figure 8: Virtual instrument panel and control indicator

Also available were the underlying raw values strategically plotted in Excel in Figure 9 to show relationships. The exact values are not so important as the trends and behaviors. For example, discontinuities and abrupt jerks inconsistent with regular flight can be attributed to acquisition errors because no exemplars with such events were used.

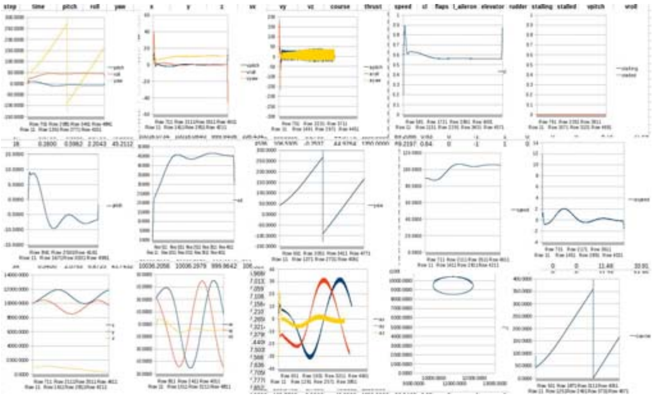


Figure 9: Excel data plots

6.2 Qualitative external validation

Qualitative external validation was also from the pilot's perspective, but from outside the cockpit. Plotting this global representation in two and three dimensions, along with helpful metadetails, as in Figure 10, provided another valuable consistency check [8]. However, unlike the internal view, a pilot is not ordinarily familiar with interpreting the cause-and-effect relationships of inputs to outputs this way. Still, the same kinds of discontinuities would be apparent.

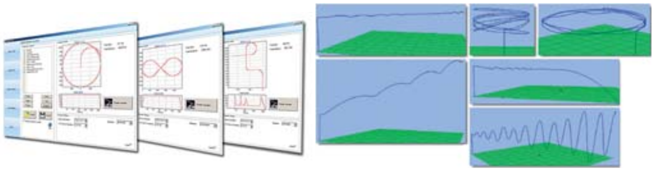


Figure 10: 2D and 3D visualization

Finally, for a richly integrated perspective, the position data exported directly to Google Earth, as in Figure 11.

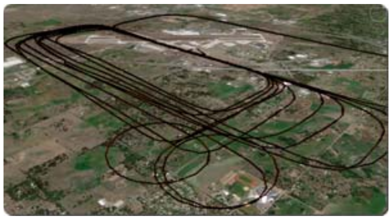


Figure 11: Google Earth track

6.3 Quantitative validation

Quantitative validation was as objective as possible given the current limitations of this prototype system. The first set of tests was a variant on the calibration process, in which the image processing analyzed color printouts with known needle positions. These tests were static because there was no way to change the needle positions without substituting another image by hand. The second set involved dynamic tests in flight where the sensors provided a reasonable approximation of the expected values. However, over time these sensors had an unfortunate tendency to drift out of calibration. They could not be recalibrated in flight, so usually only the earlier tests in a session were reliable.

6.4 Observations

The weakest link in the image processing is the cameras. On the positive side, consistency in accurate positioning was not a factor because the postprocessing successfully isolated features and produced comparable results from any reasonably similar position and perspective. Likewise, the endless vibration inherent throughout all tests surprisingly played no significant role (except in gradually nudging cameras out of alignment at times). The frame rate was high enough to capture redundant images that effectively canceled it out. On the negative side, the dynamic range of the camera sensors is poor and tends to smear colors and especially wash them out toward the low and high ends of brightness, which degraded contrast. The small lens likely contributed to this problem, which means that higher resolution alone would probably not be an improvement.

Object tracking for the controls was very effective. Except in cases where the orange square was completely washed out, the postprocessing generated positions that were within a few percent of the believed expected values. Still, for the larger project, this resolution turns out to be problematic because much of flying a helicopter involves subtle control movements — very often just pressure, not even overt movement. Large movements are comparatively rare, especially in ground maneuvers, which are the ones of greatest interest. Similarly, manual inspection shows that there is noticeable backlash (slop) in the cyclic, especially vertically, which means that small movements do not always translate into actual inputs. The size of the tracking square also plays a role: larger area is easier to follow, but it requires more movement before the image processing perceives it. Poor contrast causes the edges to appear to change, which affects the centroid that translates to the position value. Averaging multiple frames helps stabilize the raw values by smoothing them, but it simultaneously smooths away desired movements and hides actual changes until they become larger.

Needle localization was also very effective in most nonpathological cases, isolating 92 out of 100 frames on average. Translating needle positions to absolute values was always within 2.9 degrees of the expected values, with 82% being within 1.5 degrees. This error is completely acceptable because the instruments themselves are not this precise. The only influences that could not be overcome were lens flare and glare from sunlight, but even the human pilot was rarely able to interpret such cases.

7 Future work

From the standpoint of technology, better cameras would improve the results. FlyCam now offers (at a much higher price) an HD 1080P version with a larger lens, which appears to be a drop-in replacement for the ones used here [11]. Even better would be to reduce the complexity of having many cameras and use one 4K high-resolution GoPro with a fisheye lens. This approach would require another stage of image processing to correct for the spherical distortion, which could introduce its own issues, but the idea seems promising. Likewise, using better sensors and more of them for error detection and correction would provide better baseline data for performance evaluation.

Future work on the methodology and testing will involve a much richer breadth and depth of experiments, as well as many repetitions of them. This effort could lead to improved results and a better mechanism for quantitatively evaluating accuracy and precision.

Finally, future projects could involve the yaw string for actual data about the aerodynamic behavior of the helicopter in flight, which the sensor-derived approach only approximates. Similarly, using the view of the outside world could contribute to machine learning of hovering based on visual references, which is the hardest part of learning to fly for a human. Finally, despite the R22's predominant role as a training helicopter, it has few features that help the student recognize when they are doing something wrong. In particular, the maximum manifold pressure is extremely easy to exceed when a student is focused/fixated on other activities. (This condition does not cause instantaneous destruction of the engine, but it does reduce its operational life over time.) A simple warning tone based on an automated observation of the gauge would be helpful. Other such conveniences are also likely possible.

8 Conclusion

This proof-of-concept work successfully showed that an array of inexpensive cameras can collect data from complex control inputs and instrumentation outputs. The architecture met all the safety and performance requirements, although the latter could use improvement from better cameras. The image processing was able to isolate features reliably and translate their states into numerical form for later use in machine learning.

9 References

[1] Adapted from oe-xam-simulators.blogspot.com.

[2] Adapted from wikipedia.org and turbosquid.com.

[3] Adapted from X-Plane flight simulator, x-plane.com, last accessed Mar. 12, 2016.

[4] Coates, A., P. Abbeel, and A. Ng. *Apprenticeship Learning for Helicopter Control*. CACM, vol. 52, no. 7, pp. 97–105, 2009.

[5] Hempleman, M. *Image Processing for Machine Learning of Helicopter Flight Dynamics*. Masters thesis, Eastern Washington University, 2015.

[6] Liem, R. *Surrogate Modeling for Large-Scale Black-Box Systems*. Masters Thesis, MIT, Sep. 2007.

[7] Tappan, D. and M. Hempleman. *Toward Introspective Human Versus Machine Learning of Simulated Airplane Flight Dynamics*. 25th Modern Artificial Intelligence and Cognitive Science Conference, Spokane, WA, Apr. 26, 2014.

[8] Tappan, D. *A Pedagogy-Oriented Modeling and Simulation Environment for AI Scenarios*. WorldComp International Conference on Artificial Intelligence, Las Vegas, NV, July 13–16, 2009.

[9] Walter, L. "Yearly UAV Market Exceeds $11B." Aviation Week & Space Technology, Aug. 18, 2013.

[10] www.beagleboard.org, last accessed Mar. 17, 2016.

[11] www.camonetec.com, last accessed Mar. 17, 2016.

[12] www.chrobotics.com, last accessed Mar. 17, 2016.

[13] www.opencv.org, last accessed Mar. 17, 2016.

[14] www.parallax.com, last accessed Mar. 17, 2016.

[15] www.robinsonheli.com, last accessed Mar. 17, 2016.

Enhancing Bag of Visual Words with Color Information for Iconic Image Classification

Stephan Kopf[1], Mariia Zrianina[1], Benjamin Guthier[1], Lydia Weiland[2],
Philipp Schaber[1], Simone Ponzetto[2], Wolfgang Effelsberg[1]
[1] Department of Computer Science IV, [2] Data and Web Science Group
University of Mannheim, Germany
Email: {kopf, zrianina, guthier, lydia, schaber, simone, effelsberg}@informatik.uni-mannheim.de

***Abstract*—Iconic images represent an abstract topic and use a presentation that is intuitively understood within a certain cultural context. For example, the abstract topic "global warming" may be represented by a polar bear standing alone on an ice floe. This paper presents a system for the classification of iconic images. It uses a variation of the Bag of Visual Words approach with enhanced feature descriptors. Our novel color pyramids feature incorporates color information into the classification scheme. It improves the average F1 measure of the classification by 0.118.**

Keywords: semantic image search, iconic images

I. Introduction

When searching for multimedia content, image search engines like Google Images or Flickr find a large number of pictures. Most commercial search engines rely heavily on a textual description that surrounds the content, for example on a Web page or that was added manually. These search engines work well, if the topic to be searched for can be labeled with a brief and meaningful description. However, it is not always easy for users to find suitable keywords to be used in the search. This becomes even more challenging when searching for abstract topics like "climate change". Such a search request may be answered by a large variety of multimedia content. Images may show reasons for climate change, e.g., "air pollution" or possible solutions like "wind turbines". Figure 1 shows two example results of such a search query.

In this paper, our aim is to search for *iconic images*. In an iconic image, the visualized objects are not relevant on their own, but the complete scene represents a larger, more abstract topic which is understood intuitively within a certain cultural context. An example would be the picture of a polar bear standing on an ice floe. In many western countries, such an iconic picture represents global warming, and it has been used in this context for years. Both photographs in Figure 1 are typical examples of iconic images as well. Smokestacks can be considered iconic if they are associated with the topic of "air pollution", which in turn represents the larger theme "climate change". A lot of previous work (e.g., [1, 2]) focuses on identifying *canonical* (representative) views of similar scenes. Some authors label these canonical images as *iconic*. We do not follow this definition of iconic images.

Our long-term goal is the automatic classification of rich semantic concepts in images. This paper goes one step towards our goal and presents a complete system that allows the automatic classification of iconic images. We use the bag of visual words (BoVW) method as a starting point. Our color pyramid scheme improves the basic algorithm and increases its accuracy for iconic image search.

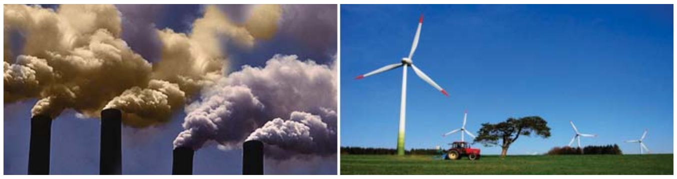


Fig. 1. Two iconic images of climate change. Smokestacks (causal attribution of climate change) and wind turbines (proposed solution) [3].

The rest of this paper is structured as follows. Section II discusses methods that extend the basic BoVW algorithm and other approaches of classifying iconic images. The used algorithm as well as the developed color pyramid feature are described in Section III. Sections IV and V present our dataset and the experimental results, respectively. Section VI summarizes the paper.

II. Related Work

Much work has been done in the area of image classification and content-based image retrieval, most of which focuses on the retrieval of specific objects [4, 5, 6] or scene categories [7, 8]. One of the most successful techniques is the Bag of Visual Words (BoVW) approach which has been widely studied in the literature [9, 10]. Several improvements for the general BoVW approach have been proposed. Lazebnik et al. [11] additionally include the location of visual words and use it alongside the visual histograms that represent the frequency of visual words for each image. Sharma [12] extended this method by adding saliency information. The computed features are weighted with their corresponding value in the saliency map.

Using the definition of [3], an iconic image concisely represents an entity that refers to a larger topic, and that is widely used in public communication. Such a topic is identified easily by media users and can trigger a substantial affective, cognitive and/or behavioral reaction. The only work in the context of image retrieval that considers iconic images was proposed by Ponzetto et al. [3]. The approach starts with a human-selected basic set of iconic images along with their caption. This basic set is enlarged by using a query-by-text approach from the images' descriptions. Outliers are filtered out in the final step.

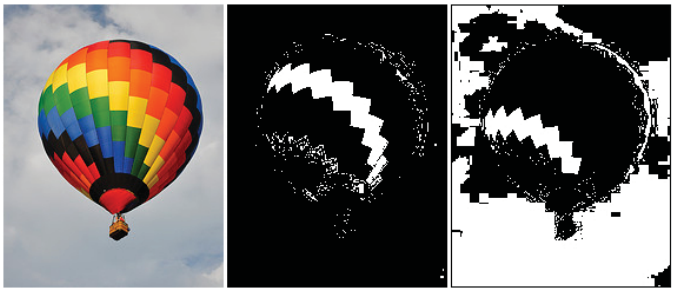


Fig. 2. An example of color masks. Left: original image. Center: color mask with hue value 30 (orange). Right: color mask with hue value 110 (blue). Color masks are computed with a range of 10. The dilation filter has not been applied in this example.

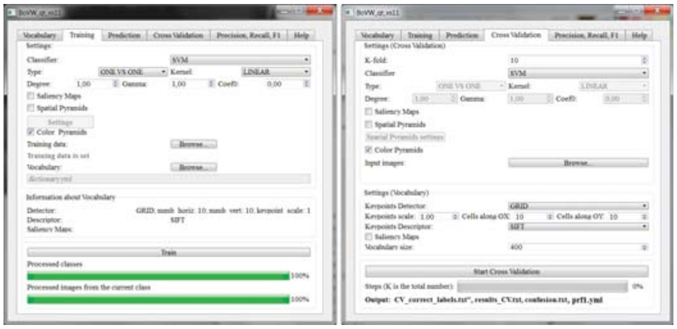


Fig. 3. GUI of the iconic image classification system. Training the classifier (left) and validation (right).

III. Classification System

In the following subsection, the basic BoVW approach is briefly discussed, followed by our proposed extension that uses color pyramids. Implementation details are given at the end of this Section.

A. Bag of Visual Words Approach

Building a vocabulary of visual words includes the detection of keypoints, the computation of descriptors, and clustering. We use the SIFT and the SURF detectors and descriptors in our system. The computationally efficient GRID detector was also added. It divides the image into equally sized cells where the center of each cell is considered as a point of interest. The descriptors that are obtained from a training set of images are clustered using the k-means algorithm. The set of computed centroids then defines the vocabulary of visual words. Our system allows the user to manually define the vocabulary size.

The next step is the training of a classifier on the labeled training dataset. We implemented the two classifiers: Support Vector Machines (SVM) and Normal Bayes Classifier (NBC). To predict a label for a new, unknown image, a visual word histogram for this image is calculated and passed to the trained classifier.

B. Color Pyramids Feature

Feature descriptors like SIFT or SURF use the local contrast but do not consider color information. We propose *color pyramids* as a novel feature to enhance the basic BoVW method with color information. The idea of color pyramids was motivated by the concept of spatial pyramids as presented by Lazebnik et al. [11]. Instead of dividing an image into spatial sub-regions, it is divided into color sub-regions (e.g., assigning a color to each pixel). If coarse to fine color intervals are used, a hierarchy is created that is similar to spatial pyramids. E.g., the colors red, yellow, green, or blue may be used on the coarsest level, and the next level splits each color into several subcolors.

In a first step, keypoints and descriptors are calculated in the original input image. To compute the color pyramids feature, the input image is converted into the HSV color space, and all channels but hue are discarded. L evenly spaced values c_k from the hue channel ($k = 1 \dots L$) are selected and L color masks M_k are calculated that contain a range r of colors around each value. The color mask M_k is defined as

$$M_k(x,y) = \begin{cases} 255, & I(x,y) \in [c_k - r, c_k + r), \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where I denotes the source image and (x, y) is the pixel coordinate. Optionally, color masks are smoothed to reduce noise. Figure 2 shows an example of two computed color masks. White means that a pixel's color is within the color range of the mask, and black means that it is not.

Next, a complete histogram of visual word vectors is computed by using all keypoints along with their descriptors. For each created color mask $M_k, k \in 1, ..., L$, all keypoints that lie on black pixels in the mask are filtered out. By using only the remaining keypoints and their descriptors, a histogram of visual word vectors v_k is computed that is specific to the considered color mask M_k. All vectors v_k are then concatenated into one large visual word histogram vector $(v_1, v_2, ..., v_L)$ that represents an image and also captures its color information. This vector may now contain duplicates of visual words due to the partially overlapping color masks. The vector is then used as feature in the BoVW method.

C. Implementation

We use the OpenCV library for C++ and the QT framework for the implementation of our system. The SVM implementation is based on the LibSVM library. Our system supports a simple graphical user interface where the functionality is divided into six categories (see the tabs in Figure 3). The source code[1] is available under the GNU public license.

IV. Dataset

Each global topic of iconic images includes several more narrow sub-categories. For example, the categories mushroom, reef, or summer forest are included in the larger topic of biodiversity. By classifying images from such sub-categories, it is possible to distinguish iconic images from the global topics.

The iconic image dataset was generated based on the pipeline described in [3]. The seed images along with their keywords for each topic were chosen based on the results of

[1] The source code of the system is available at: http://ls.wim.uni-mannheim.de/de/pi4/research/projects/iconicimages/

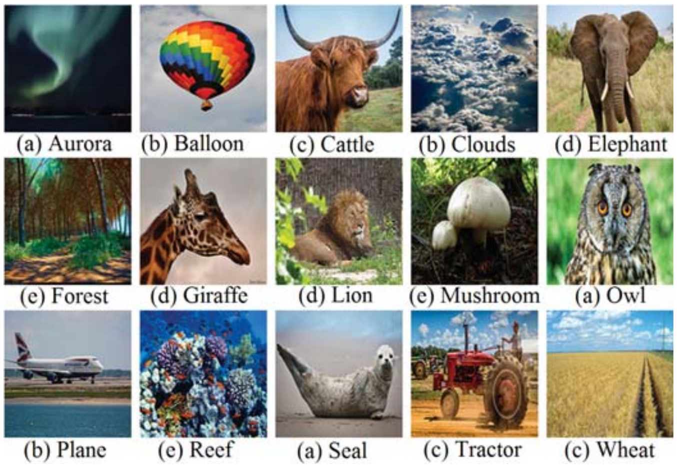


Fig. 4. Image examples from the used dataset. The name of the class label is specified for each category. The global topics are: (a) North Nature, (b) Air, (c) Agriculture, (d) Africa Nature, (e) Biodiversity.

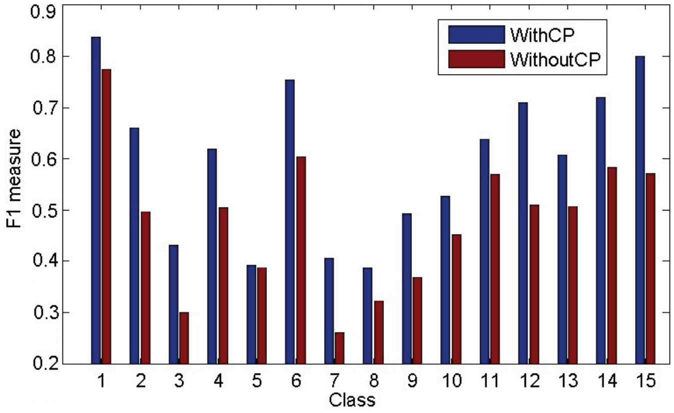


Fig. 5. Achieved F1 measure for each class with and without using color pyramids.

Google image search, restricted to the National Geographic and Wikipedia encyclopedias. Afterwards, based on the requests with the collected keywords which represents names for the used categories, dataset images were gathered from Flickr. Topics that have less than 100 pictures or with licenses other than Creative Commons were not selected. The remaining images were filtered manually based on their correspondence to their topic.

The created dataset consists of fifteen categories with 100 images in each. Each of the categories belongs to one of the five global topics. Figure 4 shows one example image for each of the categories.

V. Experimental Results

This section presents the classification results of our proposed technique that uses color pyramids and compares it to the basic BoVW approach. The color pyramids technique is orthogonal to other existing methods that improve the BoVW approach and can be combined with them. For example, it can be used together with spatial pyramids or features weighted by saliency maps.

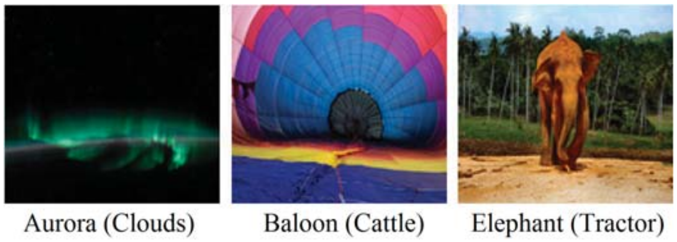


Fig. 6. Examples of false negative classification without using color pyramids. Under each image, its actual class label is written along with the assigned wrong class label in parentheses.

We made detailed preliminary tests to identify suitable and robust parameters for the BoVW approach. For the comparison presented here, the following parameters are used: Keypoints are selected with the GRID method (10×10 regions) in combination with the SIFT descriptor. To achieve multi-class classification, an SVM with a linear kernel and a "one against one" approach was used. One SVM is trained for each pair of classes, and a label for an entity is assigned in a maximum voting process. To implement the color pyramids method, ten different color masks were computed with the hue values equally distributed between 0 and 180. The range was set to 10 to create slightly overlapping color masks. Each mask was then smoothed with a Dilation filter of size 9. The accuracy is computed with tenfold cross validation. If a category that is assigned to an image is incorrect, even though the larger topic is correct, it is still considered as an error.

Figure 5 shows the F1 measure for each category. Using color pyramids as features leads to better results. The biggest increase in F1 measure is 0.144 for the category *giraffe* (class 7). There is no significant benefit (0.0039) in the case of *elephants* (class 5) which are either gray or dark brown. On average, the use of color pyramids improves the F1 measure by 0.118.

When comparing the confusion matrices (see Figure 7), a decrease of type 1 and 2 errors can be seen when using color pyramids. For instance, the category aurora is partially misclassified as clouds in the original approach. This error drops significantly when colors are considered, because the sky in the aurora images usually contains green colors that are unlike the blue sky in the cloud pictures. *Wheat* pictures where the dominant color is beige, are no longer misclassified as *cattle* (green grass), *forest* (green color), *mushroom* (beige, green, yellow, red colors) or *owl* (brown, white, green colors) as often. Figure 6 shows examples of false classifications of the original BoVW approach.

To compare the run time of the standard and the advanced BoVW methods, the dataset was limited to 1000 pictures in total. The same test settings are used as before. We used a laptop[2] for this evaluation that is comparable to standard workplace computers. All time measurements were carried out five times on each machine. Comparing the run time of the different descriptors when training the vocabulary, the SIFT descriptor requires 28 minutes whereas SURF requires 6 min-

[2] Intel Core i7-2670QM (2.20 GHz), 4GB RAM, Windows 7.

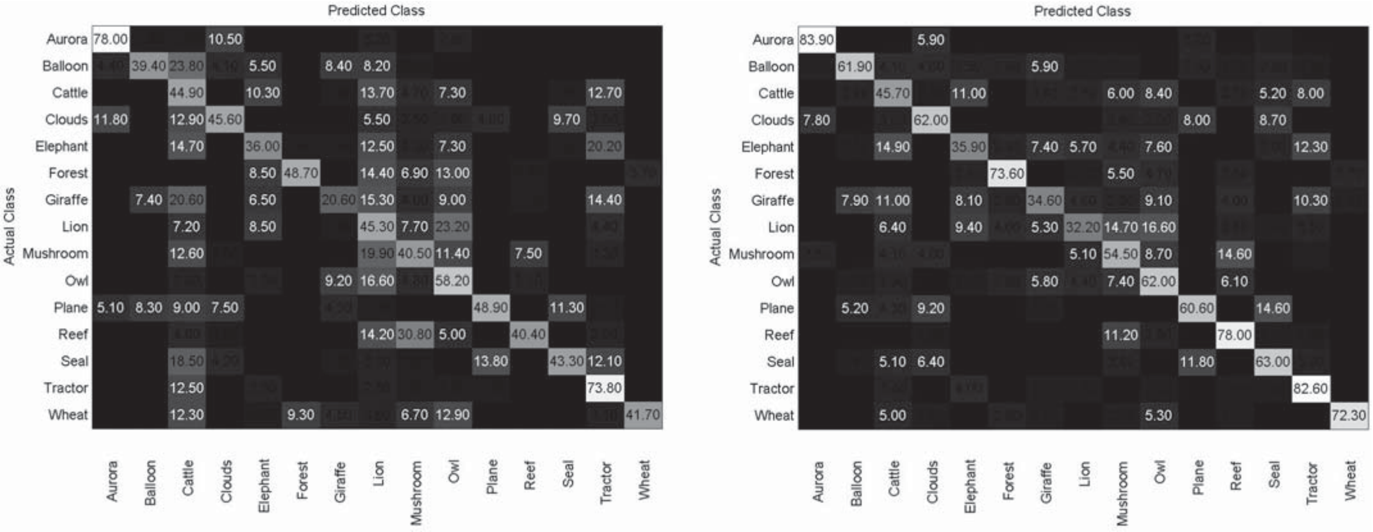


Fig. 7. Confusion matrices comparing the standard BoVW method with the SIFT descriptor (left) and the advanced method using color pyramids (right).

utes only. This is an expected result since SURF descriptors are designed to be computationally efficient. When changing the vocabulary size, the computation time increases with larger vocabulary sizes and varies between 33 minutes (size of 100) to 51 minutes (size of 500). The computation time of the color pyramids depends on the number of color masks used. The run time increases up to a factor of 6, but drops significantly if less than 10 masks are used or if the overlap between the color masks is small.

VI. Conclusion

We presented a system for the classification of iconic images, which is a highly challenging task due to the rich semantics included in such images. As a novel feature, we proposed color pyramids that enhance the standard BoVW method with color information. This makes it possible to distinguish between similar textures like grass or wheat by considering their colors. Using this feature increases the average F1 measure over all categories of iconic images by 0.117. The source code of the system is available for download.

The decision whether an image is iconic or not is still mainly made by a human observer. Familiarity with the global topic and the image context play an important role here. As future work, we would like to develop algorithms that generally answer the question about iconicity in multimedia documents. To achieve this goal, a combined analysis of text and image search will be required.

VII. Acknowledgments

The sample images shown in this paper have been provided by Flickr users under Creative Commons licenses. Thanks to Billy Idle (Aurora), Len Radin (Balloon), Martin Liebermann (Cattle), Farrukh (Clouds), Brittany H. (Elephant), Moyan Brenn (Forest), Steve Wilson (Giraffe), Finn Frode (Lion), Ryan Steele (Mushroom), Edgar Barany C (Owl), Bill Damon (Plane), thinkpanama (Reef), Northwest Power and Conservation Council (Seal), Mobilus In Mobili (Tractor), and Rae Allen (Wheat) for providing the pictures.

References

[1] T. L. Berg and A. C. Berg, "Finding iconic images," in *IEEE CVPR Workshops*, June 2009, pp. 1–8.

[2] R. Raguram and S. Lazebnik, "Computing iconic summaries of general visual concepts," in *IEEE CVPR Workshops*, June 2008, pp. 1–8.

[3] S. P. Ponzetto, H. Wessler, L. Weiland, S. Kopf, W. Effelsberg, and H. Stuckenschmidt, "Automatic classification of iconic images based on a multimodal model," in *Bridging the Gap between Here and There - Combining Multimodal Analysis from International Perspectives, Interdisciplinary Conference on*, 2014.

[4] S. Kopf, T. Haenselmann, and W. Effelsberg, "Shape-based posture and gesture recognition in videos," in *Proc. SPIE 5682, Storage and Retrieval Methods and Applications for Multimedia*, 2005, pp. 114–124.

[5] S. Wilk, S. Kopf, and W. Effelsberg, "Robust tracking for interactive social video," in *IEEE Applications of Computer Vision (WACV)*, 2012, pp. 105–110.

[6] S. Richter, G. Kuehne, and O. Schuster, "Contour-based classification of video objects," in *Proc. SPIE 4315, Storage and Retrieval for Media Databases*, 2001, pp. 608–618.

[7] U. Altintakan and A. Yazici, "Towards effective image classification using class-specific codebooks and distinctive local features," *Multimedia, IEEE Transactions on*, vol. 17, no. 3, pp. 323–332, March 2015.

[8] S. Suchitra and S. Chitrakala, "A survey on scalable image indexing and searching," in *Computing, Communications and Networking Technologies (ICCCNT), Fourth International Conference on*, 2013, pp. 1–5.

[9] J. Mukherjee, J. Mukhopadhyay, and P. Mitra, "A survey on image retrieval performance of different bag of visual words indexing techniques," in *Students' Technology Symposium (TechSym), IEEE*, 2014, pp. 99–104.

[10] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, "Generating descriptive visual words and visual phrases for large-scale image applications," *Image Processing, IEEE Trans. on*, vol. 20, no. 9, pp. 2664–2677, 2011.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE CVPR*, vol. 2, 2006, pp. 2169–2178.

[12] G. Sharma, "Discriminative spatial saliency for image classification," in *IEEE CVPR*. IEEE, 2012, pp. 3506–3513.

A Fast Texture Feature Extraction Method Based on Gabor Wavelets

X.L. Wang[1], **X. Wang**[2] and **L. Hu**[2]
[1]School of Information Engineering, Changan University, Xi'an, Shaanxi, China
[2]School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China

Abstract – *With the development of computer vision, robots need to detect target objects from image sequence for autonomous navigation. To identify targets, the perceptual system of autonomous robots first needs to segment the images into nonoverlapping but meaningful regions based on low-level features such as color, texture measures and shapes etc.. Being an important component, Gabor wavelets are often used to extract the texture features due to its being a mathematical approximation to the spatial receptive field of a simple cell in the V*1 *area of human brain. The problem with these Gabor texture measures is the high computational cost involved in the convolution in the feature extraction process. To partially solve this problem, in this paper, we carefully study the behaviors of the Gabor wavelets used to form the texture features and find out that only a small subset of the filters have important contributions to the identification process. Experimental results show that, by removing redundant filters, better performance can be achieved in a much shorter time.*

Keywords: Gabor Wavelets, Texture Measure, Image Segmentation

1 Introduction

To imitate humans, an intelligent robot should be made up of elements capable of performing functions such as sensing, perception, cognition, planning, control and actuation [1]. Perception is an awareness of things through physical senses, especially, vision. Perceptual learning is the ability to construct compact representations of sensory events based on their statistical properties in the perceptual level as opposed to the behavior or cognitive level [2-3]. To operate in realistic environments and to recognize objects in a given image, a robot must have the ability to form percepts on its own (by natural association among features of sensory information) and, based on that, to segment the image into non-overlapping but meaningful regions whose union is the entire image [4]. In general, by being meaningful, the regions in the segmented image should be homogeneous and should have well-defined boundaries, denoting reliable objects. With high spatial frequencies being filtered out, image segmentation reduces the amount of storage and makes behaviors resistant to the loss of information. Our motivation is to develop an object recognition vision system based on a set of features associated with each object in the image to assist in the analysis of visual data that may be applied successfully across different application domains [5].

Eighty percent of our perceived information about the external world reaches us by way of the eyes. As a result, vision is the primary sensory modality. Discussions of vision often consist of three distinct stages: sensation, perception, and recognition. As the initial stage of sensory transduction, sensation begins with photon receptors in the retina. Perception is the active process of selecting, organizing, interpreting the information detected by sensors, and transforming these raw sensory signals into distinct percepts. Being certain categories in the mind, percepts are the brain's internal representations of specific external visual stimuli and are the results of complicated interactions among multiple visual areas. The goal of perception is to allow people to recognize things through the process of assembling sensory information into a useful and reliable representation of the world. Natural images contain statistical regularities which distinguish objects from each other and from random noise. For successful visual recognition, each object must have attributes that can be used to differentiate it from others and to segment the whole image into meaningful objects.

There are many methods available for feature binding and sensory segmentation in practice. Currently, the most competitive approaches for image segmentation are formulated as clustering models for probabilistic grouping of distributional histogram data. By identifying groups of similar image primitives, the inference essentially amounts to finding local minima of some objective energy function, corresponding to stable states. Here the image primitives can be local features that are spatially smoothed in image patches and can be mathematically represented as vectors in a metric space, giving rise to the feature space. Then, the observed visual data can be explained through prototypical distributions in the feature space. For image segmentation and object identification, instead of using the color information of a single pixel, we consider the color pixels in a simple N×N region of an image, for which the combination of color histogram and Gabor texture measures is a rather standard low level representation to use. As illustrated in Fig.1, to obtain feature vectors, for each image, a moving window of size N×N hops by M pixels in the row and column directions but not to exceed the border of the image. The moving windows are overlapping to allow a certain amount of fuzziness to be incorporated so as to obtain a better segmentation performance. The window size controls the spatial locality of the result and the window hopping step controls the resolution of the result. A decrease in the step gives rise to an increased resolution but an increased processing time. In our approach, for each N×N window, a 10,000-dimensional HSV color histogram is constructed. To take spatial and frequency information into account, Gabor filters are usually used to create a set of texture measures for autonomous learning and classification applications. These measures describe the frequency and orientation of the texture. Given each image, the convolutions with Gabor filters are calculated, followed by the computation of the magnitude of each complex Gabor filter. Next the results within each N × N moving window (as used in color histogram generation) are averaged to obtain a number of texture measures, which are then appended to the color histogram to complete the formation of the feature vector. The model has been successfully applied to image segmentation tasks using 80 Gabor filters [6-7].

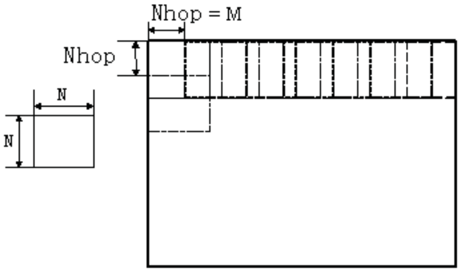


Fig. 1. Image segmentation model

The complex Gabor filters, first introduced by Gabor, are complex exponentials with a Gaussian envelope, or Gaussians modulated by complex harmonics [8]. Being products of Gaussian functions and sinusoidal functions, Gabor functions provide a good mathematical approximation of the spatial receptive field of a simple cell in the *V*1 area of the cerebral cortex. As with pure sine and cosine waves, the Gabor functions (localized frequency filters with Gaussian envelopes) can provide a complete description of any complex waveform. By optimally specifying the combination of spatial frequency and spatial location information, these functions may be utilized to characterize any complex stimulus. Therefore with a visual scene being broken down into a large number of patches, the total information in the scene could be represented by the outputs of an array of Gabor filters with different frequency, orientation, and phase tuning to be convolved with each patch. The problem with these Gabor texture measures is the high computational cost involved in the convolution in the feature extraction process. To partially circumvent this problem, in this paper, we propose to use only a small subset of the 80 Gabor wavelets to form the texture features. Experimental results show that, by removing redundant filters, better performance can be achieved in a much shorter time.

The rest of the paper is organized as follows. In Section 2, we review some related work. We next present our proposed approach in Section 3. In Section 4, an empirical study is conducted to evaluate the performances of our approach. Finally, conclusions are made in Section 5.

2 Relate work

2.1 Color histogram

The color histograms of multicolored objects have been used in a technique called Histogram Intersection as stable object representations to provide a robust, efficient cue for indexing into a large database [9]. For perceptual grouping, histogram clustering model (HCM) was proposed in [10]. To segment an image using HCM, the image is first decomposed into U not necessarily disjoint image patches, each having W features (or bins). For each image patch $i \in U$, a histogram $(h_{i,j})_{1\leq j\leq W}$ is defined as a tuple $h_i \in U^W$. If u_i denotes the number of observations (e.g., pixels) belonging to image patch i, then, $u_i = \sum_j h_{i,j}$. The probability to observe features j in a given image patch i can be estimated by

$$p_i = (p(j \mid i))_{1\leq j\leq W} \tag{1}$$

where the empirical conditional probability is,

$$p(j \mid i) = \frac{h_{i,j}}{u_i} \tag{2}$$

Based on such histogram-based feature vectors, image segmentation can be realized by partitioning the set of image patches into a number of disjoint clusters or segments.

To follow this procedure, images and videos taken from a camcorder are stored for each frame as a RGB (red, green, blue) color image. Next, the RGB color images are converted to the HSV (Hue Saturation Value) color space for use [11]. The Hue describes each color by a normalized number in the range from 0 to 1 starting at red and cycling through yellow, green, cyan, blue, magenta, and back to red. The Saturation describes the vibrancy of the color and represents the purity of a color such as the "redness" of red. The less saturation in a color, the more pale it looks (washed out). The Value describes the brightness of the color. For normalized RGB values in the ranges from 0 to 1, the conversion to HSV is done in the following manner:

$$H = \begin{cases} (0 + \dfrac{G-B}{MAX-MIN}) \times 60 & if \quad R = MAX, \\ (2 + \dfrac{B-R}{MAX-MIN}) \times 60 & if \quad G = MAX, \\ (4 + \dfrac{R-G}{MAX-MIN}) \times 60 & if \quad B = MAX, \end{cases} \tag{3}$$

$$S = \frac{MAX - MIN}{MAX} \quad (4)$$

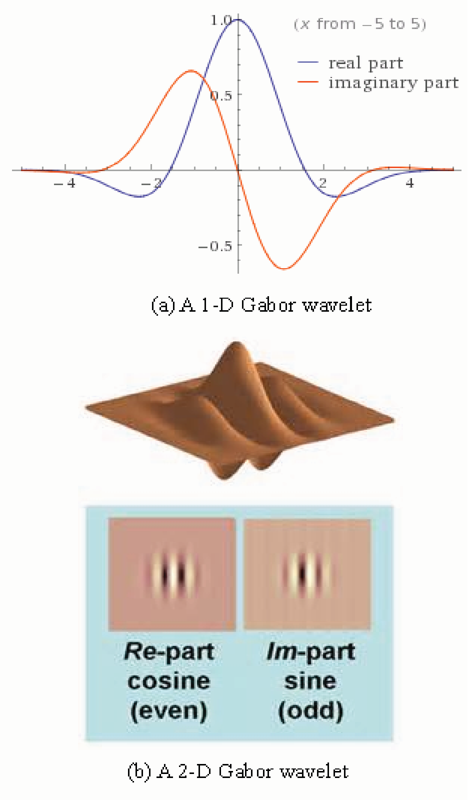
$$V = MAX \quad (5)$$

where *MAX* is the maximum value of (*R*, *G*, *B*), and *MIN* is the minimum. From the above formulas, it can be seen that, if *MAX* = *MIN*, *H* is undefined and *S* = 0, there is no hue and the color lies along the central line of grays, and that, if *MAX* = 0, *V* = 0 and *S* is undefined, the color is pure black and there is no hue, saturation and value.

As the outputs of the above formulas, the Hue values range from 0 to 360, and the Saturation and Value values range from 0 to 1. The Hue values are next normalized to be in the range [0.00, 1.00]. Orange with a bit of red lies in the range [0.00, 0.05], yellow lies in the range [0.05, 0.14], yellow-green lies in the range [0.14, 0.22], green lies in the range [0.22, 0.28], blue-green lies in the range [0.28, 0.45], blue lies in the range [0.45, 0.54], blue-violet lies in the range [0.54, 0.75], purple lies in the range [0.75, 0.81], red-violet lies in the range [0.81, 0.92], and red lies in the range [0.92, 1.00]. To extract color features, a histogram of color measurements in the HSV space for each object is computed as follows. The hue is broken into 100 bins of equal width, that is, [0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 … 1.00]. The saturations and values are evenly distributed into 10 bins, that is, [0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00]. Each color can be represented by combining the three bins, one from the hue bins, one from the saturation bins and one from the value bins. All possibilities of the combinations equal 10,000 different feature color bins for the histogram. The histogram can then be constructed for an image patch by looking at each color feature and finding the number of pixels in the patch that correspond to that feature. After doing this for all the color features in the object, there are 10,000 numbers, each representing the number of pixels of a certain color in the selection patch. The total number of pixels in the selection patch divides these 10,000 numbers, resulting in a highly sparse feature vector of a dimension as high as 10,000. There have been strong theoretical reasons and experimental evidence suggesting that the brain uses a relatively small subset of neurons to represent each information item (e.g., specific sensory stimuli from an object), rather than using either the activity of a single, individually meaningful cell or the global activity pattern across a whole cell population. This is often referred to as sparse coding [5]. Using such a high dimension space in our approach is meant to differentiate colors from each other as much as possible and to mimic sparse coding scenario.

2.2 Gabor wavelets

Wavelets are in many ways like Fourier analysis, but with an important difference that the wavelets have a limited scope. As shown in the upper part of Fig.2, one dimensional Gabor wavelets are basically a sinusoid multiplied by a Gaussian. When a function is convolved with the Gabor wavelet, the frequency information near the center of the Gaussian is captured, and frequency information far away from the center of the Gaussian has a negligible effect. As shown in the lower part of Fig.2, two dimensional form of the Gabor wavelet for image processing consists of a planer sinusoid multiplied by a two dimensional Gaussian. The sine wave is activated by frequency information in the image. The Gaussian insures that the convolution is dominated by the region of the image close to the center of the wavelet.



(a) A 1-D Gabor wavelet

(b) A 2-D Gabor wavelet

Fig. 2. Gabor wavelets (a) 1-D, (b) 2-D

To accurately describe the frequency information of a feature in an image, it is necessary to convolve the location with many instantiations of the wavelet. These instantiations typically sample at different frequencies and different orientations. Two dimensional Gabor wavelets respond to image features that are of the same orientation and frequency as the wavelet. To compute both the real and imaginary part of the wavelet, it is necessary to convolve the image with two masks that are out of phase by $\pi/2$, corresponding to the use of sine and cosine in the wavelet transform. Gabor wavelets are described by parameters that control orientation, frequency, phase, size, and aspect ratio and can take a variety of different forms. To fully understand what each of these parameters means, we will look at the full wavelet equation and discuss each of the parameters in turn.

The complex Gabor filters are complex exponentials with a Gaussian envelope, or Gaussians modulated by complex harmonics. From experimental data fitting, a mathematical approximation of the spatial receptive field of a simple cell can be provided by a Gabor function. When placing the origin of the coordinates at the center of the receptive field, the observed receptive field structures using a Gabor function can be approximated as,

$$G(x,y,\theta,\lambda,\varphi,\sigma,\gamma) = e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}} e^{j(2\pi\frac{x'}{\lambda}+\varphi)} \quad (6)$$

where $x' = x\cos\theta + y\sin\theta$, $y' = y\cos\theta - x\sin\theta$, θ specifies the orientation of the wavelet, λ specifies the wavelength of the sine and cosine waves and determines the spacing of light and dark bars that produce the maximum response, φ specifies the phase of the sine and cosine waves and determines where the ON-OFF boundaries fall within the receptive field, σ specifies the radius of the Gaussian, and finally, γ specifies the aspect ratio of the Gaussian. These five parameters control the wavelet. Given each image patch as used in color histogram generation, the convolutions of the image patch with each of these Gabor wavelets are calculated to generate the orientation features. In other words, each wavelet coefficient is computed by convolving a location in the image with the wavelet kernel and captures information about one combination of phase, orientation, and frequency. In practice, this results in a description across multiple frequencies and orientations. In work [6-7], the following set of parameters are used, $\theta \in \{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$, $\lambda \in \{\sqrt{2}, 2\sqrt{2}, 3\sqrt{2}, 4\sqrt{2}, 5\sqrt{2}\}$, $\varphi \in \{0, \pi/2\}$, $\sigma = \lambda$, and, finally, $\gamma = 1$. This standard configuration results in 8 orientations, 5 frequencies, and 2 phases for a total of 40 complex wavelets where each wavelet has a real and imaginary component. The magnitude of each corresponding sine and cosine is calculated and averaged to obtain 40 texture measures. For each image patch, these 40 Gabor texture measures are appended to the 10000-dimensional HSV color histogram to form a local feature vector of a total dimension of 10040, which, together with a similarity measure and a clustering algorithm, is subsequently processed to form percepts. This feature extraction method has been used successfully in [6-7] for image segmentation task and is a good starting point for our optimization.

2.3 Learning

Once a set of features is obtained, the recognition task reduces to partitioning the feature space. Each feature appears as a point in the feature space and patterns pertaining to different classes will fall into different regions in the feature space. Learning is this process of classifying a pattern into the right category. For many cases where there exists no a priori knowledge of categories into which the patterns are to be classified, the input patterns group themselves by natural association based on some properties in common. It is expected that the degree of natural association is high among members belonging to the same category and low among members belonging to different categories according to some similarity measures. As a result, patterns belonging to the same cluster should be very close together in the pattern space, while patterns in different clusters should be further apart from one another. This learning process is called unsupervised learning. In unsupervised learning, there is no class labeling available, nor do we know how many classes there are within the input patterns.

To discover similarities and dissimilarities and to reveal the organization of patterns into "sensible" clusters, a major issue is to define a "similarity" measure between two feature vectors and, after setting up an appropriate measure for it, to design an algorithm to search for similarities and dissimilarities among these patterns and then cluster the vectors on the basis of the adopted similarity measure.

It is desired that the similarity measure is given in numerical form to indicate the degree of resemblance between patterns in a group, between a pattern and a group of patterns, or between pattern groups. Many different mathematical functions have been suggested for this purpose with the most popular ones being the family of Minkowski-metrics. In a *K*-dimensional space, two commonly used special cases of this metric family are the L_1-norm and L_2-norm in mathematics, which are obtained by setting $r = 1$ or 2, respectively, in the Minkowski power metric formula,

$$dist_{i,j} = \left(\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^{r} \right)^{1/r} \quad (7)$$

where x_i and x_j are two vectors in *d*-dimensional space. Based on the similarity measure, it is expected that patterns belonging to the same cluster should be very close together in the pattern space, while patterns in different clusters should be further apart from one another.

To form perceptual groups based on partitioning the feature vectors in the feature space, unsupervised learning algorithms, also called clustering, are usually classified into several broad methods: hierarchical methods, partitioning methods, density-based methods, grid-based methods, model-based methods, and graph theoretical methods. The minimum spanning tree method is a graph analysis of arbitrary point sets of data. In a graph, two points can be connected by either a direct edge or a sequence of edges called a path. A loop in a graph is a closed path. A connected graph has one or more paths between any pair of points. A tree is a connected graph without closed loops. A spanning tree is a tree that contains every point in the data set. If a value is assigned to each edge in the tree, the tree is called a weighted tree. For example, the weights for each edge can be the distance between the two points. The weight of a tree is the total sum of edge weights in the tree. The minimum spanning tree (MST) is the spanning tree that has the minimal total weight among all possible spanning trees for the data set. The minimum spanning tree has the following property that can be used for clustering if the weight associated with each edge denotes the distance between the two points. That is, the weight associated with every edge in the minimum spanning tree will be the shortest distance between two subtrees that are connected by that edge. Therefore, removal of the longest edge will theoretically result in a two-cluster grouping. Removal of the next longest edge will result in a three-cluster grouping, and so on. These correspond to choosing breaks where maximum weights occur in the sorted edges. In this research, a popular MST-based clustering algorithm is used for image segmentation based on our color and texture features extracted [12].

3 A optimized subset of Gabor wavelets

As indicated by Equation (6), Gabor function is the product of a Gaussian function and a sinusoidal function. The sine wave is activated by frequency information in the image. The Gaussian insures that the convolution is dominated by the region of the image close to the center of the wavelet. To accurately describe the frequency information of a feature in an image, it is necessary to convolve the location with many instantiations of the wavelet. These instantiations typically sample at different frequencies and different orientations, giving rise to many coefficients from convolving with a variety of wavelets. Gabor wavelets respond to image features that are of the same orientation and frequency of the wavelets. In other words, not all instantiations contribute equally to the image frequency information and redundant filters exist. To fully understand what each of these parameters means to remove redundant components, we will look at the five parameters that control the wavelet and discuss each of the parameters in more details.

θ rotates the wavelet about its center and specifies the orientation of the wavelet. This particular set uses eight different orientations over the interval 0 to π, i.e. $\theta \in \{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$. The orientation of the wavelets dictates the angle of the edges or bars for which the wavelet will respond. This wavelet is symmetric about the origin and the convolution values will have the same magnitude but opposite sign. Orientations from π to 2π would be redundant due to the even/odd symmetry of the wavelets.

λ specifies the wavelength of the sine and cosine waves. Wavelets with a large wavelength will respond to gradual changes in intensity in the image. Wavelets with short wavelengths will respond to sharp edges and bars. This particular set uses five wavelengths, i.e. $\lambda \in \{\sqrt{2}, 2\sqrt{2}, 3\sqrt{2}, 4\sqrt{2}, 5\sqrt{2})$.

φ specifies the phase of the sine and cosine waves. The cosine wavelets are thought to be the real part of the wavelet and the sine wavelets are thought to be the imaginary part of the wavelet. Therefore, a convolution with both phases produces a complex coefficient. The mathematical foundation of the algorithm requires a complex coefficient based on two wavelets that have a phase offset of $\pi/2$, i.e. $\varphi \in \{0, \pi/2\}$.

σ specifies the radius of the Gaussian. The size of the Gaussian is sometimes referred to as the wavelet's basis of support. The Gaussian size determines the amount of the image that effects convolution. This parameter is usually proportional to the wavelength such that wavelets of different sizes and frequencies are scaled versions of each other, i.e. $\sigma = \lambda$.

γ specifies the aspect ratio of the Gaussian. This parameter is included such that the wavelets could also approximate some biological models [5][10][11]. The wavelets used here have circular Gaussians, i.e. $\gamma = 1$.

This set of parameters, that is, 8 orientations, 5 frequencies, and 2 phases, yields a total of 80 different wavelets, which are shown in Fig.3.

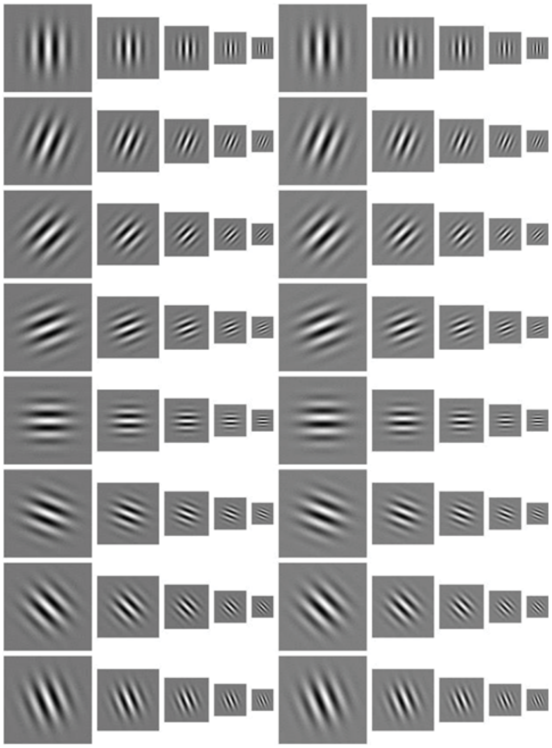


Fig. 3. Eighty Gabor wavelet masks

To find the value of a wavelet coefficient for a point in an image, the coefficient at that point is obtained by performing a convolution. The texture measure coefficients are computed by convolving a patch in the image with all 80 wavelet filters. Therefore, much of the computation of the texture measure extraction is in Gabor wavelet convolutions, and the wavelet convolutions are the dominating factor in the computational effort of the feature extraction algorithm. Fortunately, convolving with a variety of wavelets is neither efficient nor necessary. By optimally specifying the combination of spatial frequency and spatial location information, a small subset of these 80 filters may be utilized to characterize any complex stimulus and fulfill our tasks.

Since the wavelengths are determined by the pixel size of the image and they are fixed in some sense, the reduction of the number of wavelets can be realized by reducing the number of orientations, that is, θ's. There are eight orientations in the current solution. However, they are not all necessary. For the same location in an image, according to Equation (6), as minimum as four orientations can determine a Gabor wavelet. That is, one zero point, one maximal point and two other points on the left side and right side of the maximal. Based on these observations, there are nine options,

(1) $\theta \in \{\mathbf{0}, \pi/8, \mathbf{4\pi/8}, 5\pi/8\}$ (2) $\theta \in \{\mathbf{0}, \pi/8, \mathbf{4\pi/8}, 6\pi/8\}$
(3) $\theta \in \{\mathbf{0}, \pi/8, \mathbf{4\pi/8}, 7\pi/8\}$ (4) $\theta \in \{\mathbf{0}, 2\pi/8, \mathbf{4\pi/8}, 5\pi/8\}$
(5) $\theta \in \{\mathbf{0}, 2\pi/8, \mathbf{4\pi/8}, 6\pi/8\}$ (6) $\theta \in \{\mathbf{0}, 2\pi/8, \mathbf{4\pi/8}, 7\pi/8\}$
(7) $\theta \in \{\mathbf{0}, 3\pi/8, \mathbf{4\pi/8}, 5\pi/8\}$ (8) $\theta \in \{\mathbf{0}, 3\pi/8, \mathbf{4\pi/8}, 6\pi/8\}$
(9) $\theta \in \{\mathbf{0}, 3\pi/8, \mathbf{4\pi/8}, 7\pi/8\}$

By this optimization, we can reduce the texture feature extraction time by using one of the above nine options to minimize the number of Gabor wavelet convolutions so as to consume only a part of the original total time.

4 Experiments

In this section, we present the results of experiments conducted to evaluate the performance of our proposed method. The method was tested by performing a segmentation task on two images using the original 80 Gabor filters and the proposed reduced set of 40 Gabor filters. We implemented all the algorithms in C++. All the experiments were performed on a computer with Intel Core i3-2350 2.30 GHz CPU and 4 GB RAM. The operating system running on this computer is Windows 7. We use the timer utilities defined in the C standard library to report the CPU time. The experimental results are presented in Fig.4 and Table 1.

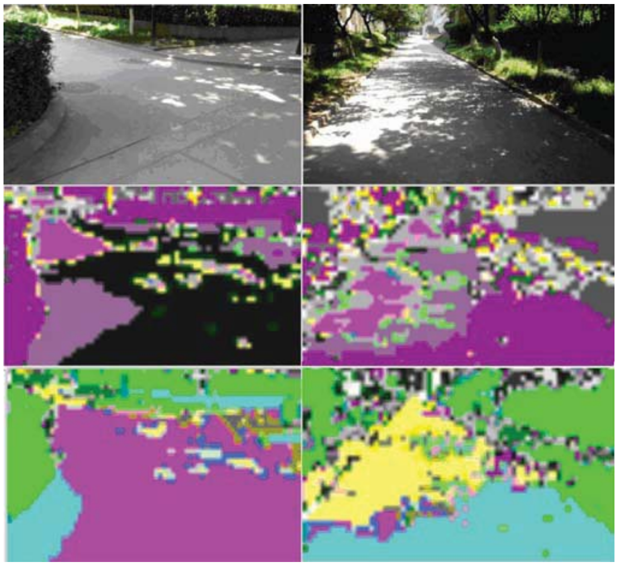


Fig. 4. Image segmentation results on two images.

By visualizing the experimental results of segmenting two test images using the nine reduced subsets of orientation for Gabor filters one by one, we find out that the best option is the fifth one, i.e., $\theta \epsilon \{\mathbf{0}, 2\pi/8, \mathbf{4\pi/8}, 6\pi/8\}$. In Fig.4, the images on the first line are the original images. The images on the second line are the segmentation results using the original 40 Gabor texture measures. The images on the third line are the segmentation results using the reduced set of 20 Gabor texture measures. From the figure, it is clearly seen that our optimized approach has obtained much better performance in the segmentation quality in comparison with the original approach.

Table 1. Running time performances in seconds.

Image	80 Gabor filters	40 Gabor filters
Image 1	117s	64s
Image 2	140s	56s

The running time performances of the newly proposed approach and the original approach are summarized in Table 1. There are two columns in the table, representing the running time performance of Gabor texture measure extraction using the original approach, i.e., the first column, and our proposed approach, i.e., the second column, respectively. From the table, we can see that our proposed approach outperforms the original approach by a factor of 2 in running time performance. Unfortunately, there is no obvious way to know which of the nine options is the best in practice, but to find the optimum through trial and error at current time. We will leave it for our future work.

5 Conclusions

To identify targets, image segmentation techniques partition an image into nonoverlapping but meaningful regions based low-level features such as color, texture measures and shapes etc.. Being a mathematical approximation to the spatial receptive field of a simple cell in the *V*1 area of human brain, Gabor wavelets are often used as to extract texture features. The problem with these Gabor texture measures is the high computational cost consumed in the convolution calculation of the feature extraction process. To partially solve this problem, in this paper, we carefully study the behaviors of the Gabor wavelets used to form the texture features in the previous work and find out that only a small subset of the filters has important contributions to the identification process. Experimental results show that, by an optimized subset of Gabor filters, better performance in terms of effectiveness can be achieved in a much shorter time. In future work, we will focus on how to choose the small optimized subset of the filters in an automatic way without human experts' intervention.

Acknowledgments

The authors would like to thank Chinese National Science Foundation for its valuable support of this work under award 61473220 and all the reviewers for their valuable comments.

References

[1] T. Fukuda, R. Michelini, V. Potkonjak, S. Tzafestas, K. Valavanis, and M. Vukobratovic. “How far away is ‘artificial man’?”. IEEE Robotics & Automation Magazine, 8(1):66-73, March 2001.

[2] B.A. Olshausen and D.J. Field. “Emergence of simple-cell receptive field properties by learning a sparse code for natural images”. Nature, 381:607-609, June 1996.

[3] R. Goldstone. “Perceptual learning”. Annual Review of Psychology, 49:585-612, Feb 1998.

[4] J.M. Buhmann, T. Lange, and U. Ramacher. “Image segmentation by networks of spiking neurons”. Neural Computation, 17(5):1010-1031, March 2006.

[5] J.E. Hunter. “Human motion segmentation and object recognition using fuzzy rules”. In Proceedings of the 14th Annual IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2005), Nashville, TN, pp.210-216, Aug 2005.

[6] X. Wang, M. Tugcu, J.E. Hunter, and D.M. Wilkes, “Exploration of configural representation in landmark learning using working memory toolkit”. Pattern Recognition Letters, 30(1):66-79, Jan 2009.

[7] J.E. Hunter, M. Tugcu, X. Wang, C. Costello, and D. Mitch Wilkes. “Exploiting sparse representations in very high-dimensional feature spaces obtained from patch based processing”. Machine Vision and Applications, 22(3):449-460, April 2011.

[8] D. Gabor. “Theory of communications”. Journal of Institute of Electrical Engineering, 93(26):429-457, Nov 1946.

[9] M.J. Swain and D.H. Ballard. “Color indexing”. International Journal of Computer Vision, 7(1):11-32, Nov 1991.

[10] F.C.N. Pereira, N. Tishby and L. Lee. “Distributional clustering of English words”. In Proceeding of the 31st Meeting of the Association for Computational Linguistics, Columbus, OH, pp.183-190, 1993.

[11] N. Petkov and P. Kruizinga. “Computational models of visual neurons specialised in the detection of periodic and a periodic oriented visual stimuli: Bar and grating cells”. Biol Cybern, 76(2):83–96, Feb 1997.

[12] X. Wang, X.L. Wang, and D.M. Wilkes. “A divide-and-conquer approach for minimum spanning tree-based clustering”. IEEE Transactions on Knowledge and Data Engineering, 21(7):945–958, July 2009.

Nano Particles Size Measurement Based on the Partial Differential Equation

Fang Zhang[1,2], Ping Liu[1,2], Zhitao Xiao*[1,2], Lei Geng[1,2], Jun Wu[1,2], Ying Chen[1,2], Meng Wang[1,2], and Hongxia Tian[1,2]

[1] School of Electronics and Information Engineering, Tianjin Polytechnic University, Tianjin, 300387, China
[2] Tianjin Key Laboratory of Optoelectronic Detection Technology and System, Tianjin, China

Abstract - *The optical and magnetic properties of Nanoparticles (NPs) are highly related to the sizes and shapes of the NPs. Applying image processing technology to analyze and process the transmission electron microscopy (TEM) image of NPs can effectively enhance the efficiency and accuracy of the size measurement of NPs. In this paper, partial differential equation (PDE) methods are used for filtering and segmenting NP images. Based on the segmentation results, size measurement and evaluation are achieved automatically. The experimental results show that the proposed PDE-based NP size measurement method is effective.*

Keywords: Nanoparticles; Partial Differential Equation Image Processing Method; Filtering; Level Set Image Segmentation; Size Measurement

1 Introduction

Nanotechnology is widely applied in catalytic science, medicine, new materials, electric power, composite materials industries, etc., which plays an important role in many high-tech areas [1]. Since the properties of materials synthesized with Nanoparticles (NPs) are highly related to the sizes and shapes of the NPs, the characterization of the microstructure of nano materials is important for recognizing their characteristics and application areas and promoting their development. The NP size measurement is one of the key technologies. Existing techniques for measuring the size of NPs include the dynamic light scattering method, the X-ray diffraction line width method, the X-ray small angle scattering method and the transmission electron microscope (TEM) method [2]. Among those methods, the TEM method, which enables the observation of the size distribution and shape of NPs, has shown to be reliable.

Processing and analyzing NP images based on image processing technology is an important method of the NP size measurement, a key step of which is the segmentation of individual particles. Due to the nonuniform gray levels of NPs and the weak edges of certain particles, it is critical segment individual particles accurately. In recent years, a branch of the image processing methods based on the partial differential equations (PDE), i.e., the level set image segmentation method [3], has attracted significant research interests. With this method, the edge evolution curve is implicitly represented as the zero level set of a higher-dimensional function. The level set function is evolved under the control of PDE, until the zero level set evolves to the object boundary of the images. This evolution has many advantages. For example, it can automatically and flexibly deal with the change of the zero level set topology (such as breaking, merging). It can also effectively split weak edges of the target. Li et al. [4] proposed the DRLSE model, and Chan and Vese [5] proposed the CV model. However, these two segmentation models are not ideal for images with weak edges. Later, Li et al. [6-7] proposed the local region scalable fitting (RSF) model, where the local area information is embedded into the regional variational level set, which is then used to drive the evolution of the curve. This way good segmentation is achieved. In this paper, we adopt the level set image segmentation method to segmenting the NPs accurately.

In this paper, we study the accurate measurement of spherical NPs in TEM images, including the accurate measurement of the diameter and sphericity. We also measure other parameters such as area and perimeter, study their statistics, and evaluate the uniformity of the NPs. The results provide useful references for the preparation of NPs and for studying the characteristics of the particles.

2 Size parameters measurement method of NPs

Image processing-based processing and analysis of NPs mainly involves five steps, i.e., image preprocessing, individual segmentation of NPs, edge fitting, pixel calibration and parameter measurement. The block diagram of the proposed method in this paper is shown in Fig. 1.

2.1 Image filtering based on the PDE of NPs

NP images have the characteristics of weak edges and strong noise. In this paper, we use the PDE filtering method [8] for image de-noising, which can suppress noise and maintain clear edges simultaneously.

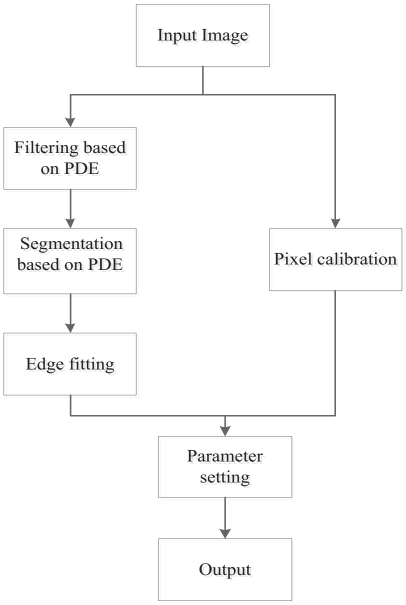


Figure 1. Parameters measurement process of NPs

The PM model proposed by Perona and Malik [9] uses a nonlinear diffusion equation that controls the diffusion rate by gradient:

$$\begin{cases}\partial_t u = div[c(|\nabla u(x,y,t)|)] \\ u(x,y,0) = I(x,y)\end{cases} \tag{1}$$

where div is the divergence operator, ∇ is the gradient operator,

$$c(|\nabla u|) = \frac{1}{1+\left(|\nabla u|/k\right)^2} \tag{2}$$

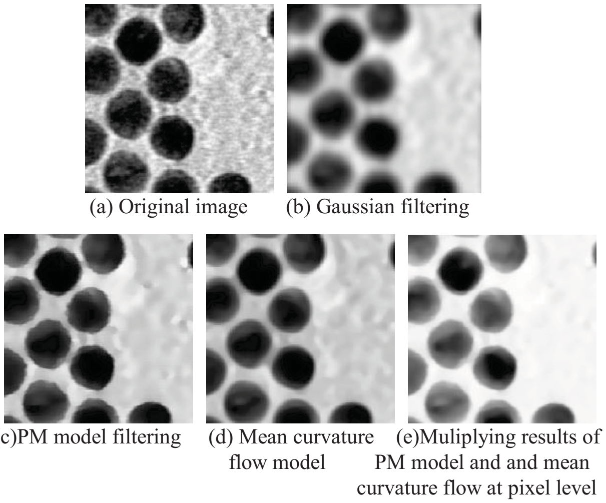
and k is a constant. Because $c(\cdot)$ is a decreasing function of the gradient, the intensity of the diffusion is stronger where the image gradient is smaller and weaker where the image gradient is greater. Consequently, the PM model can maintain the image boundaries while diffusing.

Alvarez proposed the mean curvature flow model [10], which can effectively solve the problem of edge blurring caused by the diffusion via controlling the diffusion direction of the equation. This is achieved by avoiding diffusion in the vertical direction of the image edge

$$\begin{cases}\partial_t u = |\nabla u|\, div[\frac{\nabla u}{|\nabla u|}] \\ u(x,y,0) = I(x,y)\end{cases} \tag{3}$$

where $|\nabla u| div[\frac{\nabla u}{|\nabla u|}]$ is a second derivative for u along the edge direction.

The filtering results of the spherical NPs are shown in Fig. 2. Fig. 2(a) shows the spherical NPs. Fig. 2(b)-(d) show the results of Gaussian filtering, the PM model based on PDE, and the mean curvature flow model, respectively. From Fig. 2(b), Gaussian filtering can remove the noise, but the particle edges also become fuzzy. Fig. 2(c) shows that the PM model is sensitive to isolated noise, stops spread at the noise points with great gradients, and the noise points are retained. From Fig. 2(d), when the local noises were concentrated, the mean curvature flow model treats those noises as smooth areas and retains them, so a "block" effect appears. However, the mean curvature flow model keeps the edges. As shown in Fig. 2(e), multiplying the results of the mean curvature flow model and the PM model at pixel level can effectively remove the influence of strong noise and highlight the real edge of particles.


(a) Original image (b) Gaussian filtering

(c)PM model filtering (d) Mean curvature flow model (e)Muliplying results of PM model and and mean curvature flow at pixel level

Figure 2. Comparison of the filtering results of the spherical NPs

2.2 Image segmentation based on the PDE of NPs

The segmentation of NPs is critical for processing and analyzing particle images. The results directly affect the accuracy of the fitting measurement and the statistical analysis. Due to the non-uniform gray levels of the filtered results of NPs and weak edges, the local region scalable fitting (RSF) level set model is used to segment the NPs accurately and completely. The geometric shapes of the particles can then be restored by fitting the segmented particles.

Let Ω be the image domain, and $I:\Omega \rightarrow R$ be a given gray level image. C denotes a closed contour in the image domain Ω, which separates Ω into two regions: the inside region Ω_1 and the outside region Ω_2. For a given point $x \in \Omega$, it has a circular neighborhood of radius ρ, known as $O_x = \{y : x - y < \rho\}$, so the energy function in the neighborhood of each point is:

$$\varepsilon\begin{pmatrix}\phi, f_1(x),\\ f_2(x)\end{pmatrix} = \lambda_1 \int_\Omega \left(\int_\Omega \begin{matrix} K(x\text{-}y)\left|I(y)\text{-}f_1(x)\right|^2 \\ H(f(y))dy \end{matrix} \right) dx + \lambda_2 \int_\Omega \left(\int_\Omega \begin{matrix} K(x\text{-}y)\left|I(y)\text{-}f_2(x)\right|^2 \\ (1\text{-}H(f(y)))dy \end{matrix} \right) dx + \nu \int_\Omega \delta(\phi(x))\left|\nabla\phi(x)\right| dx + \mu \int_\Omega \frac{1}{2}\left(\left|\nabla\phi\right|-1\right)^2 dx \tag{4}$$

where $K(x-y)$ is a Gaussian kernel function, ϕ is the level set function that employs a signed distance function [11], $H(\cdot)$ is the Heaviside function [12],

$$f_1(x) = \frac{K(x-y)*\left[H(\phi(y))I(y)\right]}{K(x-y)*H(\phi(y))} \tag{5}$$

$$f_2(x) = \frac{K(x-y)*\left[1-H(\phi(y))I(y)\right]}{K(x-y)*\left[1-H(\phi(y))\right]} \tag{6}$$

At the edge of the target, $f_1(x)$ and $f_2(x)$ are close to the intensities of the internal and external image contours, and the energy function is minimized.

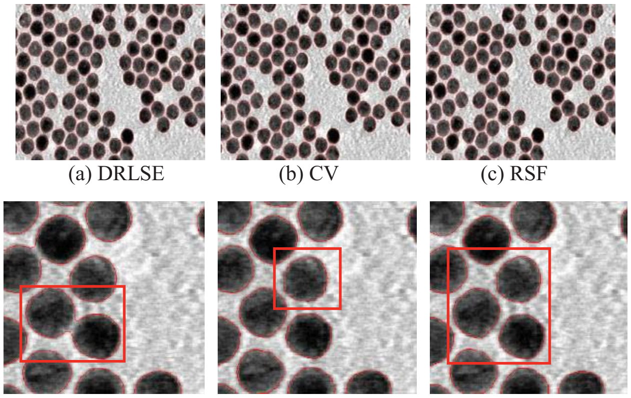
The Euler-Lagrange method is used to minimize (4). Then according to the gradient descent flow, the partial differential equations can be obtained to describe the evolution of the level set function:

$$\frac{\partial\phi}{\partial t} = -\delta(\phi)(\lambda_1 e_1 - \lambda_2 e_2) + \nu\delta(\phi)div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \mu\left(\nabla^2\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right) \tag{7}$$

where $e_i(y) = \int K_\sigma(y-x)\left|I(y)-f_i(x)\right|^2 dx$

Since the RSF model fits mainly in local areas, it can give good segmentation results for images with fuzzy edges and uneven strengths.

In the experiment of NPs segmentation, the initial contour determined by binarization is evolved using the DRLSE model [4], CV model [5] and RSF model, respectively. The DRLSE model segmentation results for spherical NPs are shown in Fig. 3(a) and Fig. 3(d). From the partial enlarged detail of Fig. 3(d), we can see that the DRLSE model traps in local minima and leads to particle adhesion. The CV model results are shown in Fig. 3 (b) and Fig. 3 (e). From Fig. 3(e) we can see that the edge location of the particles is not exact and the boundary of the curve is incorrectly located in the interior of the particles. The results in Fig. 3 (c) and Fig. 3 (f) show that the RSF model can locate the exact edges.



(a) DRLSE (b) CV (c) RSF

(d) the partial enlarged detail of fig (a) (e) the partial enlarged detail of fig (b) (f) the partial enlarged detail of fig (c)

Figure 3. Comparison of the level set segmentation results of the spherical NPs

2.3 Pixel calibration

In the process of image size measurement, the pixel size of the image needs to be transformed into the actual geometry size of the object. Thus we need to calculate the real size that a pixel represents in the image to be measured. The scale of TEM images has some characteristics, such as that the grey value of the scale approaches zero, the scale is around the left corner of the image, and the shape of the scale is an elongated rectangle. We extract the ruler and calibrate the image in this paper according to these characteristics. We first extract the scale part in the lower left corner of the image as region of interest, obtain and mark the binary image. We then find the the largest rectangle as the scale area. According to the actual length L and the pixel number N of the scale long side, the actual size of each pixel is calculated as $k = L/N$. In this paper, the length of the scale is 576 pixels, the actual length is 100 nm, so k = 0.1736 nm/pixel.

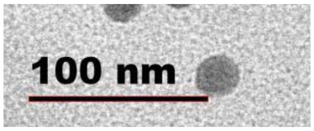


Figure 4. Result of the scale region detection

2.4 Particle parameter measurement

The sizes and shapes of the NPs directly affect the properties of the prepared particles, and the parameter measurement of the particles is the most important task for evaluating the quality of the particles. The particle parameters include the parameters characterizing the size (perimeter, area and particle diameter) and those characterizing the shape (sphericity and convexity).

(1) perimeter

The perimeter is an important parameter of the particles, which generally refers to the length of the particle boundary.

(2) area

The area of a particle is obtained according to the boundary of the particle and the number of pixels in the boundary.

(3) particle diameter

The particle diameter is the most important parameter characterizing the particle size. The size of a regularly shaped particle can be characterized by certain length variable, or one or more characteristic parameters.

(4) sphericity

The sphericity originally refers to the ratio of the surface area and volume of the 3-D object. In order to describe the 2-D object, it is defined as:

$$S = \frac{r_i}{r_c} \tag{8}$$

where r_c is the circumradius of the object, r_i is the inscribed circle radius. When the object is round, the sphericity reaches the maximum (S=1). When the object has other shapes, S<1, as shown in Fig. 6. The sphericity is not affected by translation and rotation of the region and scale change.

(5) convexity

The convexity of an object is calculated by

$$C_{conv} = \frac{A}{A_{conv}}, \tag{9}$$

where A is the area of the object, and A_{conv} is the convex hull area of the object. When the object is convex, the value of C_{conv} is 1. When the object is concave or has hole, the value of C_{conv} is less than 1 [13]. This paper aims to evaluate the uniformity of the particles by measuring the relative parameters of the particles. Thus we target complete and non-adherent particles in this paper. By using the convexity of the target, the adherent particles can be excluded. An empirical threshold value 0.96 is chosen, as shown in Fig. 7.

We obtain the diameters of the spherical NPs by fitting the particle boundaries as circles using least squares, as shown in Fig. 5 (a). We also obtain the circumcircle and incircle, whose diameters can be used for calculating the sphericity, as shown in Fig. 5(b).

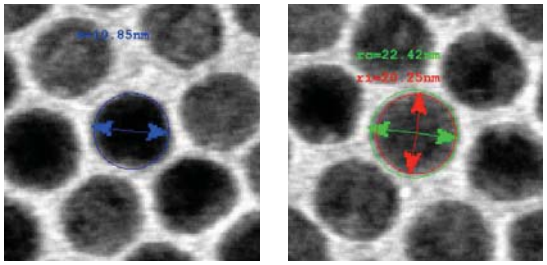


Figure 5. Fitting measurement results of the spherical NPs

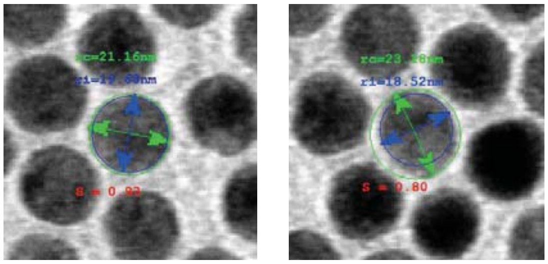


Figure 6. The sphericity measurement results of the spherical NPs

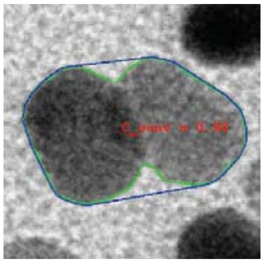


Figure 7. The adherent NPs

3 Size parameters measurement method of NPs

In this paper, we measure the NP parameters based on preprocessing, particle segmentation and boundary fitting. This section analyzes the measurement results and evaluates the particle uniformity by the measured parameters.

3.1 Parameter measurement results analysis

In this paper, we use the fitted circle diameter of the particle boundary as the diameter of the spherical particles to represent the particle size. In addition, we use the incircle diameter of the particle as the shortest diameter of the

Table 1 THE LONGEST DIAMETER MEASUREMENT RESULTS OF SOME SPHERICAL NPS

Number	Manual measurement /nm						Our method	Absolute error	Relative error
	1	*2*	*3*	*4*	*5*	*average*	*/nm*	*/nm*	
1	23.82	24.18	24.08	23.86	24.39	24.06	23.49	0.57	2.39%
2	21.96	21.97	21.91	21.89	21.10	21.76	21.37	0.39	1.81%
3	22.01	22.17	22.29	21.78	21.57	21.96	21.60	0.36	1.65%
4	22.30	21.87	21.30	22.09	22.40	21.99	21.66	0.33	1.50%
5	22.84	23.02	22.17	23.06	23.00	22.82	22.48	0.34	1.48%

6	22.36	22.68	21.86	21.94	21.90	22.15	22.02	0.13	0.57%
7	21.03	21.83	21.93	21.04	20.95	21.36	21.28	0.08	0.36%
8	23.06	23.00	22.84	22.53	23.18	22.92	22.68	0.24	1.05%
9	22.35	21.79	22.22	21.77	21.76	21.98	21.60	0.38	1.72%
10	22.78	22.15	22.03	21.98	22.46	22.28	22.32	0.04	0.18%

spherical NPs, and the circumscribed circle diameter of the particles as the longest diameter of the spherical NPs to calculate the sphericity of the NPs. To evaluate the accuracy, we get the manual measurement results by Image J software as the standard. To reduce the random errors of the manual measurement, we use the mean value of five measurement results is used. Table 1 shows part of the measurement results.

Table 2 THE PARAMETER MEASUREMENT RESULTS OF SOME SPHERICAL NPS

Number	Fitted diameter	The longest diameter	The shortest diameter	Sphericity *S*	Area *A*	Perimeter *B*
	/nm	*/nm*	*/nm*		*/nm^2*	*/nm*
1	21.98	23.49	19.68	0.84	389.15	72.71
2	21.15	23.14	19.10	0.83	365.03	69.91
3	21.93	22.48	21.41	0.95	393.17	73.19
The standard deviation of 100 particles	0.98	1.21	0.94	0.036	29.22	3.25

We measured the diameters of 100 spherical NPs. Part of the measurement results for other parameters of the particles are shown in Table 2. The statistical results show that the average relative error of the longest diameter between the calculated results and the manual measured results was 1.98%. The relative error is less than 3% for 71% of the particles and less than 5% for 94% of the particles. Thus, the measurement results are accurate.

To summarize, the proposed NP measurement method based on PDE can achieve automatic measurement of the spherical NPs size with a high accuracy.

3.2 Uniformity evaluation

The uniformity of particles has important influence on the nano materials. The full width at half maximum (FWHM) of particle size is used to evaluate the uniformity of particles. Based on the measurement results, we first draw the histogram of the parameters distribution, which is generally Gaussian-like. The FWHM, as illustrated in Fig. 8, is the difference between the two points that have a frequency equal to half the peak frequency. A smaller FWHM of the particle size indicates more uniform particle sizes and better effect of preparation.

Fig.9 shows the statistical distribution of the fitted diameter of spherical NPs. The measured FWHM is 1.95 nm for the No. 008 spherical NPs, and 2.56 nm for the No. 009 spherical NPs. Based on the results, we can conclude that the No. 008 spherical NPs are more uniform than the No. 009 spherical NPs. Through the histogram analysis, we can evaluate and analyze the preparation of the NPs, and study the physical and chemical properties of the NPs.

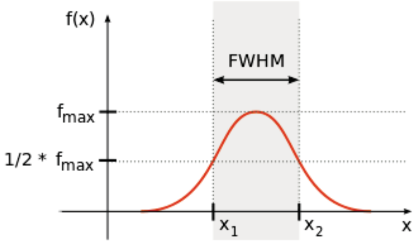
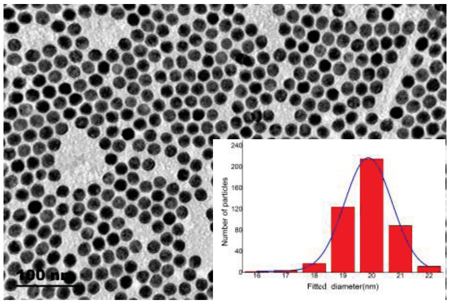
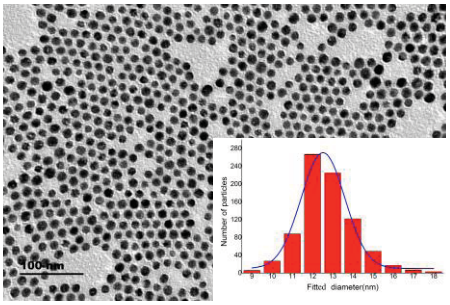


Figure 8. FWHM schematic



(a) No. 008 of the spherical NPs and its fitted diameter histogram, the statistical result is 19.89±0.98nm
(The center value of the diagonal is 19.89nm, the FWHM is 0.98nm)



(b) No. 009 of the spherical NPs and its fitted diameter histogram, the statistical result is 12.52±1.28nm (The center value of the diagonal is 12.52nm, the FWHM is 1.28nm)

Fig.9. The histogram of the fitted diameter statistical results of the spherical NPs

4 Conclusions

The size and shape parameters of NPs can be used to evaluate the particle uniformity of the preparation process and are related to the other properties of the particles. This paper proposes an automatic size measurement method of the NPs based on PDE. Firstly, the NP images are smoothed by the PDE filtering method. Then the level set segmentation model is used to segment the NPs and the edges of the particles are fitted. Finally, the size and shape parameters of the NPs are calculated after calibration. The experimental results show that the method in this paper can effectively measure the sizes of NPs.

5 Acknowledgment

This work is sponsored by Tianjin Science and Technology Supporting Projection under grant No. 14ZCZDGX00033, Tianjin Research Program of Application Foundation and Advanced Technology under grant No.15JCYBJC16600, and Specialized Research Fund for the Doctoral Program of Higher Education of China under grant No. 20131201110001.

6 References

[1] C.E. Fowler, D. Khushalani, B. Lebeau, S. Mann. “Nanoscale Materials with Mesostructured Interiors”, Advanced Materials, Vol. 13 No. 9, pp. 649-652, 2001.

[2] P. Bowen. “Particle Size Distribution Measurement from Millimeters to Nanometers and from Rods to Platelets”, Journal of Dispersion Science & Technology, Vol. 13 No. 5, pp. 631-662, 2002.

[3] Z. Kaihua, Z. Lei, L. Kin-Man, Z. David. “A Level Set Approach to Image Segmentation With Intensity Inhomogeneity”, IEEE Transactions on Cybernetics, Vol. 46 No. 2, pp. 546-557, 2016.

[4] C.M. Li, C.Y. Xu, C.F. Gui, et al. “Distance regularized level set evolution and its application to image segmentation”, IEEE Transactions on Image Processing, Vol. 19 No. 12, pp. 3243-3254, 2010.

[5] T. Chan, L. Vese. “Active contours without edges”, IEEE Transactions on Image Processing, Vol. 10 No. 2, pp. 266-277, 2001.

[6] C.M. Li, C.Y. Kao, J.C. Gore, et al. “Implicit active contours driven by local binary fittingenergy”, IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, pp. 339-345, 2007.

[7] C.M. Li, C.Y. Kao, J.C. Gore, et al. “Minimization of region-scalable fitting energy for image segmentation”, IEEE Transactions on Image Processing, Vol. 17 No. 10, pp. 1940-1949, 2008.

[8] D. Gabor. “Information theory in electron microscopy”, Laboratory Investigation, Vol. 14, pp. 801-807, 1965.

[9] P. Perona, J. Malik. “Scale-space and edge detection using anisotropic diffusion”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12 No. 7, pp. 629-639, 1990.

[10]L. Alvarez. “Image selective smoothing and edge detection by nonlinear diffusion”, SIAM Journal on Numerical Analysis, Vol. 29 No. 3, pp. 845-866, 1992.

[11]W. Dejun, Z. Jiali, et al. “Level set methods, distance function and image segmentation”, Proceedings of the 17th International Conference on Pattern Recognition, pp. 110-115, 2004.

[12]Y.S. Sun, P. Li, B.Y. Wu. “An Improved Approach to Image Segmentation Based on Mumford-Shah Model”, Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, pp. 3996-4001, 2006.

[13] C. Steger, M. Ulrich, C. Wiedemann. “Machine Vision Algorithms and Applications”. Germany: Wiley-VCH, Weinheim, 2007.

Volumetric Segmentation

Dumitru Dan Burdescu, Marius Brezovan, Liana Stănescu,
Cosmin Stoica Spahiu, Daniel Costin Ebâncă
Computers and Information Technology Department,
Faculty of Automatics, Computers and Electronics,
University of Craiova, Dolj, Romania

Abstract - *Image segmentation plays a crucial role in effective understanding of digital images, planar or volumetric images. The current research in graph based methods is oriented towards producing approximate solution (or sub-optimal solution) for such graph matching problem to reduce processing time. We are introducing an algorithm for volumetric segmentation based on virtual tree-hexagonal structure (prisms) constructed on the image voxels to improve the speed of segmentation. Here, a graph-based theoretic framework is considered by modeling image segmentation as a graph partitioning problem using input spatial graph. Then we can use the graph facilities and their related algorithms and computational complexity can be viewed as slow as the fundamental graph algorithms. The key to the whole algorithms of volumetric segmentation method is the prism cells as vertices. Volumetric segmentation algorithm contains many other algorithms but only segmentation algorithm is presented based on the limited space of paper.*

Keywords: Graph Based Segmentation, Color Based Segmentation, Syntactic-based Segmentation, Spatial Graph Algorithms, Dissimilarity

1 Introduction and related work

Higher-level problems such as object recognition and image indexing can also make use of segmentation results in matching to address problems such as figure-ground separation and recognition by parts.

In our paper, we develop a visual feature-based method which uses a spatial graph constructed on cells of prism with tree-hexagonal structure containing less than half of the image voxels. Thus, the volumetric image segmentation is treated as a spatial graph partitioning problem.

It was determined the normalized weight of an edge by using the smallest weight incident on the vertices touching that edge [1]. Other methods for planar images [2], [3] use adaptive criterion that depends on local properties rather than global ones. In contrast with the simple graph-based methods, cut criterion methods capture the non-local cuts in a graph, beeing designed to minimize the similarity between pixels that are being split [4], [5]. The normalized cut criterion [5] takes into consideration self similarity of regions. An alternative to the graph cut approach is to look for cycles in a graph embedded in the image plane. In [6] and [7] the quality of each cycle is normalized in a way that is closely related to the normalized cut approaches. Other approaches to digital planar image segmentation consist in splitting and merging regions according to how well each region fulfills some uniformity criterion. Such methods [8] use a measure of uniformity of a region. In contrast [2] and [3] use a pair-wise region comparison rather than applying a uniformity criterion to each individual region. Complex organizing phenomena can emerge from simple computation on these local cues [9]. A number of approaches to segmentation are based on finding compact regions in some feature space [10]. Our previous works for planar segmentation algorithms [11], [12], [13] and [14] are related to the works in [2] and [3] in the sense of pair-wise comparison of region similarity. In this paper, we are using new algorithms for planar segmentation based on graphs. Our work for planar segmentation consists in adding new steps for volumetric segmentation algorithms that allow us to determine regions closer to it [15], [16], [17] and [18].

2 Virtual tree-hexagonal structure

Volumetric segmentation module creates virtual cells of prisms with tree-hexagonal structure defined on the set of the digital image voxels of the input RGB volumetric image and a spatial grid graph having tree-hexagons (prism) as cells of vertices. In addition for each component, the dominant color of the region is extracted. The surface extraction module determines for each segment of the image its boundaries. The boundaries of the determined visual objects are closed surfaces represented by a sequence of adjacent tree-hexagons. At this level, a linked list of voxels representing the surface is added to each determined component. This implies that there will be less ambiguity in defining volumetric surface and volumes [16], [17], [18].

Let I be an initial volumetric image having the three dimensions h × w × z (e.g. a matrix having 'h' rows, 'w' columns and 'z' deep of matrix voxels). To construct a tree-hexagonal grid (prism cells) on these voxels we retain an eventually smaller image with:

$$\begin{gathered} h = h - (h - 1) \bmod 2; \\ w = w - w \bmod 4; \\ z = z \end{gathered} \tag{1}$$

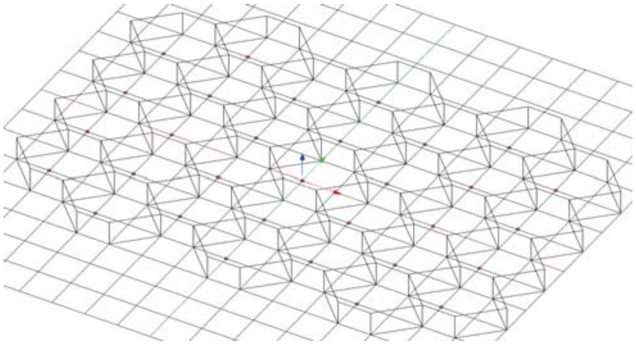


Fig. 1. Virtual Tree-Hexagonal structure constructed on the input digital image voxels.

Each tree-hexagon from the tree-hexagonal grid contains sixteen voxels: twelve voxels from the frontier and four interior voxels. We select always the left up voxel from the four interior voxels to represent with approximation the gravity center of the tree-hexagon, denoted by the pseudo-gravity center. We use a simple scheme of addressing for the tree-hexagons of the tree-hexagonal grid that encodes the volumetric location of the pseudo-gravity centers of the tree-hexagons as presented in Figure 1. Each tree-hexagon (prism) represents an elementary item and the entire virtual tree-hexagonal structure represents a spatial grid graph, G = (V; E), where each tree-hexagon H in this structure has a corresponding vertex v $\in$ V . The set E of edges is constructed by connecting tree-hexagons that are neighbors in a 8-connected sense. The vertices of this graph correspond to the pseudo-gravity centers of the hexagons from the tree-hexagonal grid and the edges are straight lines connecting the pseudo-gravity centers of the neighboring hexagons, as presented in Figure 2.

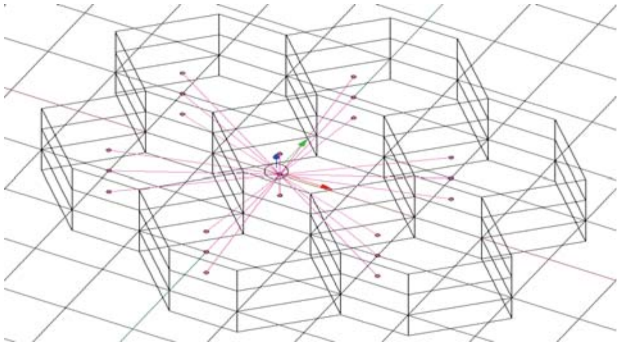


Fig. 2. The grid graph constructed on the pseudo-gravity centers of the tree-hexagonal grid.

We associate to each tree-hexagon H from V two important attributes representing its dominant color and the coordinates of its pseudo-gravity center, denoted by c(h) and g(h). The dominant color of a tree-hexagon is denoted by c(h) and it represents the color of the voxel of the tree-hexagon which has the minimum sum of colors distance to the other twenty voxels. Each tree-hexagon H in the tree-hexagonal grid is thus represented by a single point, g(h), having the color c(h). By using the values g(h) and c(h) for each tree-hexagon, information related to all voxels from the RGB initial image is taken into consideration by the spatial segmentation algorithm.

3 Volumetric segmentation algorithm

Let V = $\{h_1, \ldots., h_{|V|}\}$ be the set of tree-hexagons (prisms) constructed on the volumetric image voxels as presented in previous section and G = (V; E) be the undirected spatial grid-graph, with E containing pairs of prism cells (tree-hexagons) that are neighbors in a 16-connected sense. Components of an input digital image represent compact volumes containing voxels with similar properties. The set V of vertices of the graph G is partitioned into disjoint sets, each subset representing a distinct visual object of the initial image.

Definition 1. Let G = (V; E) be the undirected spatial graph constructed on the tree-hexagonal structure of an input digital image, with V = $\{h_1, \ldots, h_{|V|}\}$. A proper segmentation of V, is a partition S of V such that there exists a sequence $\langle S_i; S_{i+1}; : : : ; S_{f-1}; S_f \rangle$ of segmentations of V for which:

- S_f is the final segmentation and S_i is the initial segmentation,
- S_j is a proper refinement of S_{j+1} (i.e., $S_j \subset S_{j+1}$) for each j = i . . . . f − 1,
- segmentation S_j is too fine, for each j = i; : : : ; f − 1,
- any segmentation S_l such that $S_f \subset S_l$, is too coarse,
- segmentation S_f is neither too coarse nor too fine.

Our volumetric segmentation algorithm starts with the most refined segmentation, $S_0 = \{\{h_1\}. \ldots . \{h_{|V|}\}\}$ and it constructs a sequence of segmentations until a proper segmentation is achieved. Each segmentation Sj is obtained from the segmentation S_{j-1} by merging two or more connected components for which there is no evidence of a boundary between them. For each component of a volumetric segmentation, a spanning tree is constructed and thus for each segmentation algorithm we use an associated spanning forest. The evidence of a boundary between two components is determined by taking into consideration some features in some model of the digital image. When starting, for a certain number of segmentation components the only considered feature is the color of the volumes associated to the components and in this case we use a color based region model. When the components become complex and contain too much tree-hexagons, the color model is not sufficient and geometric features together with color information are considered. In this case we use a syntactic-based with a color-based region model for volumes. In addition, syntactic features bring supplementary information for merging similar volumes in order to determine objects. In the color of graph-based model, the volumes are modeled by a vector in the RGB color space. This vector is the mean color value of the dominant color of tree-hexagons belonging to the regions. The evidence for a volumetric surface between two volumes is based on the difference between the internal contrast of volumes and the external contrast between them [2], [16] and [18]. Both notions of internal contrast and external contrast between two volumes are based on the dissimilarity between two colors.

Definition 2. Let G = (V; E) be the undirected spatial graph constructed on the tree-hexagonal structure of a volumetric input image and S a color-based segmentation of V. The segmentation S is too fine in the color-based region model if there is a pair of components (C′; C″) ∈ S for which *adjacent(C'; C'') = true* and *ExtVar(C'; C'') = IntVar(C'; C'') + tresh(C';C'')*, where the adaptive threshold tresh(C′; C″) is given by

$$tresh(C'; C'') = tresh/(min(|C'|; |C''|). \qquad (2)$$

The threshold 'tresh' is a global adaptive value defined by using a statistical model.

The maximum internal contrast between two components, (C′; C″) ∈ S is defined as follows:

$$IntVar(C'; C'') = max\{IntVar(C'); IntVar(C'')\}.$$

Algorithm 1 Volumetric Segmentation Algorithm

```
procedure SEGMENTATION(l, c, d, P, H, Comp)
    Input l, c, d, P
    Output H, Comp
    H ← CREATEHEXAGONALSTRUCTURE(l, c, d, P)
    G ← CREATEINITIALGRAPH(l, c, d, P, H)
    CREATECOLORPARTITION(G, H, Bound)
    G′ ← EXTRACTGRAPH(G, Bound, tresh)
    CREATESYNTACTICPARTITION(G, G′, tresh)
    Comp ←EXTRACTFINALCOMPONENTS(G′)
end procedure
```

The input parameters represent the digital image resulted after the pre-processing operation: the array P of the volumetric image voxels structured in 'l' lines, 'c' columns and 'd' depths. The output parameters of the segmentation procedure will be used by the surface extraction procedure: the tree-hexagonal grid stored in the array of tree-hexagons H, and the array Comp representing the set of determined components associated to the objects in the input digital volumetric image.

The color-based segmentation and the syntactic-based segmentation are determined by the procedures CREATECOLORPARTITION and CREATESYNTACTICPARTITION respectively.

The color-based and syntactic-based segmentation algorithms use the tree-hexagonal structure H created by the function CREATEHEXAGONALSTRUCTURE over the voxels of the initial volumetric image, and the initial triangular grid graph G created by the function CREATEINITIALGRAPH. Because the syntactic-based segmentation algorithm uses a graph contraction procedure, CREATESYNTACTICPARTITION uses a different graph, G, extracted by the procedure EXTRACTGRAPH after the color-based segmentation finishes.

4 Segmentation results and quantitative evaluation

A true volumetric segmentation remains a difficult problem to tackle due to the complex nature of the topology of volumetric objects, the huge amount of data to be processed and the complexity of the algorithms that scale with the new added dimension [19]. Martin thesis [20] states that human segmentation can be used as the ground-truth reference in benchmarking segmentations produced by different methods.

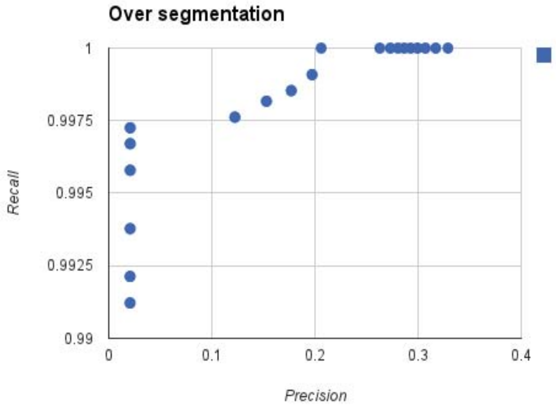


Fig.3. Experimental results.

The segmentation method used for the experimental results is based on simple hysteresis threshold. All voxels with the density within a specified threshold 'tresh' will be treated as boundary voxels while the others as empty spaces. The over-segmented volume has high recall and low precision (Figure 3), while the under-segmented image has low recall because it fails to find salient features for the volume and also low precision [18], [21].

5 Conclusions

In this paper, a graph-based theoretic framework is taken into consideration by modeling digital image segmentation as a graph partitioning and optimization problem using input spatial graph. We are introducing an algorithm for volumetric segmentation based on virtual tree-hexagonal structure (prisms) constructed on the image voxels. The key to the whole algorithms of volumetric segmentation method is the prism cells.

6 References

[1] Janakiraman, T., Mouli, P.C.. Image segmentation using Euler graphs. Int. J. of Computers, Communications and Control 5(3), pp. 314–324, (2010).

[2] Felzenszwalb, P., Huttenlocher, W.. Efficient graph-based image segmentation. International Journal of Computer Vision 59(2), pp. 167–181, (2004).

[3] Guigues, L., Herve, L., Cocquerez, L.P.. The hierarchy of the cocoons of a graph and its application to image segmentation. Pattern Recognition Letters 24(8), pp, 1059–1066, (2003).

[4] Gdalyahu, Y., Weinshall, D., Werman, M.. Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. IEEE Transactions on Pattern Analysis and Machine Intelligence 3(10), pp.1053–1074, (2001).

[5] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), pp. 885–905, (2000).

[6] Jermyn, I., Ishikawa, H.. Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(8), pp.1075–1088, (2001).

[7] Camilus, K.S., Govindan, V.. A review on graph based segmentation. International Journal of Image, Graphics and Signal Processing 5, pp. 1–13, (2012).

[8] Daniel Weinlanda, Remi Ronfardb, Edmond Boyerc. A survey of vision-based methods for action representation, segmentation and recognition, in Computer Vision and Image Understanding, Vol. 115(2), pp. 224–241, (2011).

[9] Malik, J., Belongie, S., Leung, T., Shi, J.. Contour and texture analysis for image segmentation. International Journal of Computer Vision 43(1), pp. 7–27, (2001).

[10] Comaniciu, D., Meer, P.. Robust analysis of feature spaces: color image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), pp. 603–619, (2002).

[11] Brezovan, M., Burdescu, D.D., Ganea, E., Stanescu, L.. An adaptive method for efficient detection of salient visual object from color images. In: Proceedings of the 20th International Conference on Pattern Recognition. pp. 2345–2349. Istanbul, Turkey, (2010).

[12] Burdescu, D.D., Brezovan, M., Ganea, E., Stanescu, L.. A new method for segmentation of images represented in a HSV color space. In: Proceedings of the Advanced Concepts for Intelligent Vision Systems Conference. pp. 606–617, (2009).

[13] Stanescu, L., Burdescu, D.D., Brezovan, M.. A comparative study of some methods for color medical images segmentation. EURASIP Journal on Advances in Signal Processing 128, (2011).

[14] Stanescu, L., Burdescu, D., Brezovan, M., Mihai, G.. Creating New Medical Ontologies for Image Annotation. Springer-Verlag, ISBN 978-1-4614-1908-2, (2011).

[15] Burdescu, D.D., Brezovan, M., Stanescu, L., Stoica-Spahiu, C.. Computational Complexity Analysis of the Graph Extraction Algorithm for 3D Segmentation, in: IEEE Tenth World Congress on Services-SERVICES, pp. 462-470, ISBN: 978-1-4799-5069-0, (2014).

[16] Burdescu, D.D., Brezovan, M., Stanescu, L., Stoica-Spahiu, C.. A Spatial Segmentation Method, International Journal of Computer Science and Applications, ©Technomathematics Research Foundation, Vol. 11, No. 1, pp. 75 – 100, ISSN: 2324-7037, (2014).

[17] Burdescu, D.D., Brezovan, M., Stanescu, L., Stoica-Spahiu, C.. New computational complexity analysis for a spatial segmentation algorithm, Science and Information Conference (SAI), pp.355-363, ISBN: 978-0-9893-1933-1, (2014).

[18] Dumitru Dan Burdescu, Daniel Costin Ebâncă, Florin Slabu. Graph-Based Segmentation Methods for Planar and Spatial Images, Special Issue of the International Journal of Computer Science & Applications (IJCSA), ISSN 0972-9038, vol. 12 (no. 2), pp. 120-143, (2015).

[19] Powers, D.M.. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Journal of Machine Learning Technologies 2(1), pp. 37–63, (2011).

[20] Martin, D.. An empirical approach to grouping and segmentation, Ph.D. Thesis, University of Berkeley (2002).

SESSION

VISION APPLICATIONS

Chair(s)

TBA

A Computer Vision Algorithm for Omnidirectional Bee Counting at Langstroth Beehive Entrances

Vladimir Kulyukin
Department of Computer Science
Utah State University
Logan, UT, USA
vladimir.kulyukin@usu.edu

Sai Kiran Reka
Department of Computer Science
Utah State University
Logan, UT, USA
saikiran.reka@aggiemail.usu.edu

***Abstract*—A computer vision algorithm is proposed for omnidirectional bee counting in images of Langstroth beehive entrances captured in situ with a miniature camera connected to a multi-sensor, solar-powered electronic beehive monitoring device. The algorithm consists of three stages: pre-processing, landing pad identification, and omnidirectional bee counting. In the pre-processing stage, an approximate image region where the landing pad is likely to be is cropped and the brightness of the cropped image adjusted. The landing pad identification is obtained through iterative reduction of the cropped image to the actual landing pad. Omnidirectional bee counts are computed by dividing the total number of bee pixels by the average number of pixels occupied by individual bees. The algorithm was evaluated on 1,781 images from two electronic beehive monitoring devices deployed in Langstroth beehives with live honeybees and achieved an accuracy of over 80 per cent compared to the ground truth obtained from human evaluators.**

***Keywords*—computer vision; contour analysis; color analysis; electronic beehive monitoring; sustainable computing**

I. Introduction

The Apis millifera, also known as the Western honeybee, is responsible for one out of every three daily mouthfuls that the average U.S. resident eats [1]. Since 2006 honeybees have been disappearing from amateur and commercial apiaries. This trend has been called the colony collapse disorder (CCD) [2]. Other growing threats to the health of honeybee colonies include Varroa mites, American and European foulbrood, and nosema [3].

The high rates of colony loss threaten the world's food supply chains and necessitate continuous beehive monitoring. Unfortunately, continuous beehive monitoring cannot be done by human apiarists due to obvious problems with logistics and fatigue. However, recent advances in sensor technologies have made it possible to monitor many critical variables associated with honeybee health in situ. There is an emerging consensus among researchers and practitioners that significant scientific and practical insights will likely come from transforming traditional apiaries into smart worlds monitoring their status through multiple sensors, recognizing bee behavior patterns, and notifying all interested parties about deviations and anomalies. For example, NASA researchers believe that climate changes can be investigated through pollination data [4], because beehive data clusters may relate location and pollination timings to satellite data and ecosystem models.

Most approaches to electronic beehive monitoring (EBM), defined here as electronic capture and analysis of data from beehives over regular time intervals, depend on the grid for power and on the cloud for data transmission (e.g., [5, 6]). However, grid- and cloud-dependent EBM enlarges the electricity consumption and carbon footprints of cloud data centers which already account for two percent of overall U.S. electrical usage [7, 8]. According to the Smart 2020 forecast by the Climate Group of the Global e-Sustainability Initiative [9], so far quite accurate, the global carbon footprint of cloud data centers is expected to grow, on average, 7% per annum between 2002 and 2020. In 2010, McAfee, a U.S. computer security company, reported that the electricity required to transmit the trillions of spam e-mails annually is equivalent to powering two million U.S. homes and generates the same amount of greenhouse gas emissions as that produced by three million cars [10]. Consequently, there is a critical need to seek ecologically sustainable EBM solutions that use renewable power sources and minimally depend on the cloud for data transmission and analysis.

Since electronic beehive monitoring devices (EBMDs) increasingly use multiple sensors [11], principled answers must be sought to the question of what sensors should be included and why. While sensor accuracy is a significant factor, power consumption, field reliability, and ergonomics must also be considered. The latter consideration is critical for the broader acceptance of a specific sensor by amateur and commercial apiarists, regionally, nationally, and internationally, because ease of deployment plays an important role in technology adoption.

The position presented in this paper is that computer vision can contribute to solving various problems posed by electronic beehive monitoring (EBM). In particular, computer vision can be used to solve the bee counting problem because of recent advances in miniature cameras coupled with small computational devices such as Raspberry Pi (www.raspberrypi.org) or Arduino (www.arduino.cc) that are powered with solar.

The bee counting problem, well-known in apiary science, is the problem of obtaining accurate counts of bees entering or exiting a given beehive per unit of time. One of the variables monitored by human apiarists as they inspect their beehives is forager traffic. Foraging is an indicator of honeybee colony health, colony age structure, honey flow, pollination, and climate (e.g., [12]). Consequently, accurate estimates of forager traffic levels are important not only to apiarists but also to growers, climate scientists, and sustainable farmers. Forager traffic levels can be estimated by human observers with stopwatches. However, since human observation cannot be continuous, abrupt changes in forager traffic will likely be missed.

The remainder of this paper is organized as follows. In Section II, related work is presented. In Section III, hardware and software details of BeePi®, a solar-powered, multisensor EBMD, are presented and in situ data collection is described. In Section IV, a vision-based algorithm is presented for omnidirectional bee counting on landing pads of Langstroth beehives used by most apiarists in the U.S. [2, 3]. In Section V, the experiments are presented of evaluating the algorithm on over 1,781 images captured by two deployed BeePi EBMDs in North Logan, UT. In Section VI, the results of the experiments are discussed. Section VII summarizes the results and the findings of this investigation.

II. Related Work

Because of the importance of forager traffic counts, there have been multiple research and commercial attempts to automate bee counting at hive entrances. One of the first electrical bee counters was proposed by Lundie [13]. Lundie's design was subsequently adopted and improved upon by Faberge [14] through the production of electrical impulses generated by bees tripping a balance arm. Similar electrical bee counters were proposed by other researchers (e.g., Ericson et al. [15] and Liu et al. [16]).

In earlier electrical bee counting devices, no distinction was made between counting bees entering and exiting a hive. This problem was addressed in subsequent research through bi-directional bee counters. For example, Struye et al. [17] proposed a design for a bi-directional bee counter. This design was adopted by Lowland Electronics in Belgium to manufacture bi-directional bee counters in the 1990s. These devices count bees passing through special portals equipped with infrared (IR) sensors. Bees are counted when they cross infrared beams.

Dank and Gary [18] designed a box-like extension fixed at the hive entrance to estimate the forager traffic. Each bee passes through special tubes in the box attached to the entrance. The tubes are coated with paraffin so that bees cannot only crawl through them. A mesh bag at the end of the tubes is used to collect the bees and weigh them. The bees can escape from the mesh bag.

Bromenshenk et al. [6] designed and deployed bi-directional, IR bee counters in their multi-sensor SmartHive® system. The researchers found their IR counters to be more robust and accurate than capacitance and video-based systems. Since the IR counters required regular cleaning and maintenance, a self-diagnostic program was developed to check whether all of the emitters and detectors were functioning properly and the bee portals were not blocked by debris or bees.

In addition to IR devices, some researchers used radio frequency identification (RFID) to solve the bee counting problem. For example, Schneider et al. [19] investigated pesticide effects on honeybee colonies by exposing workers from a colony of approximately 2,000 bees to contaminated sugar syrup at a feeder. The effects of pesticide exposure were measured as the RFID-detected return rate of foragers from the feeder.

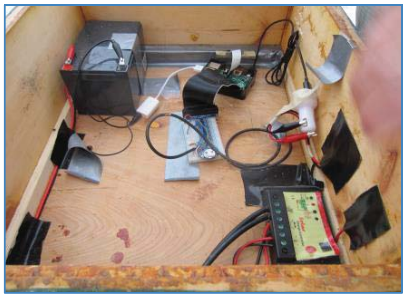


Figure 1. BeePi hardware components

III. In Situ Data Capture

A. Hardware

Images for this investigation were captured through a solar-powered, electronic beehive monitoring device (EBMD), called BeePi [20]. A fundamental objective of the BeePi design is reproducibility: other researchers and practitioners should be able to replicate our results at minimum cost and time commitments. The current BeePi hardware components are shown in Fig. 1: a Pi Model B+ 512MB RAM computer, a Pi T-Cobbler, a half-size breadboard, a DS18B20 temperature sensor, and a Pi camera. For solar harvesting, the Renogy 50 watts 12V monocrystalline solar panel was coupled with the Renogy 10 Amp PWM solar charge controller and the UPG 12V 12Ah F2 sealed lead acid AGM deep-cycle rechargeable battery.

All hardware components fit in a shallow Langstroth super, except for the solar panel that is placed on top of a beehive (see Fig. 2). The solar panels are tied to the hive supers with bungee cords. The Pi camera is placed outside to take static snapshots of the beehive's entrance, as shown in Fig. 3, with a plastic cover placed above it to protect it from the elements.

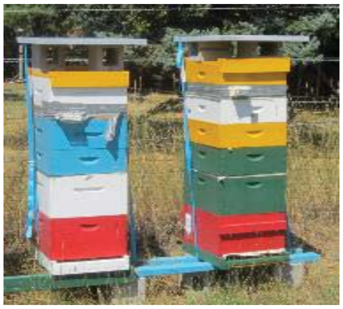


Figure 2. Solar panels on hive tops

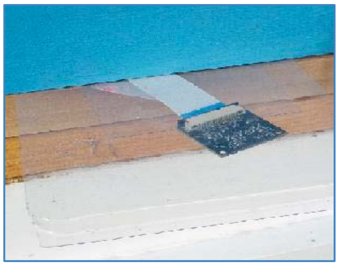


Figure 3. Pi camera looking down on landing pad

Four BeePi EBMDs were assembled in 2015 and deployed at two Northern Utah apiaries to collect 28GB of audio, temperature, and image data in different weather conditions [20]. Except for drilling narrow holes in inner hive covers for temperature sensor and microphone wires, no structural beehive modifications were done to the hives prior to deployment.

B. Software

All data collection is done in situ on the raspberry pi computer. The collected data are saved on a 16GB sdcard inserted into the pi computer. In situ data collection software is written in Python 2.7.

When the system starts, three data collection threads are spawned. The first thread collects temperature readings every 10 minutes and saves them in a text log. The second thread collects 30-second wav recordings every 15 minutes. The third thread saves PNG pictures of the beehive's landing pad every 15 minutes. The size of the image captured for the camera is 550KB with a resolution of 720 × 480 pixels.

A cronjob, i.e., an automated task that runs at specific intervals, monitors the threads and restarts them after hardware or software failures. For example, during a field deployment the camera of one of the EBMDs stopped functioning due to excessive heat. The cronjob kept periodically restarting the picture thread until the temperature went down and the camera started functioning properly again.

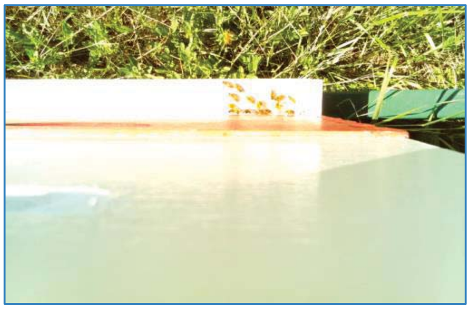


Figure 4. Sample captured image

IV. Vision-Based Bee Counting

The vision-based bee counting algorithm is omnidirectional in that it does not distinguish incoming and outgoing bee traffic. The reason why no directionality is integrated is two-fold. First, a robust vision-based solution to directionality will likely require video processing. Since the BeePi relies exclusively on solar power, in situ video capture and storage will reduce device operation times and make EBM less continuous. Second, omnidirectional bee counting can still be used as a valuable estimate of forager traffic so long as it accurately counts bees on landing pads.

The algorithm is implemented in JAVA with JDK 1.7 and the OpenCV 2.4.4 (www.opencv.org) bindings. The algorithm consists of three stages: pre-processing, landing pad identification, and omnidirectional bee counting. In the pre-processing stage, an approximate image region where the landing pad is likely to be is cropped and the brightness of the cropped region adjusted. The landing pad identification is obtained through iterative reduction of the cropped image to the actual landing pad. Omnidirectional bee counts are computed by dividing the total number of bee pixels by the average number of pixels occupied by individual bees obtained from camera calibration experiments.

A. Pre-Processing

Several in situ camera calibration experiments were conducted to estimate the coordinates of the image region where the landing pad is likely to be. The coordinates of the region are set in a configuration file and used in the algorithm to crop the region of interest. The lower image in Fig. 5 shows the output of the cropping step. Note that there may be some grass in the cropped image. The dimensions of the cropped region are intentionally set to be larger than the actual landing pad to compensate for camera swings in strong winds.

Image brightness varies greatly with the weather. When the sun is directly above the beehive, brightness is maximal. However, when the sun is obscured by clouds, captured images tend to be darker. Both cases have a negative impact on bee counting. To compensate for these two conditions, image brightness is dynamically adjusted to lie in (45, 95), i.e., the brightness index should be greater than 45 but less than 95. This range was experimentally found to yield optimal results. Fig. 6 illustrates how brightness adjustment improves omnidirectional bee counts. The upper image on the right in Fig. 6 shows a green landing pad extracted from the cropped image on the left without adjusted brightness. The lower image on the right in Fig. 6 shows a green pad extracted from the same image with adjusted brightness. Only four bees were identified in the upper image on the left whereas in the lower image eight bees were identified, which is closer to the twelve bees found in the original image by human counters.

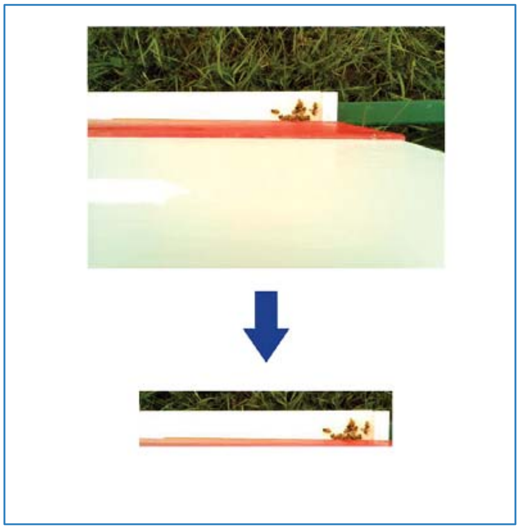


Figure 5. Cropping a landing pad region

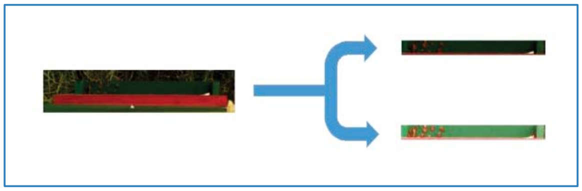


Figure 6. Adjusting image brightness

B. *Landing Pad Identification*

The three steps of the landing pad identification are shown in Fig. 7. The first step in identifying the actual landing pad in the approximate region cropped in the pre-processing stage is to convert the pre-processed RGB image to the Hue Saturation Value (HSV) format, where *H* and *S* values are computed with Equation (1).

$$H = \begin{cases} 60^\circ\left(\dfrac{G' - B'}{\Delta}\ mod\ 6\right) & if\ C_{max} = R', \\ 60^\circ\left(\dfrac{B' - R'}{\Delta} + 2\right) & if\ C_{max} = G', \\ 60^\circ\left(\dfrac{R' - G'}{\Delta} + 4\right) & if\ C_{max} = B'. \end{cases} \qquad (1)$$

$$S = \begin{cases} 0 & if\ C_{max} = 0, \\ \dfrac{\Delta}{C_{max}} & if\ C_{max} \neq 0. \end{cases}$$

In (1), R', G', B' are the *R*, *G*, *B* values normalized by 255, $C_{max} = max\{R', G', B'\}$, and $\Delta = C_{max} - min\{R', G', B'\}$. The value of *V* is set to C_{max}. In the actual implementation, the format conversion is done with the cvtColor() method of OpenCV. The inRange() method of OpenCV is subsequently applied to identify the areas of green or white, the two colors in which the landing pads of our beehives are painted. Noise is removed through a series of erosions and dilations. The white pixels in the output image represent green or white color in the actual image and the black pixels represent any color other than green or white.

To further remove noise from the image and reduce it as closely as possible to the actual landing pad, contours are computed with the findContours() method of OpenCV and a bounding rectangle is found for each contour. The bounding contour rectangles are sorted in increasing order by the *Y* coordinate, i.e., increasing rows. Thus, the contours in the first row of the image will be at the start of the list. Fig. 7 shows the bounding rectangles for the contours computed for the output image of step 3 in Fig. 7.

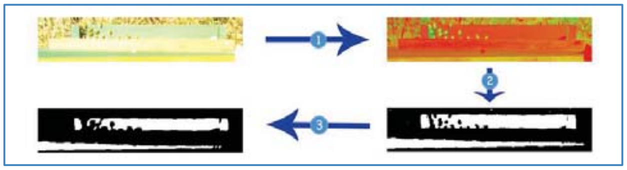


Figure 7. Landing pad identification steps: 1) HSV conversion; 2) color range identification; 3) noise removal

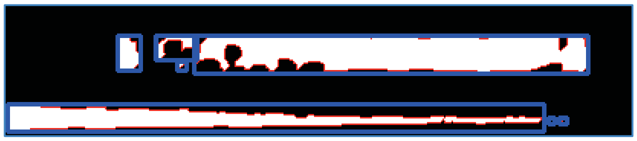


Figure 8. Bounding rectangles of found contours

Data analysis indicates that if the area of a contour is at least half the estimated area of the landing pad, the contour is likely to be part of the actual landing pad. On the other hand, if the area of a contour is less than 20 pixels, that contour is likely to be noise and should be discarded. In the current implementation of the algorithm, the estimated area of the green landing pad is set to 9,000 pixels and the estimated area of the white landing pad is set to 12,000 pixels. These parameters can be adjusted for distance.

Using the above pixel area size filter, the approximate location of the landing pad is computed by scanning through all the contours in the sorted list and finding the area of each contour. If the area is at least half the estimated size of the landing pad of the appropriate color, the *Y* coordinate of the contour rectangle is taken to be the average *Y* coordinate and the scanning process terminates. If the contour's area is between 20 and half the estimated landing pad area, the *Y* coordinate of the contour is saved. Otherwise, the current contour is skipped and the next contour is processed. When the first contour scan terminates, the average *Y* coordinate, $\mu(Y)$, is calculated by dividing the sum of the saved *Y* coordinates by the number of the processed contour rectangles.

After $\mu(Y)$ is computed, a second scan of the sorted contour rectangle list is performed to find all contours whose height lies in $[\mu(Y) - H, \mu(Y) + H]$, where *H* is half of the estimated height of the landing pad for the appropriate color. While the parameter *H* may differ from one beehive to another, as the alignment of the camera differs from one hive to another, it can be experimentally found for each beehive. For example, if the camera is placed closer to the landing pad, then *H* will have a higher value and if the camera is placed far from the landing pad, *H* will have a lower value. In our case, *H* was set to 20 for green landing pad images and to 25 and for white landing pad images.

A bounding rectangle is finally computed after the second scan to enclose all points in the found contours. To verify whether the correct landing pad area has been identified, the area of the bounding rectangle is computed. If the area of the bounding rectangle is greater than the estimated area of the landing pad, the bounding rectangle may contain noise, in which case another scan is iteratively performed to remove noise by decreasing *H* by a small amount of 2 to 4 units. In most of the cases, this extra scan is not needed, because the landing pad is accurately found. Fig. 9 illustrates the three steps of the contour analysis to identify the actual landing pad.

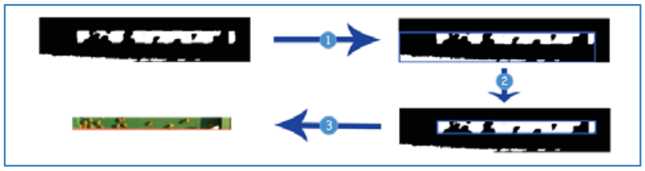


Figure 9. Contour analysis: 1) 1st contour scan; 2) 2nd contour scan; 3) pad cropping

Foreground and background pixels are separated on color. In particular, for green landing pads, the background is green and the foreground, i.e., the bees, is yellow; for white landing pads, the background is white and the foreground is yellow. All pixels with shades of green or white are set to 255 and the remaining pixels are set to 0. Three rows of border pixels of the landing pad image are arbitrarily set to 255 to facilitate bee identification in the next step. Fig. 10 shows the output of this stage. In Fig. 10, the green background is converted to white and the foreground to black. Since noise may be introduced, the image is de-noised through a series of erosions and dilation with a 2 x 2 structuring element.

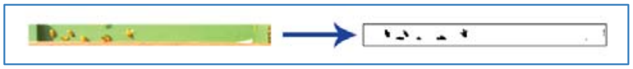


Figure 10. Background and foreground separation

C. Bee Counting

To identify bees in the image, the image from the previous stage is converted to grayscale and the contours are computed again. Data analysis suggests that the area of an individual bee or a group of bees vary from 20 to 3,000 pixels. Therefore, if the area of a contour is less than 20 pixels or greater than 3,000 pixels, the contour is removed.

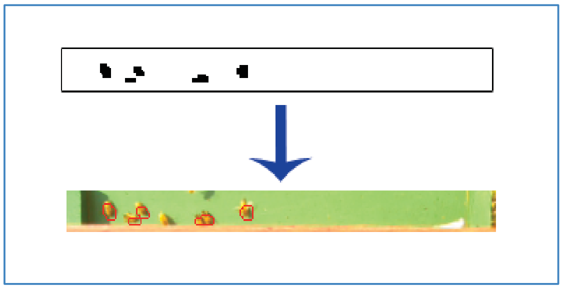


Figure 11. Omnidirectional bee counting

The area of one individual bee is between 35 and 100 pixels, depending on the distance of the pi camera from the landing pad. The green landing pad images were captured by a pi camera placed approximately 1.5m above the landing pad with the average area of the bee being 40 pixels. On the other hand, the white landing pad images were captured by a pi camera placed approximately 70cm above the landing pad where the average area of an individual bee is 100 pixels. To find the number of bees in green landing pad images, the number of the foreground pixels, i.e., the foreground area, is divided by 40 (i.e., the average bee pixel area on green landing pads), whereas, for the white landing pad images, the foreground area is divided by 100 (i.e., the average bee pixel area on white landing pads). The result is the most probable count of bees in the image. In the upper image in Fig. 11, five bees are counted by the algorithm. The lower image in Fig. 11 shows the found bees in the original image.

V. Experiments

A sample of 1,005 green pad images and 776 white pad images were taken from the data captured with two BeePi EBMDs deployed at two Northern Utah apiaries [20]. Each image has a resolution of 720 x 480 pixels and takes 550KB of space. To obtain the ground truth, six human evaluators were recruited. Each evaluator was given a set of images and asked to count bees in each image and record his or her observations in a spread sheet. The six spread sheets were subsequently combined into a single spread sheet.

Table I gives the ground truth statistics. The human evaluators identified a total of 5,770 bees with an average of 5.7 bees per image in images with green landing pads. In images with white landing pads, the evaluators identified a total of 2,178 bees with a mean of 2.8 bees per image.

Table II summarizes the performance of the algorithm ex situ on the same green and white pad images. The algorithm identified 5,263 bees out of 5,770 in the green pad images with an accuracy of 80.5% and a mean of 5.2 bees per image. In the white pad images, the algorithm identified 2,226 bees out of 2,178 with an accuracy of 85.5% and an average of 2.8 bees per image. The standard deviations of the algorithm were slightly larger than those of the human evaluators.

Table I. Ground Truth

Pad Color	Num Images	Total Bees	Mean	STD
Green	1,005	5,770	5.7	6.8
White	776	2,178	2.8	3.4

Table II. Accuracy (%) of the Algorithm

Pad Color	Num Images	Total Bees	Mean	STD	ACC
Green	1,005	5,263	5.2	7.6	80.5
White	776	2,178	2.8	4.1	85.5

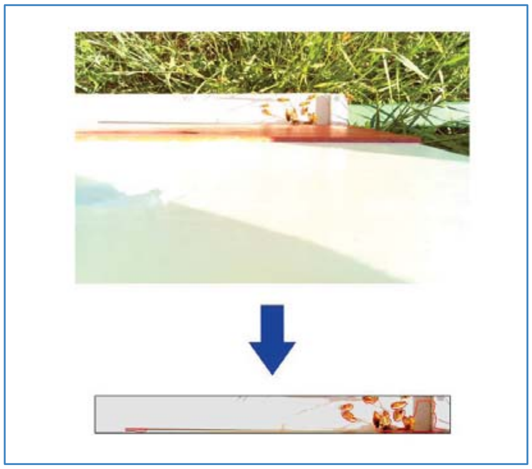


Figure 12. False positives

VI. Discussion

Our analysis of the results identified both true negatives and false positives. There appear to be fewer true negatives than false positives. The main reason for true negatives is the algorithm's conservative landing pad identification, which causes some actual bees to be removed from the image. The bees on the sides of the landing pad are also typically removed from the image. Another reason for true negatives is image skewness due to wind induced camera swings. If the landing pad is skewed, then a part of the landing pad is typically cropped out during the bounding rectangle computation. In some images, some actual bees were removed from images during image de-noising, which resulted in lower bee counts compared to human counts.

False positives were primarily caused by occasional shades, leaves, or blades of grass wrongly counted as bees. Fig. 12 gives an example of false positives. A human evaluator counted 9 bees in the upper image whereas the algorithm counted 28 bees on the landing pad (lower image in Fig. 12) cropped out of the upper image. The shade pixels on the right end of the cropped landing pad were counted as bees, which resulted in a much higher bee count than the human evaluator's count.

VII. Summary

A computer vision algorithm was presented for omnidirectional bee counting at Langstroth beehive entrances. The algorithm was evaluated on a total of 1,781 images and achieved an accuracy of over 80 per cent compared to the ground truth obtained from six human evaluators. The performance of the algorithm can be further improved through a more accurate identification of the landing pad's skew and image rotation before actual bee counting.

Acknowledgment

The first author expresses his gratitude to Neil Hattes who jumpstarted this project by donating a raspberry pi computer, a mother board, a temperature sensor, and a monitor. The in situ data collection software was originally developed and tested on this equipment. All bee packages and beehive woodenware used in this study were funded by the first author.

References

[1] Walsh, B. "A world without bees." Time, pp. 26-31, August 19, 2013.

[2] Conrad, R. *Natural beekeeping: organic approaches to modern apiculture*. Chelsea Green Publishing, White River Junction, Vermont, 2013.

[3] Crowder, L. & Harrell, H. *Top-bar beekeeping: organic practices for honeybee health*. Chelsea Green Publishing, White River Junction, Vermont, 2012.

[4] NASA HoneyBeeNet Project. http://honeybeenet.gsfc.nasa.gov/.

[5] Meikle, W.G. & Holst, H. "Application of continuous monitoring of honeybee colonies." Apidologie, vol. 46, pp. 10-22, 2015.

[6] Bromenshenk, J.J., Henderson, C.B., Seccomb, R.A., Welch, P.M., Debnam, S.E., & Firth, D.R. "Bees as biosensors: chemosensory ability, honey bee monitoring systems, and emergent sensor technologies derived from the pollinator syndrome." Biosensors, vol. 5, pp. 678-711, 2015.

[7] Walsh, B. "Your data is dirty: the carbon price of cloud computing." Time, April 2, 2014.

[8] Vaugh, A. "How viral cat videos are warming the planet." The Guardian, Sept. 25, 2015.

[9] "Smart2020: Enabling the low carbon economy in the information age." The Global e-Sustainability Initiative. Avail. at http://www.smart2020.org.

[10] Berners-Lee, M. & Clark, D. "What's the carbon footprint of … email?" The Guardian, October 7, 2010.

[11] McNeil, M.E.A. "Electronic beehive monitoring." American Bee Journal, August 2015, pp. 875 – 879.

[12] Gary, N.E. Chapter 8. "Activities and behavior of honey bees." In: Graham, J.M. (ed.) The hive and the honey bee, pp. 269–372, 1992.

[13] Lundie, A. E. "The flight activities of the honey bees", United States Department of Agriculture, Dept. Bull. No. 1328, 1925.

[14] Faberge, A.C. "Apparatus for recording the number of bees leaving and entering a hive". J. Sci. Instr., vol. 20, pp. 28–311, 1943.

[15] Erickson, E.H., Miller, H.H., & Sikkema, D.J. "A method of separating and monitoring honey-bee flight activity at the hive entrance". J. Apic. Res., vol. 14, pp. 119–125, 1975.

[16] Liu, C., Leonard, J., & Feddes, J.J. "Automated monitoring of flight activity at a beehive entrance using infrared light sensors". J. Apic. Res., vol. 29(1), pp. 20–27, 1990.

[17] Struye, M.H., Mortier, H.J., Arnold, G., Miniggio, C., & Borneck, R. "Microprocessor-controlled monitoring of honeybee flight activity at the hive entrance". Apidologie, vol. 25, pp. 384–395, 1994.

[18] Danka, R. G., & Gary, N.E. "Estimating foraging populations of honey bees (Hymenoptera: Apidae) from individual colonies." Journal of economic entomology, vol 80.2, pp. 544-547, 1987.

[19] Schneider, C.W., Tautz, J., Grünewald, B., & Fuchs, S. "RFID Tracking of sublethal effects of two neonicotinoid insecticides on the foraging behavior of Apis mellifera". PLoS ONE, vol. 7(1), e30023. doi:101371/journal.pone.0030023, 2012.

[20] Kulyukin, V., Putnam, M., & Reka, S. K. "Digitizing buzzing signals into A440 piano note sequences and estimating forage traffic levels from images in solar-powered, electronic beehive monitoring." In Edtrs. S. I. Ao, Oscar Castillo, Craig Douglas, David Dagan Feng, and A. M. Korsunsky, *Proceedings of International MultiConference of Engineers and Computer Scientists* (IMECS 2016): *International Conference on Computer Science*, vol. I, pp. 82-87, March 16-18, Kowloon, Hong Kong, IA ENG, ISBN: 978-988-19253-8-1;ISSN: 2078-0958, 2016.

Automated Distortion Defect Inspection of Transparent Glass Using Computer Vision

Hong-Dar Lin, Yuan-Chin Lo
Department of Industrial Engineering and Management, Chaoyang University of Technology,
Taichung, 41349, Taiwan

Abstract –*If a car windshield with distortion flaws will make object deformation and motion blur from the driver's sight easily, the drivers can cause visual misjudgment and have safety concerns on the road. This study presents the design of an automated distortion defect inspection system of car glass. In this study, a standard pattern with vertical lines displayed on a testing glass is captured as a testing image. First, a testing image is transformed to Hough domain to obtain the coordinates of the correct axis positions of multiple vertical lines. Through the accumulator analysis to find the peak points of the vertical lines in Hough domain, an image with new vertical lines is reconstructed from the selected peak points by taking the inverse Hough transform. Secondly, the binary testing image subtracts the binary reconstructed image to obtain a binary difference image of distortion defects. Finally, the cumulated deviation ratios of distorted segments are calculated and the offset pixel ratio of distortion segments reveals the level of distortion in the image. Experimental results show that the proposed method effectively determines whether there are distortion flaws with the occurrence location, as well as distortion segment cumulative pixel ratio.*

Keywords: Industrial inspection; transparent glass; distortion defects; computer vision system; Hough Transform.

1 Introduction

If a car windshield with optical distortion flaws will make object deformation and motion blur from the driver's sight easily, the drivers can cause visual misjudgment and have safety concerns on the road. Since the distortion defects do not have regular shapes and clear boundaries, it is not easy to measure the magnitudes of distortion defects on curved windshields. Furthermore, the curved glass with the property of higher reflection increase the difficulty of discrimination of the distortion defects on car windshields. Therefore, this research aims at exploring the automated visual inspection of transmitted distortion defects of the curved car windshields.

The defective car windshields with transmitted distortion defects providing shape-distorted scene information may lead car drivers making wrong decisions when driving. Figure 1 shows the defective car windshield images with transmitted distortion defects on parking lot scene. The object shapes transmitted in the defective image are significantly distorted. The glass distortion defects may make transmitted objects look irregularly, out of focus, and blurry in the defective images. These distorted images may result in making wrong judgment by car drivers and lead to dangerous car accidents.

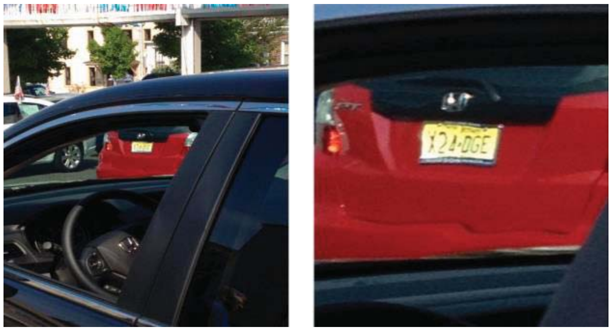


Figure 1. The defective car windshield images with transmitted distortion defects.

Inspection difficulties of surface defects are existing in manufacturing process. Surface defects affect not only the appearance of industrial parts but also their functionality, efficiency and stability. The most common detection methods for surface defects are human visual inspections. Human inspection is vulnerable to wrong judgments owing to inspectors' subjectivity and eye fatigues. Figure 2 shows the current inspection tasks by human visual judgment and the testing images with transmission of standard patterns. Furthermore, difficulties also exist in precisely inspecting distortion defects by computer-aided machine vision systems because when product images are being captured, the region of a distortion defect could expand, shrink or even disappear due to uneven illumination of the environment, different view angles of the inspectors, shapes of transmitted patterns, and so on.

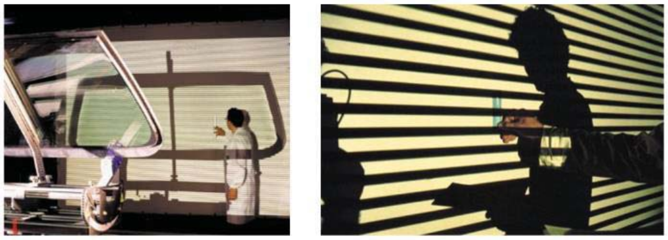


Figure 2. Current inspection tasks by human visual judgment and the testing images with transmission of standard patterns.

Current computer-aided vision system (off-line and sampling) uses a horizontal or/and vertical lines pattern transmitted on glass to acquire images and quantize distortion magnitude for screening. It is hard to precisely inspect the glass distortion flaws by current machine vision systems due to high transmission and reflection. The property of higher transmission and reflection on curved glass increases the difficulty of discrimination of the distortion defects on car glass. In this research, the testing samples with length 25.4 cm, width 20.4 cm, and thickness 0.2 cm, were randomly selected from manufacturing process of car glass. Figure 3 shows the dimension of the testing sample with high transmission and reflection. This study proposes a Hough transform based approach to inspect transmitted distortion defects on curved car glass.

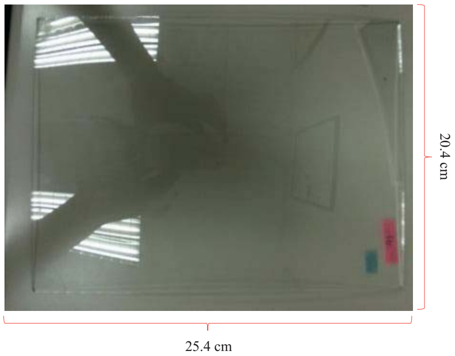


Figure 3. Dimension of the testing sample with high transmission and reflection.

2 Automated defect inspections

Automated visual inspection of surface flaws has become a critical task for manufacturers who strive to improve product quality and production efficiency [1-3]. Li and Tsai [4] proposed a wavelet-based discriminant measure for defect inspection in multi-crystalline solar wafer images with inhomogeneous texture. The proposed method performs effectively for detecting fingerprint, contaminant, and saw-mark defects in solar wafer surfaces. Chiou [5] presented an intelligent method for automatic selection of a proper image segmentation method upon detecting a particular flaw type in roll-to-roll web inspection. The results show a significant reduction in misclassification rate. Perng et al. [6] developed a fast and robust machine vision system for wire bonding inspection. A new lighting environment was devised which will highlight the slope of the bonding wire and suppress the background from being extracted.

Many researches explored the defect detection of glass related products. Adamo et al. [7] proposed a low-cost inspection system based on the Canny edge detection for online defects assessment in satin glass. Liu et al. [8] presented a method based on watershed transform to segment the possible defective regions and extract features of bottle wall by rules. Lin and Tsai [9] presented a Fourier transform-based approach to inspect surface defects of capacitive touch panels. A multi-crisscross filter is designed to filter out the frequency components of the principal band regions. The defective region would be clearly retained in the restored image. Chiu and Lin [10] applied block discrete cosine transform, Hotelling's T-squared statistic, and grey clustering technique for the automatic detection of visual blemishes in curved surfaces of LED lenses.

Regarding the distortion correction techniques, Duan and Wu [11] proposed a method for distortion correction in the barrel distortion of wide-angle lens. The cubic B-spline interpolation function was adopted to interpolate the surface and the bi-linear interpolation was used to reconstruct the gray level of pixels. Zhang et al. [12] presented a distortion-correction technique that can automatically calculate correction parameters, without precise knowledge of horizontal and vertical orientation. Based on a least-squares estimation, the algorithm considers line fits in both field-of-view directions and global consistency that gives the optimal image center and expansion coefficients. Ngo and Asari [13] presented an architecture design for real-time correction of nonlinear distortion in wide-viewing angle camera images. The architecture is designed based on the method of back mapping the pixels in the corrected image space to the distorted image space and performing linear interpolation of four neighboring pixel intensities.

Smith and Smith [14] proposed a methodology for improving the accuracy of machine vision calibration through applying regression analysis and neural network modelling. The regression analysis was employed for assisting with the data collection for neural network training and the neural network was developed for modelling the error in measured location of image features. Lin and Hsieh [15] proposed a vision system with a trapezoidal mask for image acquisition and applies cumulative sum control schemes to inspect distortion defects on curved car mirrors.

3 Proposed method

3.1 Image pre-processing

The captured testing image will be pre-processed in several steps. Figure 4 shows the original testing image and enhanced image performed the equalization approach for increasing contrast in gray levels. From the analysis of two corresponding intensity histograms, the contrast of gray levels has been increased and the vertical lines looked clearer in the enhanced image. Figure 5 depicts the enhanced normal and defective images and their corresponding binary images that the Otsu method [16] applied to do segmentation. Most of the vertical lines are clearly segmented from background in the binary images by Otsu method. The results reveal that the slight distortion defects in transparent glass surface are correctly separated in the binary image, regardless of insignificant distortion differences.

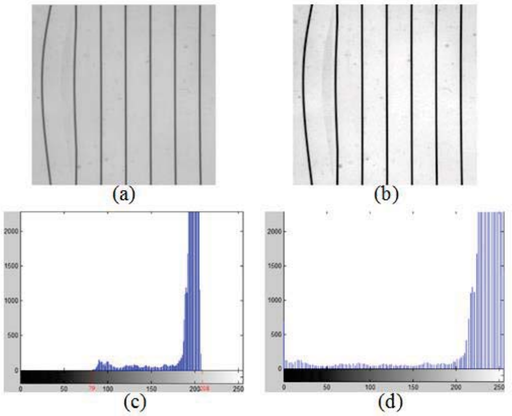


Figure 4. A testing image and its enhanced image with corresponding intensity histograms.

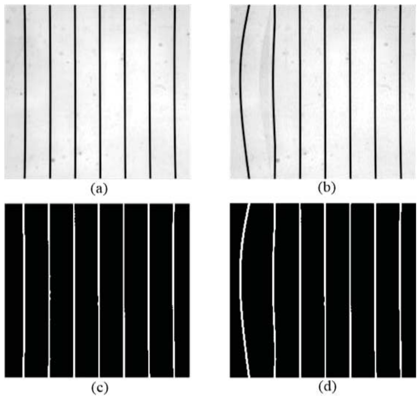


Figure 5. The enhanced normal and defective images with corresponding binary images.

3.2 Reconstruction of baselines image through Hough transform

Hough transform is a feature extraction technique used in image analysis, computer vision, and image processing. This transform was invented by P.V.C. Hough in 1962 [17]. It is commonly applied to find imperfect instances of objects within a certain class of shapes by a voting procedure. The purpose of the Hough transform is to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects [18-19].

3.2.1 Hough transform

The simplest case of Hough transform is detecting straight lines [20]. In general, the straight line equation:

$$y = ax + b \tag{1}$$

$$b = -ax + y \tag{2}$$

The equation can be represented as a point (a , b) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter a . Thus, for computational reasons, the Hesse polar form is proposed of using the parametric representation of a line:

$$x\cos\theta + y\sin\theta = \rho \tag{3}$$

The variable ρ is the distance from the origin to the line along a vector perpendicular to the line and θ is the angle between the x-axis and this vector. Each point in the (*x*, *y*) plane gives a sinusoid in the (ρ , θ) plane. M collinear points lying on the line (Equation (3)) will give M curves that intersect at (ρ_i, θ_j) in the parameter plane. The Hough transform generates a parameter space matrix whose rows and columns correspond to these ρ and θ values, respectively. The linear Hough transform algorithm uses a two-dimensional array, called an accumulator, to detect the existence of a line. The dimension of the accumulator equals the number of unknown parameters, i.e., two values in the pair (ρ , θ). The input to a Hough transform is normally a binary image that has been segmented. Figure 6 shows the two Hough parameter spaces of the normal and defective binary images shown in Figure 5, respectively. We cannot find any difference between the normal and defective images in Hough parameter spaces.

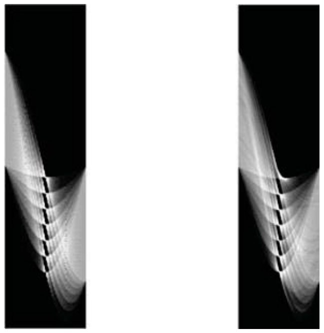


Figure 6. The two Hough parameter spaces of the normal and defective binary images, respectively.

3.2.2 Accumulators in Hough domain

Each element of the matrix has a value equal to the sum of the points or pixels that are positioned on the line represented by quantized parameters *H*(ρ , θ). So the element with the highest value indicates the straight line that is most represented in the input image. For each pixel at *f*(*x*, *y*) and its neighborhood, the Hough transform algorithm determines if there is enough evidence of a straight line at that pixel. If so, it will calculate the parameters *H*(ρ , θ) of that line, and then look for the accumulator's bin that the parameters fall into, and increment the value of that bin. The local maxima of the accumulators will give the significant lines. Figure 7 shows the corresponding relationship between spatial binary image and Hough parameter space. It indicates that the standard pattern with 7 line segments is displayed on the binary image of a testing sample and there are 7 corresponding intersection points of curves in the Hough parameter space. The 7 vertical line segments are the targets need to be detected and their positions are located between the coordinates $H(362, 91)$ and $H(617, 91)$ in the Hough parameter space.

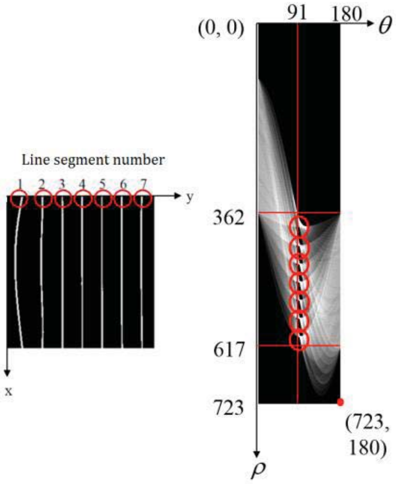


Figure 7. The corresponding relationship between spatial binary image and Hough parameter space.

By finding the bins with the highest values, typically by looking for local maxima in the accumulator space, the most likely lines can be extracted. The simplest way of finding these peaks is by applying some form of threshold. Since the lines returned do not contain any length information, it is often necessary to find which parts of the image match up with which lines. Moreover, due to imperfection errors in the edge detection step, there will usually be errors in the accumulator space, which may make it non-trivial to find the appropriate peaks, and thus the appropriate lines. Figure 8 indicates that there are 3 peaks in the accumulators of coordinates $H(362, 91)$ to $H(490, 91)$ and 4 peaks in the accumulators of coordinates $H(491, 91)$ to $H(617, 91)$ in Hough parameter space. These 7 peaks represent 7 vertical line segments in the spatial domain.

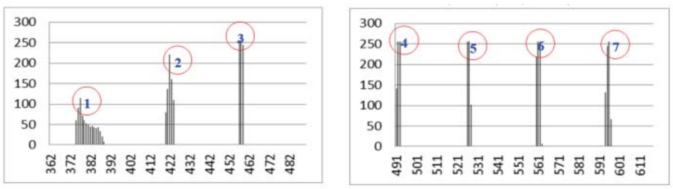


Figure 8. The 7 peaks and accumulators of coordinates $H(362, 91)$ to $H(490, 91)$ and coordinates $H(491, 91)$ to $H(617, 91)$ in Hough parameter space.

The Hough Transform generates parameter values ρ and θ for all lines that could go through each detected (by a threshold, in this example) image point. Each possible line through each point then votes for its ρ and θ values in a parameter space of possible ρ and θ values. We limit and quantize this parameter space to get an accumulator space which accumulates votes for ρ and θ values. After all possible lines through all detected points have voted, the accumulator space is searched for peaks that indicate which pairs of ρ and θ parameters got the most votes. A peak indicates the presence of line and gives its parameters and equation in the image. We let the p_i be the location of peak i in the parameter space, k is the distance between two line segments, x_0 is the initial location of the first line segment. The location of peak i can be determined as follows,

$$p_i = \max(H(x_0 + ((i-1)k), 91) \sim H(x_0 + (i \times k), 91)) \quad (4)$$

After the positions of all peaks are located in parameter space, we need to transform them back to spatial domain for obtaining a baselines image. We assume q_i be the coordinate of the peak i in spatial domain. It can be obtained as follows,

$$q_i = p_i - 360 \quad (5)$$

The peaks in parameter space transformed back to spatial domain are the baselines in the reconstructed image. Figure 9 shows the testing binary image and corresponding reconstructed baselines image with marks of line segments. If the two binary images are precisely aligned, the distortion defects can be found and located.

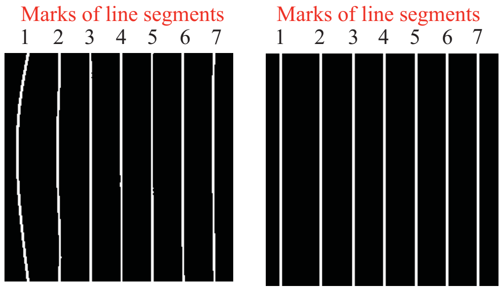


Figure 9. The testing binary image and corresponding reconstructed baselines image with marks of line segments.

3.3 Comparisons of image differences

For comparing the differences between the testing binary image and the reconstructed baselines image, we use image subtraction to obtain the resulting image expressed as follows,

$$re(x, y) = bw(x, y) - f'(x, y) \quad (6)$$

where $bw(x, y)$ is a binary image, $f'(x, y)$ is a reconstructed baselines image, and $re(x, y)$ is the resulting image that indicates the locations of detected distortion defects.

4 Experiments and analyses

To evaluate performance of the proposed approaches, experiments were conducted on real curved car glass, provided by a car windshield manufacturing company. All samples were randomly selected from manufacturing process of car glass. Testing images (40) of the curved car glass, of which 10 have no distortion defects (normal samples) and 30 have various transmitted distortion defects (defective samples), were tested. Each image of the surface has a size of 256 × 256 pixels and a gray level of 8 bits. The proposed distortion defect detection algorithm is edited in Matlab language and executed on the 7th version of the MATLAB interactive environment (data analysis, algorithm development, and model creations and applications). The system is implemented on a personal computer with CPU Inter (R) Core(TR) i5-3230M and 8 GB RAM.

4.1 Detection results for two severity levels of distortion defects

This study proposes a Hough transform based approach to inspect transmitted distortion defects on curved car glass. Figure 10 shows the initial and resulting images of the testing samples with serious and minor distortion defects. It indicates that not only the serious distortion defects but also the minor defects can be detected by the proposed method under proper parameter selection.

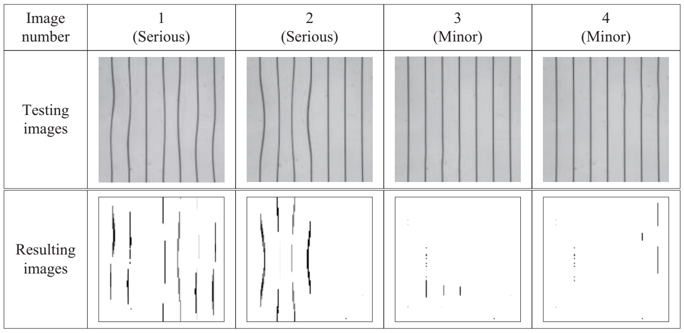


Figure 10. The initial and resulting images of testing samples with serious and minor distortion defects.

4.2 Detection results for standard patterns with different numbers of line segments

In the previous experiments, the standard pattern with 7 line segments is used to project the line pattern on testing images through the transmission of transparent glass. If we change the standard patterns with different numbers of line segments, the detection results of distortion defects will be different. The more line segments in a standard pattern, the more accuracy to present the deformation level in a testing image. We use different standard patterns with three numbers of line segments, 6, 7, and 8, to quantify the deformation of a car glass due to distortion defects. Figure 11 shows the initial, processed, and resulting images by the proposed method for the three standard patterns with different line segments. From the comparison of the resulting images, the detection result using the standard pattern with 7 line segments has better inspection performance because of less false alarms.

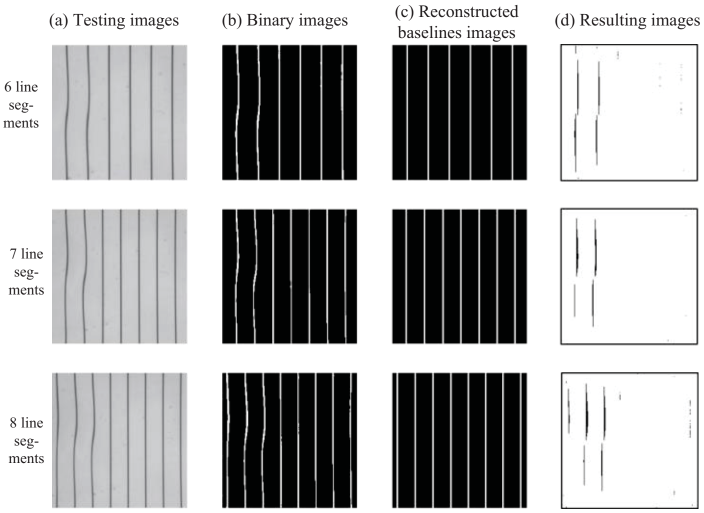


Figure 11. The initial, processed, and resulting images by the proposed method for standard patterns with different numbers of line segments.

4.3 Overall performance index of detection results for distortion defects

To present the overall detection performance of distortion defects in the testing samples, an index *CR*, called distortion defect ratio, is defined as follows,

$$CR = \frac{total\_re}{total\_f'} \tag{7}$$

where the $total\_re$ is the pixel number of detected distortion defects in a resulting image, and the $total\_f'$ is the pixel number of line segments in a reconstructed baselines image. The *CR* value is the ratio of detected distortion pixels to reconstructed baselines pixels and it locates between 0 and 1. This index indicates the deformation level of a testing image. The larger the index, the worse the distortion degree. Figure 12 shows the distribution of distortion defect ratios in the 40 testing samples. We find the samples 1 to 20 have serious distortion defects, the samples 21 to 30 have minor distortion flaws, and the samples 31 to 40 are normal based on the judgments of magnitudes of the CR values.

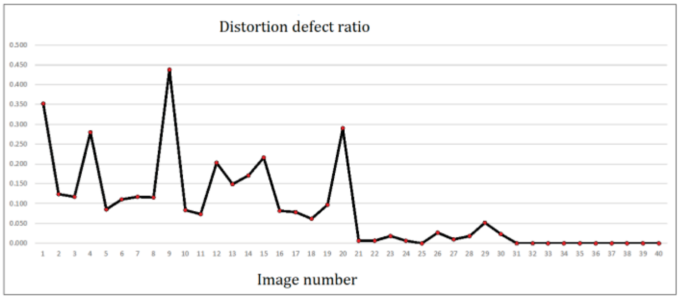


Figure 12. The distribution of distortion defect ratios in 40 testing samples.

5 Conclusions

This study proposes a novel approach based on Hough transform scheme to inspect transmitted distortion defects on curved car glass. To quantify the deformation of a car glass, a standard pattern with vertical lines transmitted on a testing glass is captured as a testing image. The testing image is transformed to Hough domain to obtain the coordinates of the correct axis positions of multiple vertical lines. Through the accumulator analysis to find the peak points of the vertical lines in Hough domain, an image with new vertical lines is reconstructed from the selected peak points by taking the inverse Hough transform. Then, the binary testing image subtracts the reconstructed baselines image to obtain a binary difference image of distortion defects. Finally, the cumulated deviation ratios of distorted segments are calculated and the offset pixel ratio of distortion segments reveals the level of distortion in the image. Experimental results show that the proposed method effectively determines whether there are distortion flaws with the occurrence location, as well as distortion segment cumulative pixel ratio.

6 Acknowledgment

This work was partially supported by the National Science Council (NSC) of Taiwan, under Grant No. MOST 104-2221-E-324-010.

7 References

[1]. S. H. Huang, Y. C. Pan, "Automated visual inspection in the semiconductor industry: A survey," *Computers in Industry*, **66**, 1-10 (2015).

[2]. H. D. Lin, "Tiny surface defect inspection of electronic passive components using discrete cosine transform decomposition and cumulative sum techniques", *Image and Vision Computing*, **26**, 603-621, (2008).

[3]. E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, J. D. Legat, "A survey on industrial vision system, applications, and tools", *Image and Vision Computing*, **21**, 171-188, (2003).

[4]. W. C. Li, D. M. Tsai, "Wavelet-based defect detection in solar wafer images with inhomogeneous texture", *Pattern Recognition*, **45**, 742-756 (2012).

[5]. Y. C. Chiou, "Intelligent segmentation method for real-time defect inspection system," *Computers in Industry*, **61**, 646-658 (2010).

[6]. D. B. Perng, C. C. Chou, S. M. Lee, "Design and development of a new machine vision wire bonding inspection system", *International Journal of Advanced Manufacturing Technology*, **34**, 323-334 (2006).

[7]. F. Adamo, F. Attivissimo, A. Di. Nisio, M. Savino, "A low-cost inspection system for online defects assessment in satin glass," *Measurement*, **42**, 1304-1311 (2009).

[8]. H. Liu, Y. Wang, and F. Duan, "Glass bottle inspector based on machine vision," *International Journal of Computer Systems Science and Engineering*, **3(3)**, 162-167 (2008).

[9]. H. D. Lin, H. H. Tsai, "Automated quality inspection of surface defects on touch panels," *Journal of the Chinese Institute of Industrial Engineers*, **29(5)**, 291-302 (2012).

[10]. Y. P. Chiu, H. D. Lin, "An innovative blemish detection system for curved LED lenses," *Expert Systems with Applications*, **40(2)**, 471-479 (2013).

[11]. M. L. Duan, K. X. Wu, "New method of correcting barrel distortion on lattice model," *Journal of Computer Applications*, 1113-1115 (2012).

[12]. C. Zhang, J. P. Helferty, G. McLennan, W. E. Higgins, "Nonlinear distortion correction in endoscopic video images", *IEEE ICIP-2000*, 34, 439-442 (2000).

[13]. H.T. Ngo, V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images", *IEEE Transactions on Circuits and Systems for Video Technology*, 15, 436-444 (2005).

[14]. L. N. Smith, M. L. Smith, "Automatic machine vision calibration using statistical and neural network methods", *Image and Vision Computing*, 23, 887-899 (2005).

[15]. H. D. Lin, K. S. Hsieh, "Automated distortion defect inspection of curved car mirrors using computer vision," *2015 International Conference on Image Processing, Computer Vision, & Pattern Recognition (**IPCV-2015**)*, 361-367 (2015).

[16]. N. Otsu, "A threshold selection method from gray level histogram", *IEEE Transactions on Systems, Man and Cybernetics*, 9, 62-66, (1979).

[17]. P. V. C. Hough, "Method and means for recognizing complex patterns," United States Patent Office 3069654 (1962).

[18]. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recognition, 11(2), 111-122 (1981).

[19]. R. K. Satzoda, S. Sathyanarayana, T. Srikanthan, "Hierarchical additive Hough transform for lane detection," IEEE Embedded Systems Letters, 2(2), 23-26 (2010).

[20]. R. C. Gonzalez, R. E. Woods, Digital Image Processing (3/e), Pearson Education, Upper Saddle River, New Jersey (2008).

Face recognition using Eigensurface on Kinect depth-maps

Marcelo Romero[1], Cesar Flores[1], Vianney Muñoz[1] and Luis Carlos Altamirano[2]
Universidad Autónoma del Estado de México[1] and *Benemérita Universidad Autónoma de Puebla*[2]
{mromeroh}@uaemex.mx

Abstract

This paper introduces an original investigation that takes advantages of an economical depth-map camera and state of the art algorithms for 3D face recognition. For this research, we have prototyped a face recognition system that uses Kinect depth-maps to grant access into our Human Computer Interaction Laboratory. Our implemented prototype uses Eigensurface for recognition, embedded into an on-line functional system that accessing its enrolled-users' records would be able to allow/deny access into our facilities. Our face recognition system consists of three main stages. Firstly, a user's depth-map is acquired using the Kinect Sensor and Matlab's standard image acquisition Toolbox. Secondly, it is carried-out an innovative and effective face detection process that corresponds a point-cloud face model to the acquired image to detect and extract the facial surface. After that, the face image is processed to get smoothed and scaled data. Thirdly, recognition is performed based on the Eigensurface algorithm. We have evaluated both face recognition scenarios: identification and verification. So far, our preliminary experimental results are encouraging our final aim in obtaining an economical and functional face recognition system.

1. Introduction

It is generally accepted that biometric–based recognition techniques have several advantages against commonly used ways to access to private facilities or services [18]. For example, keys, magnetic cards, tokens, passwords and similar will grant access to anyone using them, no matter if he/she is authorized. Furthermore, such identity items can be easily stolen or calculated [23-26].

As described in [18] physiological characteristics such as face, fingerprint, voice, retina, iris, ear and voice are some of the technologies to measure biometric features. Most of these technologies are classified as active due they need of cooperation of the subject [24]. Face recognition takes advantage as a non-intrusive way and only requires a minimum cooperation of a subject. Taking that advantage, governmental agencies have supported investigations to develop more robust face recognition techniques organizing conferences and collecting face databases to for evaluation [19-22].

It is said that face recognition techniques based in intensity images have limitations due the 3D world is projected to a 2D image, thus losing depth information and creating ambiguity [26].

In comparison to 2D techniques, 3D face recognition allows us to measure the curvature of facial areas like cheeks, jaw line, nose, forehead, etc. without variations because of lighting, face orientation and background at cost of some computational complexity, but, in a pose invariant way [18][25].

Different techniques have been investigated to obtain 3D information, for example: a) scanning systems based on laser scanners which produce highly accurate results; b) structured light which uses the principle of stereo vision and only does require a camera and any projection system; c) stereo vision, which using multiple 2D images taken from different angles, attempts to extract 3D information [23]. In this paper we are experimenting the structured light sensor.

Typically, 3D information can be represented as: depth maps, a 2D image with range values, or point clouds, which are a set of unorganized points in $\mathbb{R}^3$. Relations among vertices in point clouds can be computed to represent the surface of the 3D model [17][26].

In essence, this paper describes a straightforward pipeline for face recognition, which has been developed within a research project [6]. This research is aim to develop an economical biometric system based on the Kinect 3D sensor [11] for face recognition. We have previously been studied the Kinect camera [12-16] and we believe that it can be used for this task with acceptable performance.

Our face recognition system is based on Eigensurface [1], which is a direct implementation of the classical Eigenfaces [2], because of two main reasons. The first reason is that Eigensurface could be considered a base line in the 3D face recognition. The second reason is that we are using this investigation to implement a pipeline, which allows us to experiment a set of face recognition algorithms.

The rest of this paper is as follows. Section 2 presents related work in face recognition. Section 3 introduces our experimental procedure. Section 4 shows our performance evaluation. Finally, Section 5 concludes this paper and draws some venues for future work.

2. Related work

Face recognition has attracted several researchers for decades, as a consequence, related literature is well served. In this section we generally describe some related work in 3D face processing, that we consider relevant for our research.

Bronstein et al. [27] propose a system to address the issue of facial expressions. They propose to extract the facial region using geodesic distance and analyse the intrinsic properties of the facial surface that are expression-invariant. They postulate that measuring the geodesic distances between points on the facial surface simplifies the task of recognition.

Using depth maps, Spreeuwers [28] proposes a system to perform facial profile recognition. To avoid facial expressions and pose variations and using the landmarks of nose's root and tip of each point cloud, an intrinsic coordinate system is calculated. The face profile is projected in parallel plane of this coordinate system. A PCA-LDA implementation is used to compute the likelihood-ratio of two images to perform the recognition.

FaceEnforce is a commercial system that uses a 3D camera and a pattern recognition algorithm to perform 3D and 2D+3D face recognition [30]. Both, software and hardware are developed by Cybula Ltd. Although this system could deal with disadvantages against the angle of the face, lighting conditions and background clutter, it is able to work with a large number of faces.

In [29] multiple regions of the face, located by skin detection and identifying the eye pits, nose tip and nose bridge by its curvature values, are used to compare performance of the Eigenface recognition technique (PCA based). Combinations of 3 patches of the facial central region were used to recognize subjects in his system. These regions are from the area among nose and the eyes. They reported that combining the 3 patches gives the best recognition.

It is remarkable that even though the performance of Eigenfaces technique, it is a widely used technique in face recognition systems to compare their performance. The reported performance of PCA-based algorithms such as Eigenfaces varies from 61.3% to 77.6% [29].

In 2013, the Delft University of Technology (Netherlands) published three Bachelor of Science Theses [3–5] that together construct a face recognition system using the Microsoft's Kinect. In their system, the face is cropped from captured images using the Viola-Jones face detection MATLAB library. To fulfil holes in the face image they use image averaging. Later, they use a morphable model technique to extract and code features of the face in normal and geometry images. Finally, to perform recognition, they use different metrics as cosine weighted-angle, Euclidean and Manhattan distances. Although five of six subject were correctly identified by their system, they report a lack of data to perform a formal evaluation of his system.

In comparison to the Delft University of Technology, in this paper, we are presenting a face recognition system that has a novel face detection procedure using only 3D data to evaluate the Eigensurface face recognition approach proposed by Heseltine et al. in 2004 [1], that it is a natural extension of the classical Eigenfaces technique proposed by Turk and Pentland in 1991 [2].

3. Experimental procedure

Our face recognition system has been prototyped using Matlab version 2014 on a Windows 7, 2.3 GHz Intel Core i7 16 GB RAM personal computer.

Then, two graphical user interfaces have been designed to enrol and verify/identified our users and for performance evaluation we have experimented our system within our Human Computer Interaction Laboratory.

Our experimental procedure for face recognition is divided in three main steps: (1) Image acquisition, (2) Image processing, and (3) Recognition.

3.1 Image acquisition

As illustrated in Figure 1, users were asked to stand 85 cm in front of the Kinect sensor, which is the closest distance according to the Kinect's technical specifications. Then, users are requested a front pose toward the camera to capture a testing image with a neutral expression. For this task, the Kinect sensor is manually adjusted at the user's head height.

Over these conditions, RGB and depth images are captured using the Matlab image acquisition tool box for Kinect. Figure 2 shows a sample of RGB and depth images captured with our prototype.

For our experimentation, 29 subjects were enrolled as users of our system and 10 subjects were used as intruders. Twenty training images were captured per user in the same day, while test images were captured during a week and the intruder's images were captured two different days.

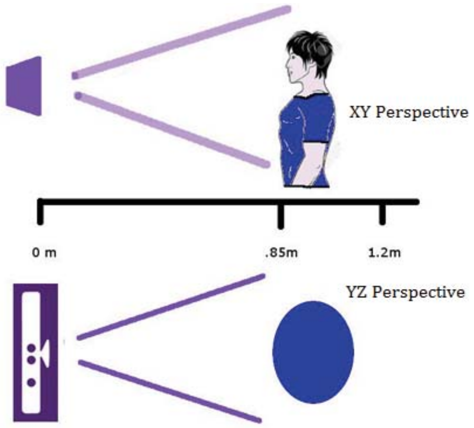
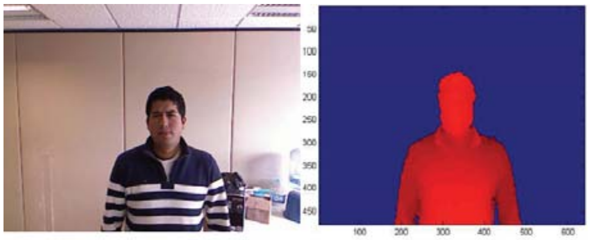


Figure 1. Within our experimental procedure, a subject is positioned 85cm in front of the Kinect camera to be recognised.



a) b)

Figure 2. Image samples acquired by our system when processing a user's verification or identification: (a) RGB image and (b) depth map image.

3.2 Image processing

Although, RGB and depth images are captured, we are using only 3D images within our face detection system, depth maps and 3D point clouds, and both types of images are computed using standard Matlab image acquisition and processing libraries [9]. The overall description of image processing step is shown in Figure 3.

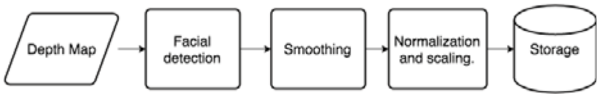


Figure 3. Once a face is detected, the facial area is stored as a face image and then smoothed, normalised and scaled before being stored.

Face detection

An essential step before recognition is face detection. To do this, we are proposing a novel two steps approach as illustrated in Figure 4. First, local maximum values are located in horizontal even slices. By doing this we are expecting to find the nose tip of the face as the minimum depth. As shown in Figure 5, some maximum depth candidates are expected along the sagittal plane. Taking advantage of the subject position in front of the camera, we are specifically analysing that area that according to the Kinects field of view could contain the subject's face.

After that, we are using those minimum depth values to correspond, using the Iterative Closest Point (ICP) algorithm, a Parke's facial model (Figure 6) implemented in another of our related work [12] to the test image. Those minimum depth values are considered as a nose tip candidate. To speed off our correspondence step we are translating the facial model at the nose tip level to every nose tip candidate identified in our previous step (see Figure 7). This prealignment also boost the method's performance.

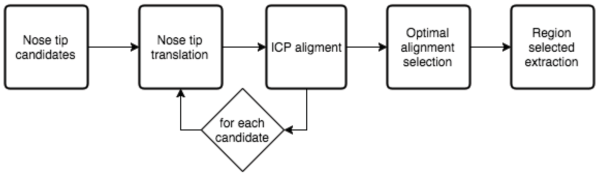


Figure 4. Our face detection approach exhaustively searches for the best alignment.

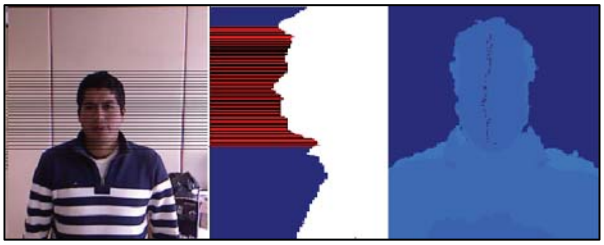
The best correspondence between the facial model and the test image is accounted as the one with the minimum ICP's adjustment error. Based on that correspondence the region of the 3D image is extracted to compute face depth map of the captured user in the scene. To avoid facial gestures, landmarks in Parke's facial model were used to crop the area among the external eye corners and the nose tip in the depth map.

Smoothing

Data loss may be found in this depth map due reflection at image acquisition step and transforming operations between 2D/3D images. These factors could reduce the overall performance of our system, so we add a smoothness step filtering the image with a nearest neighbourhood-based kernel to enhance the image quality [7][9]. Figure 8 shows the difference between a sample of depth map before and after the smoothing.

Normalization

Finally, we normalize depth maps data from 0 to 1 to make depth values relative to users' position, not to the camera's position. All used facial images are resized to a standard size of *[60x80]* pixels, which are used for training and testing our face recognition system.



(a) (b) (c)

Figure 5. Our face detection process is taking advantage on the controlled subject's locations towards the Kinect sensor. (a) The area where the nose is expected to appear, but not necessarily to be the full face within this area; (b) A profile analysis is performed to find the nose tip candidates; (c) The nose candidates found along the sagittal plane.

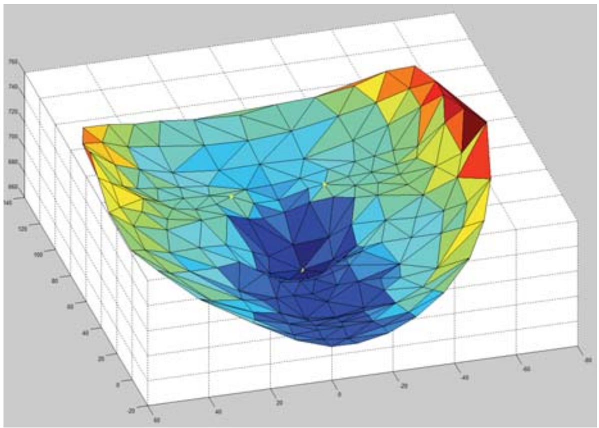


Figure 6. We are using one of our implementations of Parke's facial model to detect the face into captured images [12].

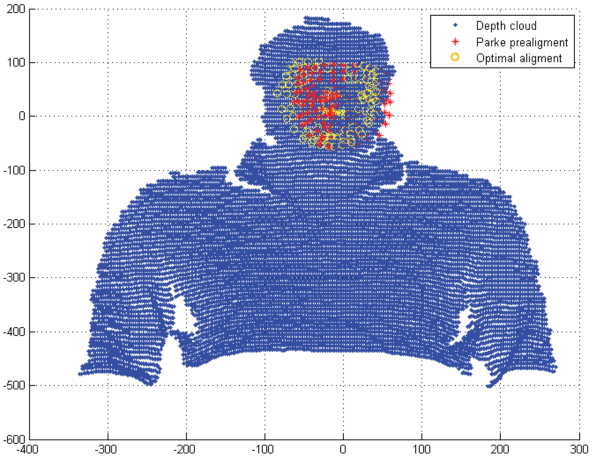


Figure 7. Corresponding a facial model to a test image using the Iterative Closest Point (ICP) algorithm to detect a face into a point-cloud 3D image.

3.3 Recognition

As we have mentioned before, we are investigating an Eigenfaces' direct implementation [2], named Eigensurface [1] for our verification and identification evaluations.

Then, considering a training set of facial surfaces, stored as orientated normalised 60x80 depth maps, represented as vectors of length 4800. We begin by reducing dimensionality to a practical value, while maximising the spread of facial surfaces within the subspace, by application of PCA to the training set of M(580) depth maps $\{\Gamma_1, \Gamma_2, \ldots \Gamma_M\}$, computing the covariance matrix:

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T$$

$$C = AA^T$$

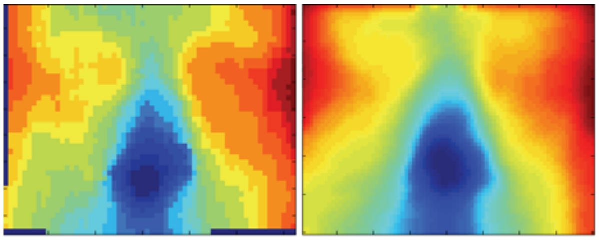
Knowing that:

$$A = [\Phi_1 \Phi_2 \Phi_3 \cdots \Phi_M]$$

$$\Phi_n = \Gamma_n - \Psi$$

$$\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n$$

Where Φ_n is the difference of the *nth* depth map from the average ψ. Eigenvectors and eigenvalues of the covariance matrix are calculated using standard linear methods. The resultant eigenvectors describe a set of axes within the depth map space, along which most variance occurs and the corresponding eigenvalues represent the degree of this variance along each axis. The M eigenvectors are sorted in order of descending eigenvalues and the M' greatest eigenvectors chosen to represent surface space. The greatest eigenvectors are calculated as the sum of the eigenvalues associated to the sorted eigenvectors at 90% of representability. As defined in [1], it is termed each eigenvector as eigensurface, which are displayed as range images in Figure 9.



(a) (b)

Figure 8. A depth map sample, (a) raw face data detected, (b) face data after image processing.

Once the surface space is defined, any image Γ could be projected using the M' more representative eigenvectors:

$$w_k = u_k^T(\Gamma - \Psi); k = 1,2,\dots,M'$$

$$\Omega = [\omega_1, \omega_2, \dots, \omega_{M'}]$$

Where Ω is the projected image in the surface space and $u_{M'}^T$ are the greatest eigenvectors

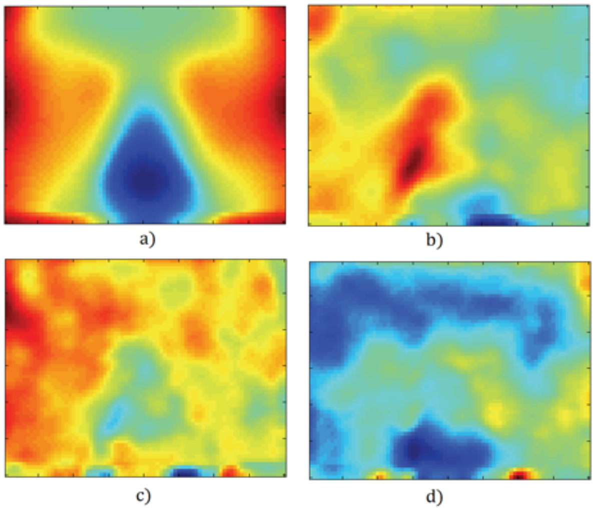


Figure 9. Our experimental face space from 580 3D face images: (a) Mean face; (b) – (d) The first, second and third Eigenvectors.

Training

Before the experimentation, the system has to be trained with the users' data. A depth maps collection per user is called the gallery. For our experimentations, 20 depth maps per user are used to define a face space for each user. Along with the face space, a mean face Ψ is calculated and the gallery is projected to the face space. The collection of the Ω is the projected gallery.

Storage

A biometric database was designed to store the personal and biometric data of all the users (see Figure 10). It was implemented in PostgreSQL and is connected to the graphical user interface (GUI) using the MATLAB's Database Toolbox.

Recognition process

In order to know how *closer* an image is to another in a face space, we have to choose a metric. If an image is closer enough to another, is likely that both are from the same user. In general, the recognition process is straightforward. Given two images, both projected in the same face space, the distance between them is computed. If the distance is lower or equal to a threshold, it is considered that both images are from the same user.

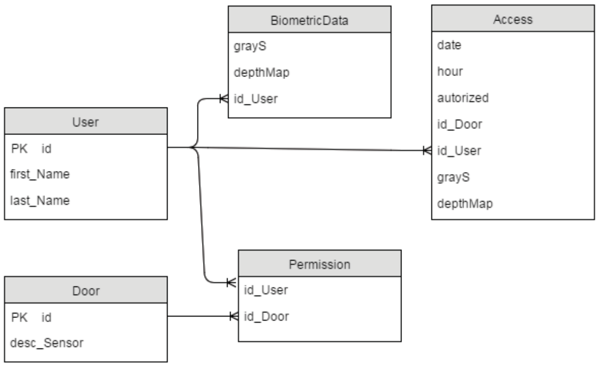


Figure 10. We have designed a database for our face recognition system. In this database, we save any attempt access performed by a subject in addition to our grayscale image gallery.

Metric

We are using the Mahalanobis distance [8] because it is a statistical metric that allows us to compare *one-to-many* images:

$$r^2 = (\Omega_g - \mu)C^{-1}(\Omega_g - \mu)$$

Where μ and C are the centroid and the covariance matrix of the projected gallery, respectively. Ω_g is the image to test. A nice advantage in using Mahalanobis distance, is that we can make our classification process as rigid as we decide, by defining a threshold in number of standard deviations.

4. Performance evaluation

We are evaluating two scenarios in this paper: verification and recognition.

For this experiment, we enrolled 29 users; from every we collected 20 depths maps for training and 19 depth maps for testing. Additionally, we used 150 intruders' depth maps.

4.1 Verification

The first scenario is face verification with the metrics: true acceptance, false acceptance, true rejections and false rejections [10]. Our testing procedure is as follows:

Once the system is initialized, a user asked for access grant into our facility presenting an ID. The system will capture his biometric facial data, if the value of the Mahalanobis distance between the gallery and the detected face is lower than a threshold, the system will grant access to this user. Figure 11 shows the user interface for verification scenario.

In this scenario the user could be an actual user of the system or an intruder, and its performance is measured using a cumulative match characteristic (CMC) curve.

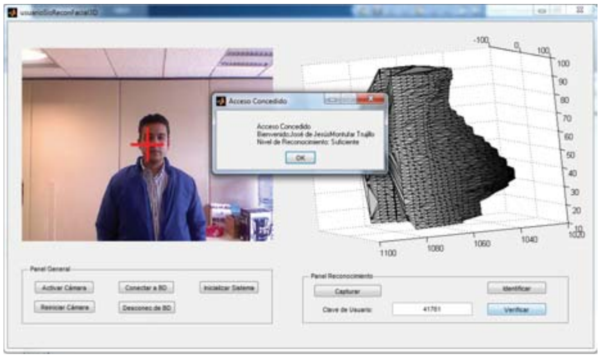
The CMC curve is a tool to summarising the cumulative percentage of correct recognition. In a CMC curve, the horizontal axis shows the ranks, where ranks refers to an ordinal position (from 1 to the size of the given testing set), and the vertical axis accumulates the fraction of the testing images that yield a correct match at every rank [13]. In Figure 12, the results of this experimentation are displayed as a CMC graph.

4.2 Recognition

The second scenario is face recognition. In this case, we have run an evaluation to measure recognition ranks when verifying 29 authorised users and 10 intruders intending to come in into our research laboratory. Our detail experimental procedure is as follow:

In recognition scenario, we use the images used in the verification scenario to measure how far, in terms of Mahalanobis distance, are every image in every user gallery compared to others users galleries. Also, the distances among all images in the users' galleries and 50 images of 10 intruders were computed. So, a *one-to-many* comparison was performed. Results of this recognition scenario are displayed in Figure 13 as a ROC curve graph.

The receiver operating characteristic (ROC) curve, is a tool to summarising the space of possible operating points for a verification system, that is, the space of actually achievable trade-offs in frequencies of the two types of errors [13].



Acceso Concedido

Acceso Concedido
Bienvenido José de JesúsMontufar Trujillo
Nivel de Reconocimiento: Suficiente

Figure 11. The system correctly grants access to a user. Note the detected face, which is displayed at upper-right part.

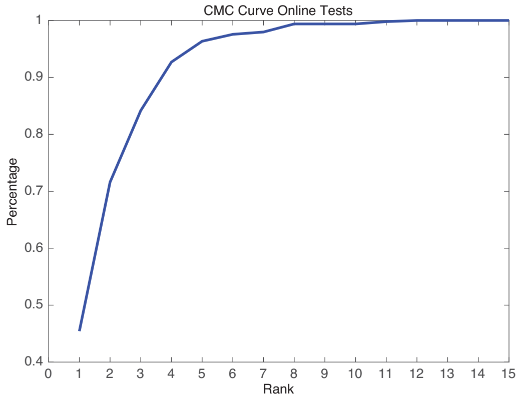


Figure 12. Verification performance evaluation illustrated by a Cumulative Matching Curve with a threshold of 3.

5. Conclusions

In this paper we have introduced our face recognition approach using Eigensurface on Kinect depth-maps.

We have experimentally evaluated our system granting/denying access into our Human Computer Interaction Laboratory in two scenarios, recognition and verification. Our face recognition approach is scoring 45% at first rank and it gets 99% at rank 8. On the other hand, from our ROC curve an EER of 14% is achieved for the verification scenario.

As part of our future work we are investigating more effective face recognition algorithm to achieve a better recognition score taking into account other factors that have been identified in our preliminary experimentation, e.g. Kinect's 3D resolution, face detection and segmentation accuracy, and facial depth map accurate computation. Our final aim is to implement an economical biometric system suitable for small and medium size organizations.

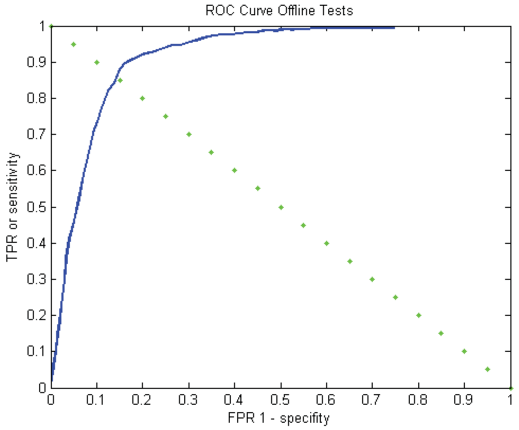


Figure 13. Verification performance evaluation illustrated by a Receiver Operating Characteristic Curve.

Acknowledgements

Authors give a sincere gratitude to the Mexican Public Education Department (SEP) for its partial financial support to carry out this research through the Red-PRODEP 3238 grant. Also, our thanks go to the Research Department of the Autonomous University of the State of Mexico (UAEMex) for their financial assistance, through the grants: 4012/2016RED and 3720/2014/CID.

References

[1] Thomas Heseltine, Nick Pears, Jim Austin (2004). Three-dimensional face recognition: An Eigensurface approach. On the 2004 International Conference on Image Processing.

[2] Matthew Turk, Alex Pentland (1991). Eigenfaces for recognition. Journal of Cognitive Neuroscience, Vol. 3, Number 1.

[3] F.S. Fikke and B. K. Gardiner (2013). 3D Face Recognition (Image Acquisition). Bachelor of Science Thesis. Electrical Engineering, Delf University of Technology.

[4] M.M.J. Gerlach and C. T. Rooijers (2013). 3D Face Recognition (Data Processing: Registration and Deformation). Bachelor of Science Thesis. Electrical Engineering, Delf University of Technology.

[5] M.A. Huijbregts and B. Stobbe (2013). 3D Face Recognition: How to make a fast and reliable database and compare the database with 2D+3D facial input?. Bachelor of Science Thesis. Electrical Engineering, Delf University of Technology.

[6] César Flores-Lovera (2016). Prototipo de un sistema biométrico por reconocimiento facial 3D utilizando el sensor Kinect. BSc Thesis. Engineering Department. Autonomous University of the State of Mexico.

[7] Rafael Gonzalez and Richard Woods (2008). Digital Image Processing. Prentice Hall.

[8] Richard Duda, Peter Hart, David Stork (2001). Pattern Classification. Wiley Interscience.

[9] Rafael Gonzalez, Richard Woods and Steven Eddins (2004). Digital image processing using Matlab. Prentice-Hall.

[10] Davis, J.; Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In proceedings of the 23rd International Conference on Machine Learning.

[11] MacCormick, J. (2016). How does Kinect work? Retrieved from http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf

[12] Serena Mejia (2016). A 3D face model based on Mexican anthropometry. Master by research thesis. Engineering Department. Autonomous University of the State of Mexico.

[13] Romero, M (2010). Landmark localisation in 3D face data. PhD Thesis, Department of Computer Science, The University of York, UK.

[14] Paduano J. (2014). Una aplicación para generar modelos faciales 3D de una profundidad utilizando el sensor Kinect y el lenguaje de programación abierto processing. BSc Final Dissertation, Autonomous University of the State of Mexico.

[15] Marcelo Romero, Juan Paduano, Vianney Muñoz (2014). Point-Triplet Spin-Images for Landmark Localisation in 3D Face Data. In proceedings on the IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications. Rome, Italy.

[16] Juan Paduano, Marcelo Romero, Vianney Muñoz (2015). Toward face detection in 3D data. In proceedings on the International Conference on Image Processing, Computer Vision, and Pattern Recognition. Las Vegas, NE, USA.

[17] Pears, Nick; Liu, Yonghuai; Bunting, Peter (2012). 3D Imaging, Analysis and Applications. Springer, UK.

[18] Jarfi R; Arabnia Hamid R. (2009). A Survey in Face Recognition. Journal of Processing Systems Vol. 5, No. 2.

[19] FBI (2011). Fingerprints and other Biometrics. US Congress Press.

[20] Phillips, J., Wechsler, H., Huang, J., & Rauss, P. (1998). The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Computing*, 296-306.

[21] Phillips, P. (2003). Overview and Summary of Face Recognition Vendor Test 2002. DARPA.

[22] Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., & Marques, J. (2005). Overview of the Face Recognition Grand Challenge. *IEEE Conference on Computer Vision* and Pattern Recognition.

[23] Zhao, W; Chellapa, R; Phillips, P.J; Rosenfeld, A. Face Recognition: A literature Survey. (2003). ACM Computing Surveys, Vol. 35, No. 4, pp. 399-458.

[24] Wayman, J; Jain, A; Maltoni, D; Mario, D. (2005). Biometric Systems. Technology, Design and Performance Evaluation. Springer. London.

[25] Bowyer, K; Chang, Kyong; Flynn, P. (2006). A survey of approaches and challenges in 3D and multi-modal 2D + 3D face recognition. ELSEVIER. Computer Vision and Image Understanding 101 pp. 1-15

[26] Pears, N; Liu Y; Bunting, P. (2012). 3D Imaging, Analysis and Applications. Springer. London.

[27] Bronstein, A; Bronstein, M; Kimmel, R. (2004) Three-Dimensional Face Recognition. International Journal of Computer Vision. Springer.

[28] Spreeuwers, L. (2015) Breaking the 99% Barrier: Optimization of three-dimensional face recognition. IET Biometrics.

[29] Chang, K; Bowyer, K; Flynn, J. (2006). Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expression. IEEE Transactions on Pattern Analysis and Machine Learning, Vol 28, No. 10.

[30] Cybula Ltd (2016). FaceEnforce: facial biometric system. England. Retrieved from www.cybula.com

SVM Feature based Tail-Lamp Pairing for Localization of Vehicle at Night-time

Yeongyu Choi[1], **Hyojin Lim**[1], **Ju H. Park**[2], **Ho-Youl Jung**[1*]
[1]Department of Information and Communication Engineering,
[2]Department of Electrical Engineering,
Yeungnam University, Gyeongsan, Republic of Korea
[*]Corresponding Author: Prof. Ph. D. Ho-Youl Jung(hoyoul@yu.ac.kr)

Abstract – *Vision based night-time vehicle detection has been an emerging research field in Advanced Driver Assistance Systems (ADAS) such as Adaptive Driving Beam (ADB), Autonomous Emergency Braking (AEB). In this paper, we propose an efficient tail-lamp pairing algorithm at night-time by using Support Vector Machine (SVM). First, thresholding method is applied to extract tail-lamp candidates in gray image. Then, nine geometrical and stochastic features are extracted from each blob. Next, the tail-lamps are classified by using SVM. And then, the tail-lamps in the same horizontal line are compared by using the same features that are extracted in the detection step. Finally the two nearest tail-lamps are paired. In the experimental results, the proposed method shows that its performance is higher than that of cross correlation method in term of night-time vehicle localization.*

Keywords: Driver assistance system, Support vector machine, Tail-lamp detection, Tail-lamp pairing.

1 Introduction

For safer night-time driving, a good visibility of the road is essential. Most of drivers want to use high beams while driving at night-time to observe the road clearly. However, the high beams including glaring, dazzling, blinding that cause a discomfort to the drivers who are driving in the opposite side of the host vehicle. In fact, lots of drivers forget to control the high beam. This is one of the reasons of car accidents in night-time driving. It is necessary to develop an automatic high beam control system that provides both convenience and safety at night-time driving for drivers. According to [1, 2], automatic high beam control facilities have been developed. Using the positions of surrounding moving vehicles, high beams are able to control to obtain the wide angle beam. This beam does not strike directly into regions of the oncoming and preceding vehicles. Thus, the drivers can obtain the best field-of-view by using the proposed intelligent high beam control system. Support vector machine (SVM) is the one of many learning-based methods [3]. SVM classification is described generally in [4, 5]. In this paper, we introduce an efficient processing of tail-lamp pairing at night-time environment. Firstly, we convert the image from RGB space to YCbCr space. The YCbCr is useful because saturated white pixels of center of tail-lamp appear in the Y channel. Moreover the scattered red pixels appear in Cr channel. Therefore, the YCbCr space is suitable for detection of tail-lamps. Secondly, we apply thresholding in the gray image by using global fixed value. Camera exposure time is fixed also. Thirdly, after thresholding step, many tail-lamp candidates are classified by using the SVM. The top-down tree structure is used for the SVM classifier [3]. The features for the SVM are calculated by utilizing geometrical and stochastic computations such as blob area, aspect ratio, maximum value, mean value and standard deviation in a bounding box. These extracted features are also used for tail-lamps pairing. The proposed algorithm saves processing time because the pairing step reuse the features that are used in classification step. In addition, the proposed algorithm shows performance is higher than that of a symmetry comparison based cross correlation in [6, 7].

2 System overview

The night-time vehicle detection system involves the processing steps of thresholding, labeling, feature extraction, classification and pairing. Firstly, binary image is obtained from grey scale image using Y channel. The binary image is obtained from thresholding step by using the global fixed value. An example of binary image is shown in Fig. 1 (b). After thresholding step, the lamp candidates are labeled and the bounding boxes of every blob (with more than 9 pixels) are obtained. In Fig. 2, the main flow chart of the proposed method is described. This algorithm is an improvement of our previous work [8].

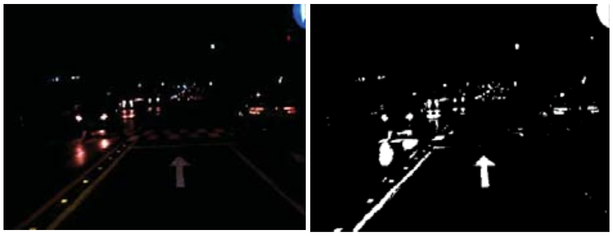

(a) (b)

Fig. 1 (a) Original input image, (b) Binary image.

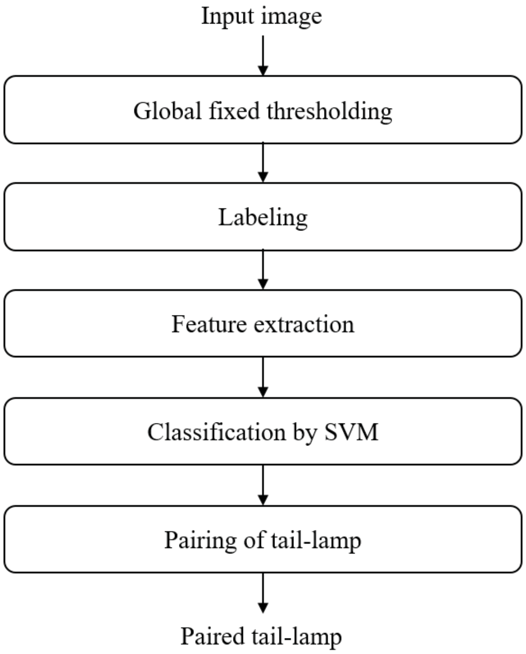


Fig. 2 Main flow chart of proposal system

2.1 Feature extraction

For feature extraction, nine feature types are extracted from YCbCr frame. Some of feature extractions are proposed in [8].

First feature: The y-coordinate of centroid point of every candidate is calculated as follow:

$$blob_{y\_centroid} = \frac{\sum blob_{y\_coordinate}}{blob_{area}} \tag{1}$$

Second feature: The ratio feature is obtained by using an area of the blob and bounding box. This ratio value is calculated as follows:

$$blob_{arearatio} = \frac{blob_{area}}{(blob_{width} * blob_{height})} \tag{2}$$

Third feature: The degree of the square shape of the bounding box is obtained by

$$blob_{squarishness} = \frac{2 * blob_{area}}{{blob_{width}}^2 + {blob_{height}}^2} \tag{3}$$

The fourth and the fifth features are extracted by using the maximum pixel value of the blob in Y channel and Cr channel.

The sixth feature and seventh features are extracted by using the mean value of the blob in Y channel and Cr channel. This value is computed as follows:

$$blob_{mean} = \frac{\sum pixel_{val}}{blob_{area}} \tag{4}$$

The eighth feature and ninth features are extracted by using the variance value of the blob in Y channel and Cr channel. This value is computed as follows:

$$blob_{standarddeviation} = \frac{\sum (pixel_{val} - blob_{mean})^2}{blob_{area}} \tag{5}$$

Every feature value is normalized to the range of [0..1]. Notice that these features are used for both SVM classification and tail-lamp pairing.

2.2 Classification

The top-down tree structured multiclass SVM is applied. This structure was developed and used in our previous work [8]. It is constructed as a binary tree which is shown in Fig. 3. From the top-down tree, we can decide how many classifiers that are needed. Originally, the proposed top-down tree is used to classify various types of blobs. However, in this paper, we consider tail-lamps only, so the other classes are not examined. In details, only 4 classifiers are considered for classification of tail-lamps.

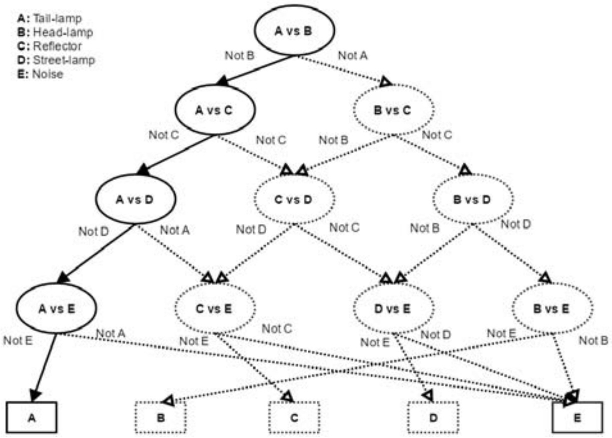


Fig. 3 Top-Down tree structured SVM classifier.

2.3 Tail-lamp pairing

Tail-lamps pairing step firstly considers more than two candidates in the same horizontal line. Next, two tail-lamp candidates will be considered as a pair of tail-lamps if their aspect ratio value belong to the range of [3..8] [9]. The paired tail-lamps have the similar shapes otherwise, the tail-lamp candidates are considered as noise.

For evaluation of similarity of two candidates, we calculate dissimilarity score noted as DS_i, where i=1, 2, 3... 9 which is the number of features for classification. The features extracted in the classification step are reused in this step. The dissimilarity score is calculated by the absolute value of differences of features between two blobs in a paired candidate.

Since the y-coordinate values of the centroid points of blobs exceed one, then the first feature is normalized as

$$DS_1 = \frac{|centroid_{y\_left} - centroid_{y\_right}|}{max_{y\_axis} - min_{y\_axis}} \qquad (6)$$

where max_{y_axis} and min_{y_axis} are the max and the min of y-coordinate value of two blob candidates. Fig. 4 shows an example of blob candidate

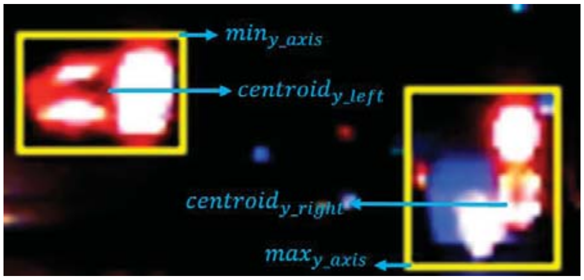


Fig. 4 Explanation of blob candidate.

Other eight features are computed by

$$DS_i = |feature_{i\_left} - feature_{i\_right}| \quad (i = 2, 3 \dots 9) \qquad (7)$$

Finally, total dissimilarity score (TDS) is computed by

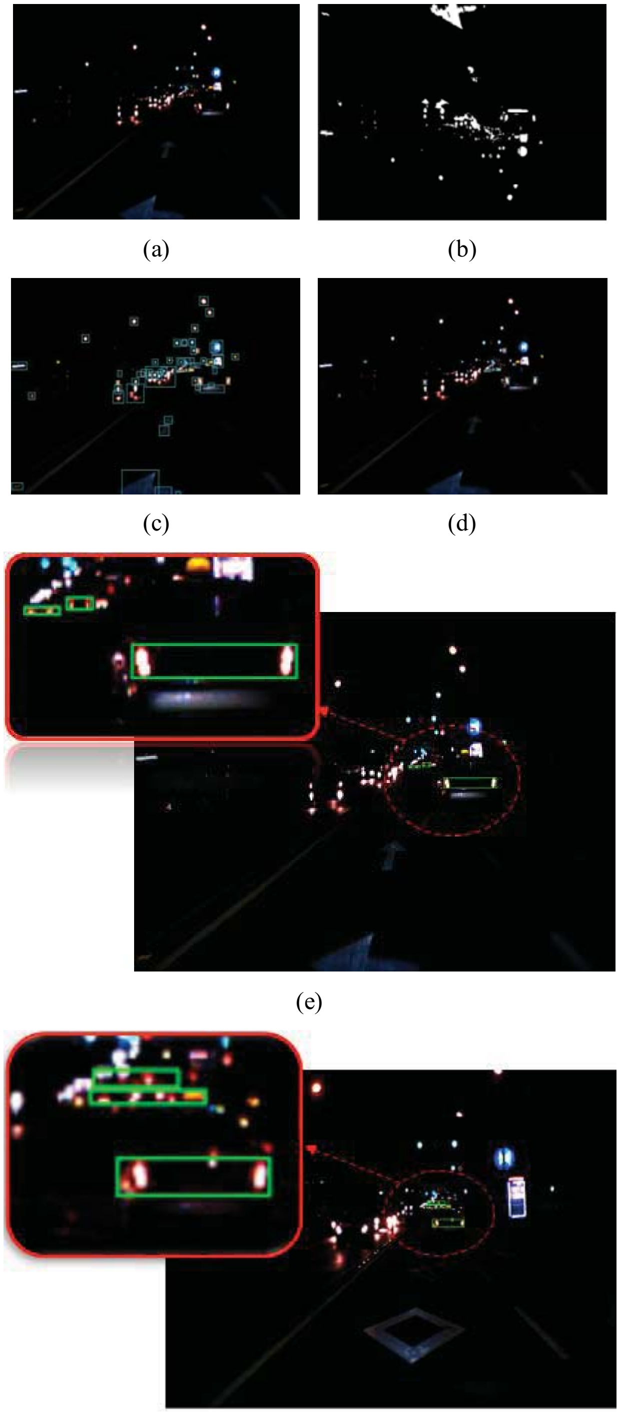
$$TDS = \sum_i DS_i \qquad (8)$$

TDS is calculated for every possible candidate pair of tail-lamps. The candidate pair having the smallest TDS are determined as a pair. If TDS is greater than the pre-determined threshold value, the pair is not made. In the simulation, the threshold value for TDS is experimentally determined as 1.5.

3 Experimental results

For the evaluation, 400 frames that are captured from urban where various light sources exist. The frames are resized to 640x480. Thresholding value is set experimentally to 30. Minimum size of blob candidate is 9 pixels. The result of segmentation is shown in Fig. 5 (c). Fig. 5 (d) describes that the number of blob candidates is decreased after classification. Fig. 5 (e) shows some paired tail-lamps

The performance of the proposed method and the cross correlation based pairing method [6, 7] are compared. The cross correlation value is used as γ = 0.85 [7]. The simulation video includes 1,011 ground truth data. The performances are shown in Table 1.



(a) (b)

(c) (d)

(e)

(f)

Fig. 5 (a) Original input image, (b) Binary image,
(c) Detected candidates,
(d) Classified tail-lamps,
(e) Correct detected pair of tail-lamp,
(f) Incorrect detected pair of tail-lamp.

Table 1. The performances of the proposed method and cross correlation.

Algorithm	Cross correlation	Proposed method
TP	781	834
FP	96	102
FN	230	177
Accuracy	70.55%	74.93%

4 Conclusions

In this paper, we proposed an efficient tail-lamp pairing algorithm at night-time. The performance of the proposed pairing method is higher than that of the conventional cross correlation method, in terms of accuracy. The processing time is reduced by reusing the features of feature extraction step for the classification and pairing steps. However, as shown in Fig. 5 (f), the proposed algorithm has some limitations. When the tail-lamps appear as merged tail-lamp or other light candidates at a binary image, the candidates could not be classified to tail-lamps. Therefore, the tail-lamps could not be paired. If tail-lamp detection step is implemented more robust, the proposed pairing method will have better pairing performance.

Acknowledgements

This work was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2014H1C1A1073236) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011- 0011096).

5 References

[1] Julien Rebut, Benazouz Bradai, Julien Moizard, and Adrien Charpentierhi. “A monocular Vision Based Advanced Lighting Automation System for Driving Assistance”. IEEE International Symposium on Industrial Electronics, Seoul Korea, pp. 311-316, 2009.

[2] P.F. Alcantarilla, L.M. Bergasa, P.Jimenez, M.A. Sotelo, I. Parra, D. Fernandez. “Night Time Vehicle Detection for Driving Assistance Light Beam Controller”. IEEE Intelligent Vehicles Symposium, Eindhoven Netherlands, pp. 291-296, 2008.

[3] Platt, John C., Nello Cristianini, and John Shawe-Taylor. “Large Margin DAGs for Multiclass Classfication”. Vol. 12, pp. 547-553, 1999.

[4] C.-W. Hsu and C.-J. Lin. "A Comparison of Methods for Multiclass Support Vector Machines". IEEE Transactions on Neural Networks, Vol. 13, No. 2, pp. 415-425, 2001.

[5] B. Zhao, Y. Liu and S.W. Xia. “Support vector machines and its application in handwritten numerical recognition”. International Conference on Pattern Recognition, Vol. 2, pp. 720-723, 2000.

[6] O’Malley, Ronan, Edward Jones and Martin Glavin. “Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions”. IEEE Transactions on Intelligent Transportation Systems, Vol. 11, No. 2, pp. 453-462, 2010.

[7] O’malley, R. M. Glavin and E. jones. “Vision-based detection and tracking of vehicles to the rear with perspective correction in low- light conditions”. IEEE Intelligent Transport Systems, Vol. 5, No. 1, 2011.

[8] Hyojin Lim, Heeyong Lee, Ju H. Park and Ho-Youl Jung. “Night-time Vehicle Detection Based on Multi-class SVM”, IEMEK J. Embed. Sys. Appl., Vol. 10, No. 5, pp. 325-333, 2015.

[9] Noppakun Boonsim and Simant Prakoonwit. “An Algorithm for Accurate Taillight Detection at Night”. International journal of Computer Applications, Vol. 100, No. 12, 2014.

CREAK: Color-based Retina Keypoint Descriptor

Yi-An Chen, Chia-Hsin Chan, and Wen-Jiin Tsai, *Member*, *IEEE*
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Abstract – *Effectively feature matching between images is key to many computer vision applications. Recently, binary descriptors are attracting increasing attention for their low computational complexity and small memory requirement. However, most binary descriptors are based on intensity comparisons of grayscale images and did not consider color information. In this paper, a novel binary descriptor inspired by human retina is proposed, which considers not only gray values of pixels but also color information. Experimental results show that the proposed feature descriptor spends fewer storage spaces while having better precision level than other popular binary descriptors. Besides, the proposed feature descriptor has the fastest matching speed among all the descriptors under comparison, which makes it suitable for real-time applications.*

Keywords: image matching, feature descriptor, keypoint descriptor, human retina, color information

1 Introduction

A great number of computer vision applications, like image search, image recognition, object tracking and image classification depend on describing particular feature points over an image. In order to represent feature points efficiently, applying robust and stable feature descriptor is necessary. However, how to make descriptor more invariant to geometric and lightning transformations while requiring low computation complexity and small amounts of memory is a big challenge. Therefore, many approaches are developed over the last decades.

The most well-known descriptor is Lowe's SIFT [1] feature descriptor which is floating-point based and provides invariance to a variety of common image transformations, but the disadvantages of SIFT are expensive cost of computation and storage. SURF [2] proposed by Bay *et. al.* is designed to improve performance of SIFT and can use less computational time to achieve similar matching rates compared to SIFT. Despite SURF is much faster than SIFT, however it is still impracticable in many real-time applications, such as embedded devices and mobile phones.

In recent years, several binary descriptors have been proposed. Unlike float-point based descriptors which need to represent image information with local gradient histogram, binary descriptors, in contrast, provide the gray value comparison around detected feature points in the image patch, and then image patch information is encoded with a fixed size binary string. Since binary descriptor use Hamming distance and XOR operation for measuring similarity between two descriptors, they can significantly decrease computational time. The performance of binary descriptors can reach as well as float-points ones, while reducing computational costs and memory requirements.

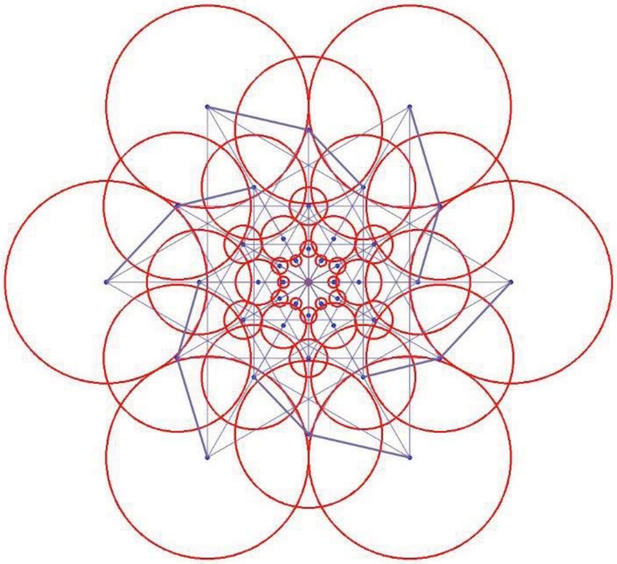


Fig. 1. The proposed CREAK orientation pairs. The lines denote the orientation pairs of FREAK and the bold lines denote the additional pairs for the proposed CREAK descriptor.

For the state-of-the-art binary descriptors, such as BRIEF [4], ORB [6] and FREAK [3], when encoding information of image patch, they only perform gray value comparisons around feature points in the image patch. However, the important information of color is ignored. Besides, their sampling pairs take only the gray value at single pixel into account and therefore sensitive to noise. In order to solve this problem, these binary descriptors offer some alternative smoothing operations before the pixel value are sampled. Although smoothing image can reduce some of noise-sensitive problems, the side effect is that smoothing image will also decrease the details of image and lead to information loss.

Inspired by the above observations, in order to make descriptor more robust and discriminative, the color information is also necessary. In this paper, we propose an alternative binary descriptor named *CREAK (Color-based REtinA Keypoint descriptor)* which is based on the FREAK descriptor and inspired by the photoreceptive cells over the retina, by comparing pixel lightness intensity and color information of pixel rather than single pixel intensity to mimic the retina of human eye. Experiments results show that the proposed feature descriptor not only preserves the ability to fast matching but also spends only less than half size of FREAK descriptor while having similar or even better matching rate than not only FREAK but also other state-of-the-art binary descriptors.

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 describes the proposed method and its implementation. Section 4 evaluates the performance of proposed method with other descriptors and the result of real matching situation. Finally, the conclusion and future works are given in Section 5.

2 Related Work

The feature descriptors can be generally categorized into two groups: one is float-point based descriptor and another is binary descriptor. SIFT [1] is the most popular float-point based descriptor in the last decade and it presents a highly descriptive power and powerful robustness against to a variety of image transformations. First, SIFT uses sequences of DoG (Difference-of-Gaussians) functions to identify potential features that are invariant to rotation and scale, then it computes a grid of oriented gradient histograms to store the descriptor into a 128-dimensional vector. Several float-point based approaches were proposed to improve performance of SIFT, SURF [2] by Bay *et. al*. is a successful one. The computation time of SURF is faster than SIFT, while its matching performance is close to SIFT's by representing features with the responses of Haar wavelets for approximating gradient orientations in the SIFT. However, SURF belongs to float-based descriptor group, it still relies on floating-point calculations to measure Euclidean distance between two descriptors, which increases time to match features across different images and make descriptors impracticable in real-time applications or low-power devices.

Another group of descriptors is called binary descriptor which were proposed to overcome the shortcomings of float-point based descriptors. Recently, binary descriptors are attracting increasingly attention due to their advantages. They calculate Hamming distance and employ fast XOR operation to measure of distance between two binary descriptors. This makes binary descriptors become more faster matching speeds than float-point ones. Furthermore, by using no more than 512 bits, a single binary descriptor requires far less space than SIFT or SURF. BRIEF [4], the first binary descriptor to describe image features achieves great speed acceleration by simply computing the gray value comparisons of random test pairs in the region of interest. Unfortunately, since simple pixel-based test pair is highly sensitive to noise or other change in local appearances, BRIEF is not robust enough to geometric and lightning transformations especially in rotation variation.

According to BRIEF's method, Rublee *et. al*. proposed the ORB [6] descriptor. By estimating first order moments within the patch, ORB can invariant to rotation. It also selects highly uncorrelated pixel pairs for binary test instead of random selected test pairs of BRIEF. Another approach different from ORB and BRIEF is the BRISK [7] proposed by Leutenegger *et. al*. which emphasizes locality by computing intensity differences between two short-distance or long-distance pixels in a predefined concentric sampling pattern. For computing the patch orientation BRISK uses pixel pairs with large distances while building binary descriptor with short ones.

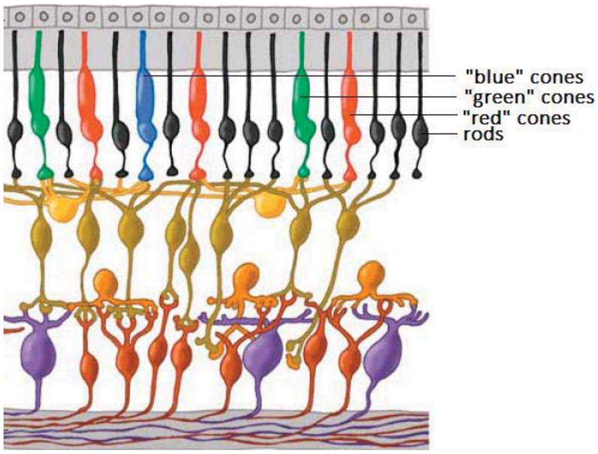
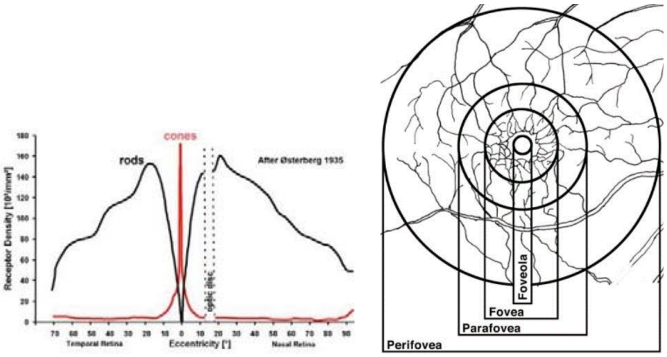


Fig. 2. Cells in the human retina are arrayed in discrete layers [16]



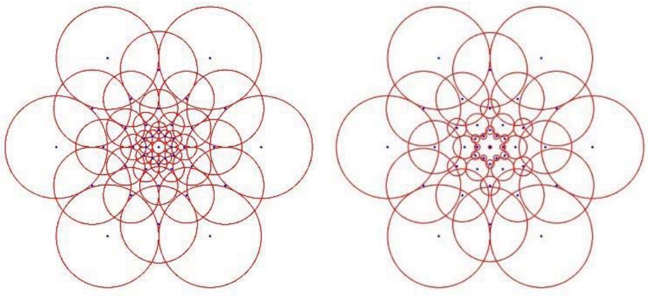
(a) (b)

Fig. 3. (a) Topography of the layer of rods and cones in the human retina [17]. (b) Human retina areas [16].

FREAK (Fast REtinA Keypoint descriptor) [3] is similar to BRISK. It also describes feature point with predefined concentric sampling pattern which was inspired by the retina patterns of human eye. Differing from BRISK, FREAK samples more points exponentially in the inner area to mimic fovea of retina. Besides, using the same learning method of ORB, FREAK also chooses an optimal set of sampling pairs.

Usually, most of binary descriptors perform serval smoothing operations before the pixel pairs are sampled, in order to handle noise-sensitive problem. However, this method also decreases spatial information of image patch. Recently, Gil Levi and Tal proposed the LATCH [8] descriptor which extracts more spatial information from image patch in each descriptor's bit by comparing pixel patches instead of individual pixel value.

Unlike the state-of-the-art binary descriptor based on gray-scale image, and with same concept as LATCH, our CREAK descriptor also extracts more spatial information in the image rather than single pixel value or pixel patch, and compares luminance as well as color information of pixel instead of single pixel intensity to make the descriptor more robust to noise.



(a) (b)

Fig. 4. Comparison of (a) FREAK and (b) the proposed CREAK sampling pattern.

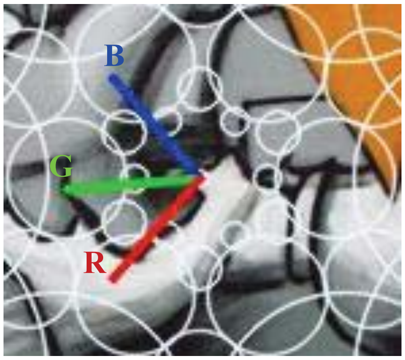


Fig. 5. The same feature point will correspond to three different sampling points (for RGB channels) after orientation estimation.

3 The CREAK Descriptor

3.1 Motivation

According to the research of neuroscience, the retina of human eyes plays a key role in the human visual system (HVS). The purpose of the retina is to receive light, convert them into neural signals, and send these signals to the brain for visual recognition [15]. The retina contains two types of photoreceptive cells: rods and cones. Rod cells have very low spatial resolution but are highly sensitive to light, so they are responsible for the information of illuminance and they are not present in the fovea region. In contrast, cone cells have very high spatial resolution but are relatively insensitive to light, they are primarily located in the fovea region and give us the ability to distinguish colors.

Current understanding that cones also can be subdivided into blue cones, green cones, and red cones based on three different response curves. The topography of the layer of rods and cones in the human retina is shown in Figs. 2 and 3. Consequently, the proposed descriptor is designed by simulating the topology and photoreceptive cells distribution of the retina to describe the features of an image, as described in the next subsection.

3.2 Sampling pattern

In order to design a sampling pattern of binary descriptor which is similar to the human retina, we referenced and modified the FREAK's sampling pattern which is also inspired by the human retina. The characteristic of FREAK's sampling pattern is that the size of its receptive fields (the size of circles in Fig. 4(a)) mimics the density of ganglion cells, which grows exponentially with the distance toward the center of the retina. Furthermore, it also makes receptive fields of the pattern overlapping each other, this can increase spatial redundancy and bring descriptor more discriminative power.

FREAK performs Gaussian smoothing on the sampling points with variable blur kernels according to its receptive field size, however, it is designed for gray level. For color information, the density of the cone cells which have a higher spatial resolution, is higher close to the center of retina than elsewhere, as shown in Fig. 3(a). But large receptive fields adopted in FREAK will decrease the spatial resolution due to the large smoothing area. In order to support color information in the proposed descriptor, we resize each blur kernel to be smaller, especially for the respective fields corresponding to fovea and foveola areas [16] as shown in Fig. 3(b), to simulate the photoreceptive cells distribution of the retina [17]. The comparison of sampling patterns adopted in FREAK and the proposed CREAK is shown in Fig. 4(b).

3.3 Orientation pairs

Since we reduce the receptive field sizes considering the newly added color information, it may decrease the descriptor tolerance for some homographic transforms, such as rotation and scaling. After investigating the orientation pairs in FREAK, we observe that: (i) the angles between feature point center and the orientation pairs are limited to several specific angles, (ii) there are less pairs in perifovea area than fovea and foveola areas, and (iii) the pairs are linked in the same layers. Therefore, we proposed to add 12 *cross-layer orientation pairs* in the perifoveal area, making the proposed CREAK descriptor having 57 pairs, while FREAK have 45 pairs, as shown in Fig. 1. These pairs can not only generate more angles to improve the tolerance of rotation, but also utilize the inter layer information of receptive fields to improve the tolerance of scaling.

For the orientation estimation, we use the same method as that of FREAK which estimates local gradients over selected pairs

$$O = \frac{1}{M}\sum_{Po\in G}\left(I(P_o^{r1}) - I(P_o^{r2})\right)\frac{P_o^{r1}-P_o^{r2}}{\|P_o^{r1}-P_o^{r2}\|} \qquad (1)$$

where M is the number of pairs in the set of all the pairs used to compute the local gradients G and $P_o^{r_i}$ is the 2D vector of the spatial coordinates of the center of receptive field.

We perform the orientation computation separately for each color channel because even with the same orientation pairs, different color channels could have different gradients and that means the same feature point will correspond to three different sampling points after orientation estimation. By doing this, more spatial information can be retrieved for same feature point to increase the performance. An example is shown in Fig. 5.

3.4 Building the Descriptor

We construct our descriptor by performing tests for the intensities of the predefined test pairs, which is a common binary descriptor construction process. Let $I(P_a^1)$, $I(P_a^2)$ denote the smoothed intensity of the pair $P_a = (P_a^1,\ P_a^2)$, the binary test function $\mathrm{T}(P_a)$ is formulated as

$$T(P_a) = \begin{cases} 1, & if\ I(P_a^1) \geq I(P_a^2) \\ 0, & otherwise \end{cases} \tag{2}$$

For the proposed CREAK descriptor which consists of three color channels, the binary tests are performed for each channel. Then, the complete binary descriptor D of size N is formed by concatenating three N/3 binary test results and defined as

$$\mathrm{D} = \sum_{i=0}^{\frac{N}{3}-1} \left[2^i T(B_i) + 2^{i+\frac{N}{3}} T(G_i) + 2^{i+\frac{2N}{3}} T(R_i) \right] \tag{3}$$

where B, G, R represent color channels blue, green, and red, while B_i, G_i, R_i are color test pairs of receptive fields with their corresponding channels, respectively. For example, if the total descriptor size N = 192, then the number of bits used by each color channel will be N/3 = 64.

In order to choose a set of test pairs that is best for describing the feature point, we employ the same training method in FREAK, but separately applied for each color channel. The training process consists of the following steps:

1) For each feature point, compute a descriptor composed of all possible test pairs. Create a matrix M whose rows are associated to the feature point and columns are associated to all the possible test pairs.
2) Compute the means of each column in M.
3) According to ORB [6], a higher variance is desired in order to produce a discriminant feature, and the mean value of 0.5 will have the highest variance for a binary distribution. Therefore, the matrix columns are sorted by the absolute value minus 0.5.
4) Keep the best column and iteratively add columns having low correlation with the selected columns.

In the proposed descriptor, test pairs are selected by training from approximately 500k feature points which are drawn from images in the PASCAL 2006 dataset [14]

4 Experimental results

The proposed CREAK descriptor have been implemented in C++ and integrated into OpenCV 3.1 for performance evaluations. The experiments are conducted following the evaluation framework presented in [12]. The framework consists of applying Gaussian blur, brightness change, rotation, and scale change to each image from the Oxford datasets proposed by Mikolajczyk and Schmid [11]. All the experiments are executed using a Desktop PC with an Intel i7 3.4 GHz processor and 12 GB of RAM.

4.1 Color Space Selection

Firstly, we measure the performance of using different commonly used color spaces, including YCrCb, Lab, and RGB. The image graffiti [11] with 12 different levels of blur, 85 levels of brightness, 73 levels of rotation degree, and 71 levels of scales are adopted in the experiments and the results in terms of matching correctness ratio (i.e., # of correct matches divided by # of matches) are shown in Table. 1. It is observed that when using the same bit-length of descriptors, RGB color space obtains the best performance on average. We also compare different descriptor lengths for RGB color space, and observe that a descriptor with length more than 192 bits does not make apparent improvement. Therefore, 192 bits (24 bytes) of descriptor length and the RGB color space are recommended and used for the proposed CREAK descriptor in the later experiments of this paper.

TABLE 1. Average matching correctness ratio comparison for various color spaces

	Blur	**Brightness**	**Rotation**	**Scale**
Lab-192	69.86%	97.63% (2nd)	93.98%	93.76%
YCrCb-192	69.21%	96.09%	92.66%	95.13%
RGB-192	70.60% (1st)	97.54%	94.07% (2nd)	95.38% (2nd)
RGB-384	70.54% (2nd)	97.66% (1st)	94.57% (1st)	96.84% (1st)

4.2 Comparison with different descriptors

In this section, the performance of the proposed CREAK descriptor is compared with a wide range of binary feature descriptors available in OpenCV, including BRIEF [4], BRISK [7], ORB [6], FREAK [3], and LATCH [8], with their default parameters. The same feature point (keypoint) detector is employed for a fair comparison of descriptor performance. The feature detector proposed in ORB is adopted in our experiments, due to its good performance and high speed. The experiment results are shown in Figs. 6-9 for test case "graffiti 1". It is observed that except for the blur transformations, in all testing conditions the proposed CREAK descriptor presents the most robust performance, compared to other binary competitors. For blur transformations, CREAK has lower performances than BRIEF and LATCH, and has comparable performances to ORB. However, both BRIEF and LATCH have extremely low performances for scale transformations, while ORB and BRIEF have bad performances for rotation transformations. A real matching case example is provided in Fig. 10.

Furthermore, we also compare the proposed CREAK descriptor with state-of-the-art floating-point based descriptors including SIFT and SURF with their default feature detectors, as shown in Figs.11-14. The experiment results show that CREAK achieve the better performance among all.

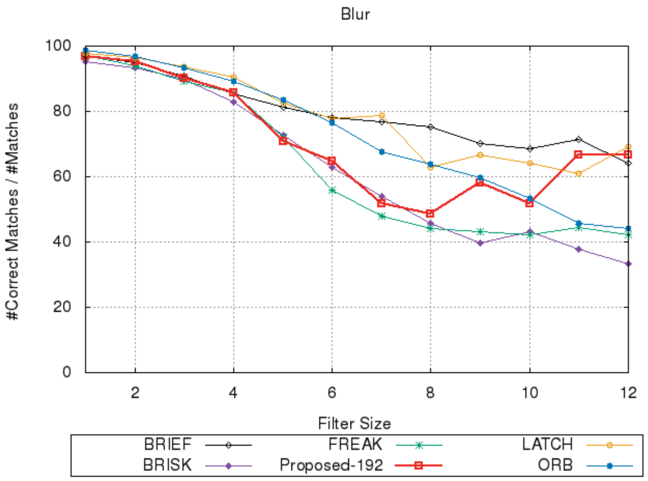


Fig. 6. Performance comparison of binary descriptors under blur transformations.

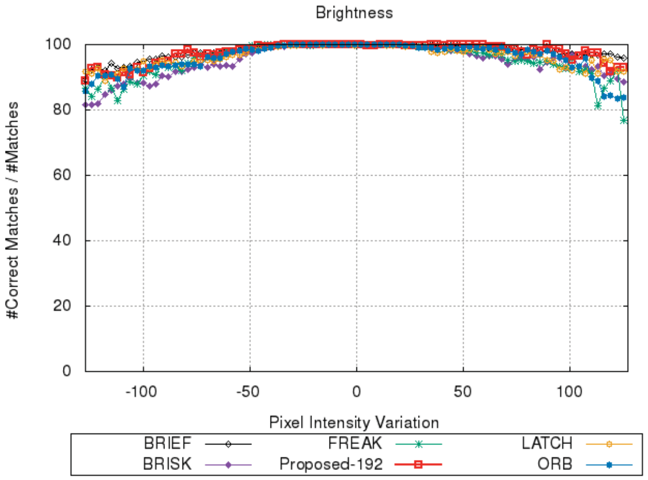


Fig. 7. Performance comparison of binary descriptors under brightness transformations.

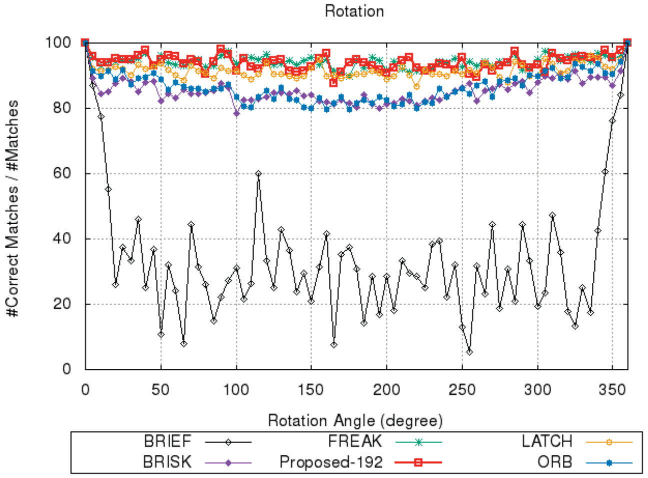


Fig. 8. Performance comparison of binary descriptor under rotation transformations.

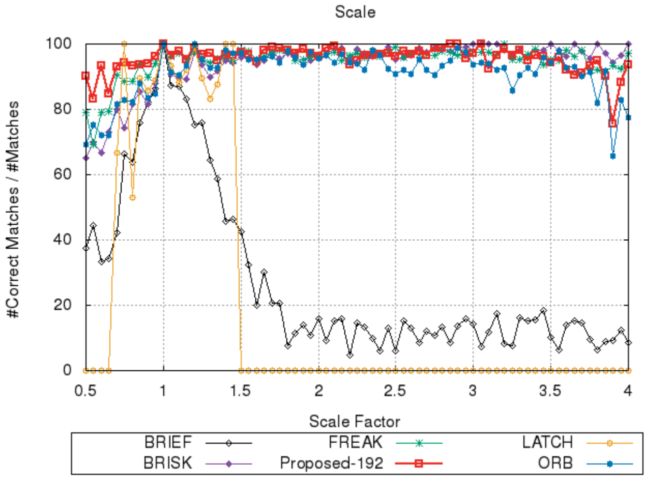
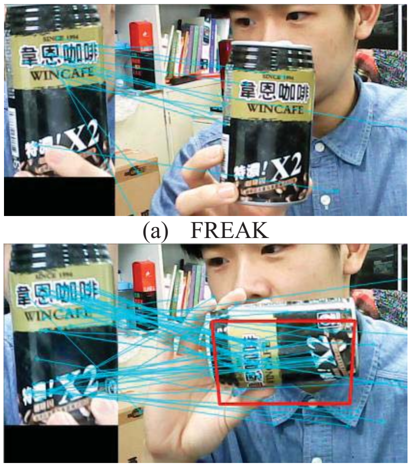


Fig. 9. Performance comparison of binary descriptors under scale transformations.



(a) FREAK

(b) CREAK

Fig. 10. A real matching case example. (a) FREAK fails for the simple upright matching test while (b) the proposed CREAK matches successfully even for the object under 90-degree rotation with view point change.

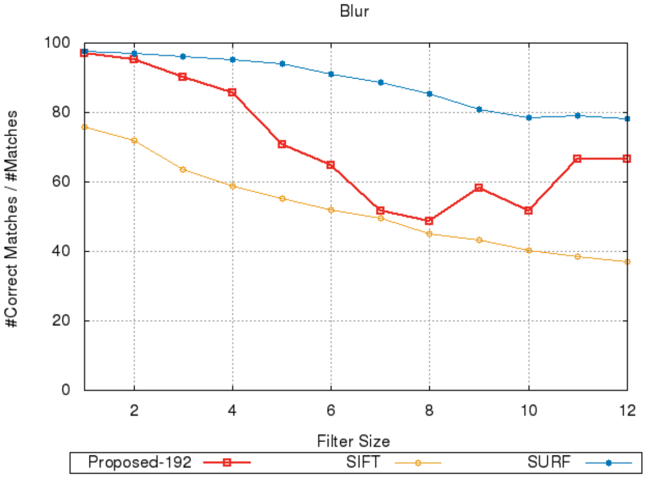


Fig. 11. Comparison of float-point based descriptors under blur transformations.

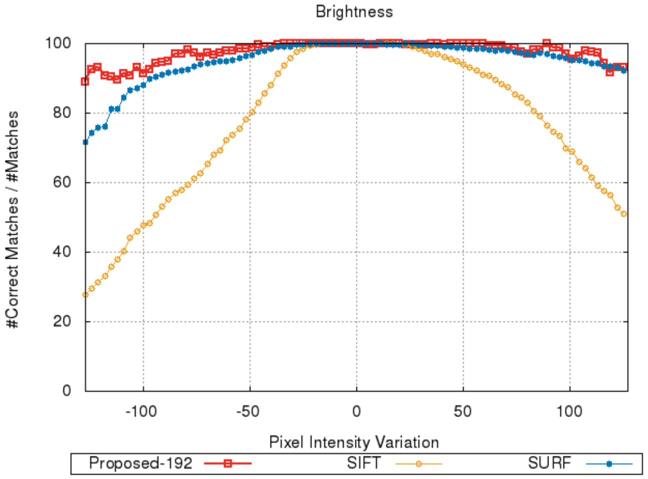


Fig. 12. Comparison of float-point based descriptors under brightness transformations.

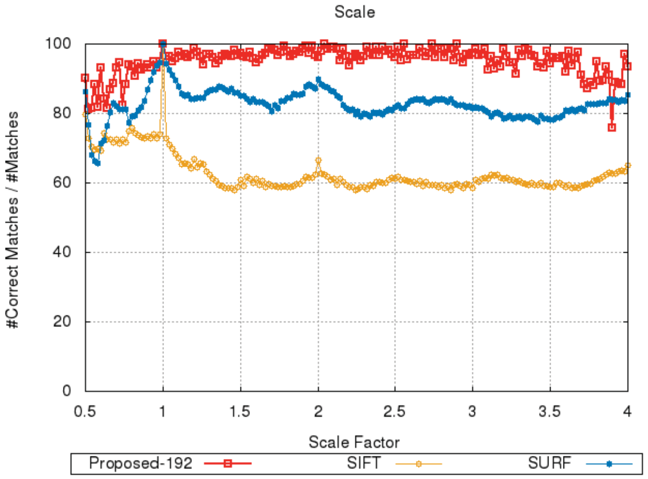


Fig. 13. Comparison of float-point based descriptors under scale transformations.

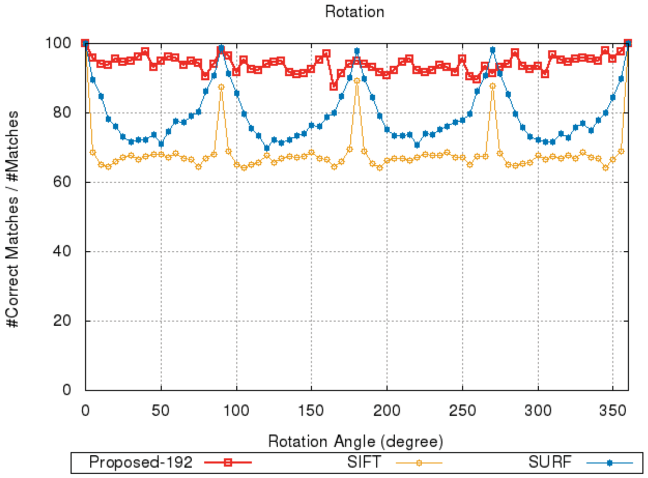


Fig. 14. Comparison of float-point based descriptors under rotation transformations.

TABLE 2. Performance comparison of computation times (in milliseconds) for different feature descriptors

Descriptor	Description	Matching	Total
SIFT	0.553	4.14	4.69
SURF	0.089	2.04	2.13
BRIEF	**0.005**	1.24	1.25
BRISK	0.013	1.53	1.54
ORB	0.024	1.17	1.19
FREAK	0.018	0.62	0.64
LATCH	0.048	1.31	1.36
Proposed	0.026	**0.32**	**0.35**

TABLE 3. Performance comparison of storage requirements (in bytes) for different feature descriptors

Descriptor	Storage
SIFT	512
SURF	256
BRIEF	64
BRISK	64
ORB	32
FREAK	64
LATCH	32
Proposed	**24**

Tables 2 and 3 show the comparison of computation time and storage requirement, respectively, for the descriptors. It is observed that the proposed CREAK descriptor not only requires the least storage space, but also achieves the least total processing time. Considering the computation time and storage requirement, we believe that the proposed CREAK descriptor will be more suitable than other descriptors for the applications requiring real-time feature matching.

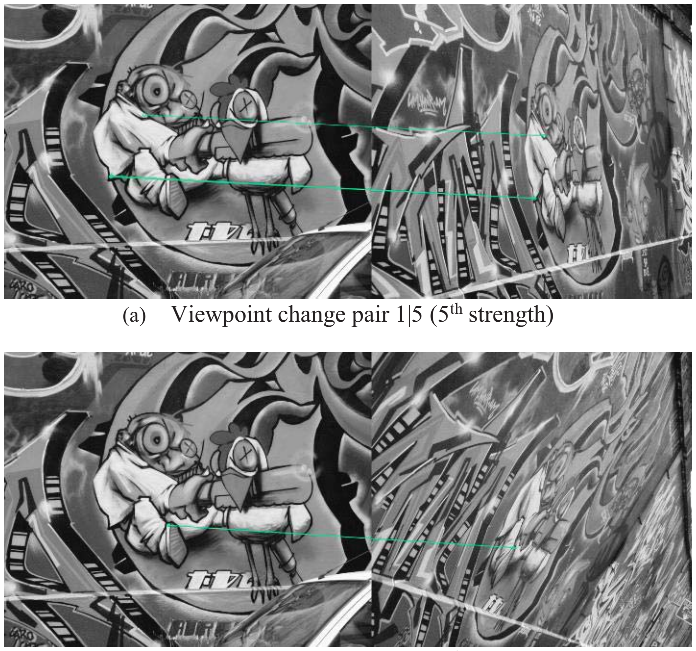
In summary, compared to our predecessor, FREAK, CREAK achieves apparent improvements for both blur and brightness transformations, having comparable performances for rotation and scaling, running at about two times faster, while saving more than half of the storage spaces.

4.3 Extremely matching case examples

Moreover, we also display two extremely matching case examples using the graffiti image from [11] with homography matrices to verify inlier matches, as shown in Table 4. Comparing with ORB, CREAK can match two more features whereas FREAK matched none in pair 1|5. It also brings us a great accomplishment that when we apply the most harder image pair 1|6, CREAK matched two features correctly, however, for other descriptors, even the SIFT and SURF, there is no match obtained. The matched feature points by using CREAK are shown in Fig. 15(a) for pair 1|5 and Fig. 15(b) for pair 1|6.

TABLE. 4. Number of match features for graffiti 1|5 and 1|6

	#Match(1|5)	**#Inlier(1|5)**	**#Match(1|6)**	**#Inlier(1|6)**
ORB	**18**	2	**16**	0
FREAK	5	0	10	0
CREAK	12	**4**	7	**2**



(a) Viewpoint change pair 1|5 (5th strength)

(b) Viewpoint change pair 1|6 (6th strength)

Fig. 15. Matched points generated by the proposed CREAK descriptor for the test case "graffiti" under different view changes.

5 Conclusion

In this paper, a novel binary descriptor is presented, which is inspired from human retina. According to the distribution of photoreceptive cells over the retina, more precisely, rods and cones, we comparing color values of the pixels around the feature point instead of the pixel gray value with our sampling pattern based on FREAK's retina sampling pattern. Experimental results show that, the proposed descriptor has better recognition rate than other widely-used binary descriptors even compared with SIFT or SURF, while having low requirements especially in storage. Besides, the matching time of the proposed method outperforms other descriptors due to it have only 192 test pairs, that makes it more suitable for visual algorithm requires real-time performance. Our future work will continue make improvement of feature descriptor based on the principle of the human retina. Last but not least, there are still space for improvement for the blur transformation case for the proposed descriptor. We will also make more effort to study how to improvement the performance for blur transformation.

ACKNOWLEDGMENT

This research is supported in part by MOST-103-2221-E-009-129-MY2 of the Ministry of Science and Technology, Taiwan, R.O.C.

6 References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l Journal of Computer Vision*, vol. 60, issue 2, pp. 91-110, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *European Conf. Computer Vision*, pages 404–417. Springer, 2006.

[3] A. Alahi, R. Ortiz, and P. Vandergheynst., "Freak: Fast retina keypoint," *Computer Vision and Pattern Recognition*, pp. 510–517, 2012.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua., "Brief: Binary robust independent elementary features," *In European Conf. Comput. Vision*, pp. 778–792, 2010.

[5] E. Tola, V. Lepetit, and P. Fua., "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 32, issue 5, pp. 815–830, 2010.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski., "Orb: an efficient alternative to sift or surf," *Int'l Conf Comput. Vision*, pp. 2564–2571, 2011.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart., "Brisk: Binary robust invariant scalable keypoints," *Int'l Conf Com-put. Vision*, pp. 2548–2555, 2011.

[8] Gil Levi and Tal Hassner., "LATCH: Learned Arrangements of Three Pat*ch Codes," IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, March, 2016

[9] Matheus A. Gadelha, Bruno M. Carvalho., "DRINK: Discrete Robust INvariant Keypoints," *Int'l Conf on Pattern Recognition (ICPR)*, 2014

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool., "A comparison of affine region detectors," *Int'l Journal of Comput. Vision*, vol. 65, pp. 43–72, 2005

[11] K. Mikolajczyk and C. Schmid., "A performance evaluation of local descriptors," *Trans. Pattern Anal. Mach. Intell*., 27(10):1615–1630, 2005

[12] I. Barandiaran, C. Cortes, M. Nieto, M. Graña, and O. Ruiz., "A New Evaluation Framework and Image Dataset for Key Point Extraction and Feature Descriptor Matching," *Int'l. Conf. Computer Vision Theory and Applications* , (VISAPP), 2013

[13] K. Mikolajczyk and C. Schmid., "An affine invariant interest point detector," *Computer Vision* (ECCV), 2002

[14] M. Everingham., "The Pascal Visual Object Classes (VOC) Challenge,"[Online].Available:http://pascallin.ecs.soton.ac.uk /challenges/VOC/databases.html

[15] Helga., "Kolb How the Retina Works", *American Scientist*, 2003

[16] A. Hendrickson., "Organization of the Adult Primate Fovea," *Macular Degeneration,* 2005

[17] G. Osterberg, "Topography of the layer of rods and cones in the human retina," *Acta ophthalmologica., Supplementum 6*, Levin & Munksgaard, Copenhagen, 1935.

Hyperspectral vision control of environmental impacts in civil works

G. Sorrosal[1], L. Solabarrieta[1], J.I. Larrauri[1], C.E. Borges[1], and A. Alonso-Vicario[1]
[1]Deusto Institute of Technology – DeustoTech Energy, University of Deusto, Unibertsitate Etorbidea 24,
48007 Bilbao, Spain

Abstract - *This work aims to develop a system for the automation of the actuation protocols against possible environmental impacts generated by civil works. This system involves from the detection, control and physic tracking of the impact to the generation of alarms and reports to facilitate the decision making. This work is part of the LIFE+ research project VisionTech4Life funded by the European Commission. The automated actuation protocol is based on inspection flights done by a drone equipped with a hyperspectral system. Processing the captured images, it will be possible to detect the presence of certain common contaminants in civil works, reducing both detection times and laboratory tests costs.*

Keywords: Hyperspectral vision, drone, environmental impacts, remote sensing

1 Introduction

The evaluation of the environmental impacts in civil works is an important aspect to integrate into the decision tools of the civil works. The Environmental Assessment (EA) includes technical and scientific studies to determine the repercussion for the environment of a specific work. The EA studies are regulated by a complex regulatory framework from European Commission, States and Autonomous Communities and they have to include an environmental surveillance plan. The objective of the surveillance plan is to identify the affected systems, the type of impact and to select indicators to measure and follow those impacts. The indicators should be few, easily quantifiable and representative of the affected system.

Currently, to control for example the impact in water flows near the civil works, laboratory analytics have to be done. In the best case, these analytics are done with a defined periodicity to determine the water quality at the sampling time. This procedure is beginning to automate driven by the European politics and legislation, but is still not the common practice in civil works.

The use of image sensors can be a solution for the automation of the surveillance plan. Multispectral or hyperspectral technology allows the measure of some environmental indicators. These indicators can be the presence of certain water contaminants (e.g. some types of hydrocarbons) or the water stress of the surrounding vegetation (outside or inside the perimeter of civil works) that are not possible to detect in the visible region of the electromagnetic spectrum. This technology is able to perform an automatic, online and *in situ* inspection of the civil works, without doing laboratory destructive analytics.

In the case of aqueous contaminants, it has been demonstrated the possibility of using multispectral and hyperspectral vision systems for monitoring the water quality as well as for the detection of the water stress of the vegetation with successfully results [1]. Several works support the inspection of the water turbidity using vision systems at different wavelengths in the visible spectrum (400 nm, 440 nm, 490 nm, 670 nm) and in the near infrared (NIR) spectrum (880 nm) [2-5]. Other dangerous contaminants of aquatic mediums are the oil and hydrocarbons [6]. The Chemical Oxygen Demand (COD) and Colored Dissolved Organic Matter (CDOM) indicators can be used to test the water quality. Again, the wavelengths used to detect the presence of these contaminants and other quality indicators are located in the visible and NIR spectrums [7-10].

For the detection of water stress in vegetation induced by the civil works, the commonly used indicators are the Normalized Difference Vegetation Index (NDVI) and the Photochemical Reflectance Index (PRI). These indices can be successfully calculated in the visible and NIR spectrum [11-15].

The majority of these works are based on aerial and satellite images. One of the problems of those systems, based on airplane or satellite images, is that it is not possible to program more or less automatic inspections with a high flight frequency.

This work aims not only to detect some indices but also to develop a complete system for the automation of the actuation protocols against possible environmental impacts generated by civil works. This system involves from the detection, control and physic tracking of the impact to the generation of alarms and reports to facilitate the decision making. The proposed method is based on a drone equipped with a hyperspectral system to carry out the inspections of the civil works looking for leaks and other environmental impacts induced by the civil works. The drone flies over the works periodically and the obtained images are processed to detect the presence of certain contaminants.

The final objective is to develop supporting tools for the environmental monitoring plan for a preventive and efficient control of the environmental impacts in civil works, accelerating the environmental control processes and actuation protocols against the produced impacts. It will also contribute to implement the principle of preventing the affection of the environment, against the corrective methods.

2 Material and methods

The monitoring task is carried out periodically with a drone (octocopter AEROT8 XL model) equipped with a hyperspectral camera and a mounted acquisition system. These inspections are based on automatic flights done at different areas of the work and following predefined flight plans. The images are continuously captured and the obtained spectral image hypercubes are corrected automatically during the flight. The light normalization of the spectral images is done using dark and white references taken before every inspection flight. After each flight, the images are downloaded and processed by computer vision algorithms in order to detect any contaminant leak or any other environmental impact in the civil work area. The obtained information about the environmental impacts is stored in the database developed to manage all the possible impacts taking place in civil works.

The hyperspectral system used in the inspection flights is a Micro-Hyperspec® VNIR A-series lineal camera (Headwall Photonics) with a Schneider-Kreuznach lens of 10 mm of focal distance. This camera works in the 400 to 1000 nm spectral range and it has a spectral resolution of less than 2 nm (325 spectral bands). The acquisition system mounted in the drone is an ARIES single board computer with an Intel quad core E3845 processor at 1.9 GHz with 2 GiB of DDR3 SDRAM. Figure 1 shows the used equipment; the drone and the mounted camera with its acquisition system. The octocopter has been specifically modified for this project in order to provide greater power to the motors to be able to flight equipped with the vision system. Moreover, as the lifting power of the motor has been increased, it has been necessary to develop a battery system capable to supply the hyperspectral camera, its acquisition system and the drone itself, for reasonable flying times.

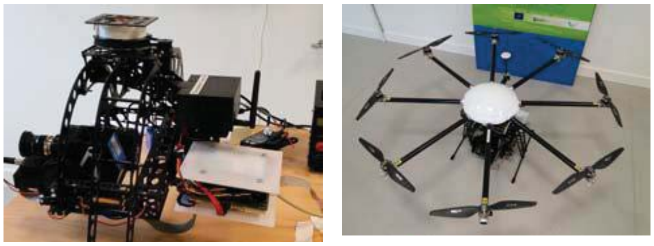


Figure 1: AEROT8 XL drone and on board vision system.

The first pilot tests have been carried out in the expansion works of the wastewater treatment plant of Villapérez (Asturias, Spain). Figure 2 shows the general plan of the sewage treatment plant where the pilot tests have been carried out. The inspection flights over the Nora River (at the top of the map), were performed to study and control possible contaminant leaks. The water stress can be analyzed from the surrounding trees and other vegetation near the river. And the presence of other contaminants can be studied in the moved soils and dumps.

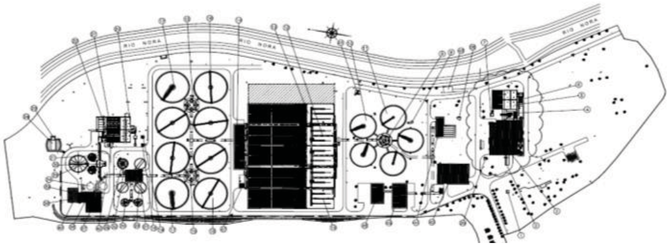


Figure 2: General plan of the sewage treatment plant in Villapérez (Asturias, Spain).

Finally, some contaminant leaks such as hydraulic lubricants and some hydrocarbon types were simulated in order to test the viability of the proposed inspection system.

3 Results

Figure 3 (left) shows a Google Earth image of a part of the civil work area, chosen to capture hyperspectral images for this research. And Figure 3 (right) shows the images obtained with the hyperspectral system. As the camera captures linear frames, the images are a little bit distorted due to the drift of the capture angles. As the goal is an early-warning system, the correction of the image distortion is not necessary. This simplifies the necessary equipment and reduces the weight on board the drone. Therefore in this project it is enough with the obtained images to identify if there is a contaminant impact or not.

Figure 3 (right) contains two containers with controlled pollutants, marked as P1 (pollutant 1: agricultural petrol) and P2 (pollutant 2: synthetic oil). And there is also a water pool marked as WP. Spectral profiles of a random point of each sample have been drawn in Figure 4. The differences between the water pool and the pollutants are evident along the entire spectrum.

The first results obtained processing manually the captured images from only few inspection flights, demonstrate that this technology is capable to observe the spectral signature of the different contaminant elements.

4 Conclusions

A methodology to control the environmental impacts of the civil works based on multispectral images is

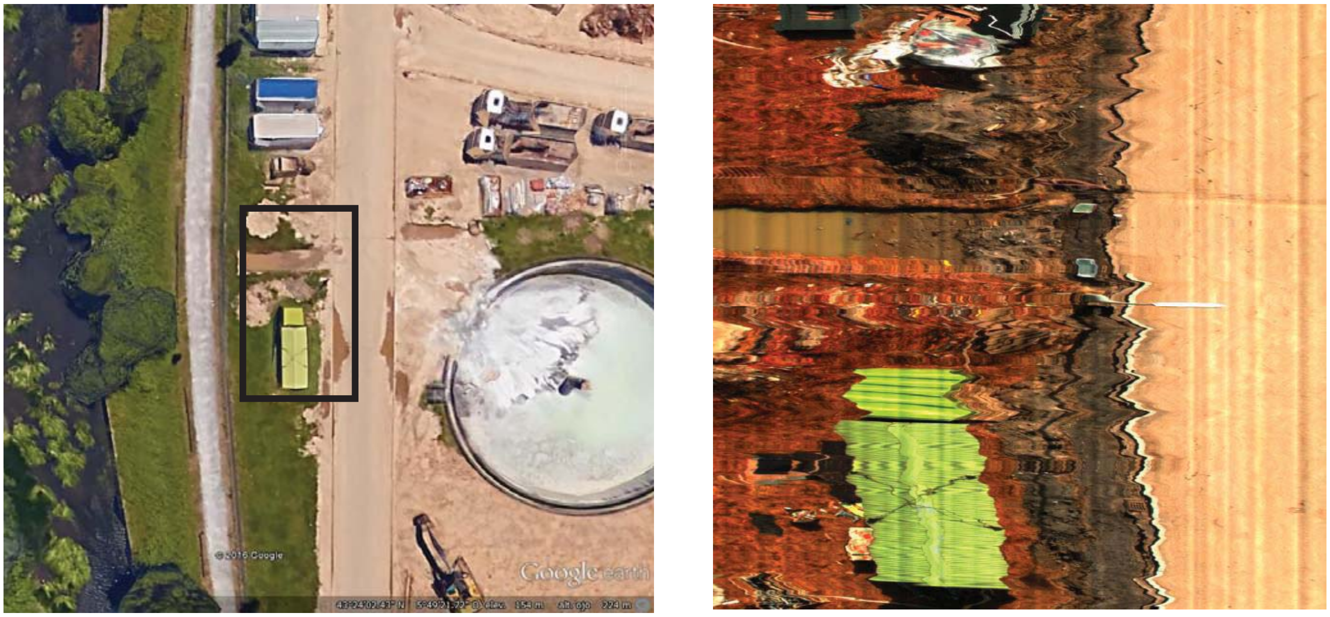


Figure 3: (Left) Google Earth image for a part of a wastewater treatment plant in Villapérez (Asturias). (Right) Combination of 3 spectral images obtained with the hyperspectral camera equipped in the drone.

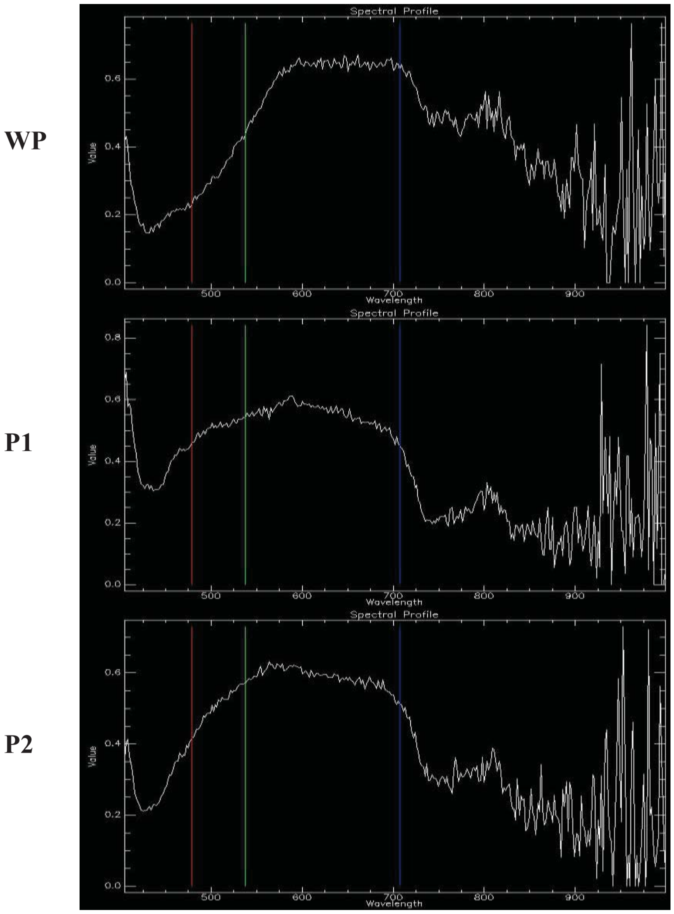


Figure 4: Spectral profiles of water pool (WP), pollutant 1 (agricultural petrol) and pollutant 2 (synthetic oil). Coloured vertical lines (red, green and blue) are the selected spectrum wavelengths to compose RGB images as Figure 3 (right).

presented in this work. As explained in the previous paragraphs, only the prototype design and implementation, and the first tests have been done. But with only these tests, different spectral signatures of controlled contaminants have been identified. These observations together with the excellent quality and accuracy of the images taken flying at 20 and 35 meters, suggest that these results will be promising and perfectly usable in civil works to automatically detect possible environmental impacts.

The next step will be the implementation of the computer vision algorithms necessaries to automatically process the images captured in the drone inspections. The final objective is to develop an automatic inspection system to detect the possible leaks and other environmental impacts and to automatically generate the inspection reports in order to carry out the containment necessary actions. Finally, a secondary objective is to study the possibility of reducing the necessary spectral bands to perform the inspection minimizing the equipment costs with specifically selected multispectral cameras for each environmental impact.

Acknowledgment

The research team would like to thank the financial support of the European Commission by the research project VisionTech4Life (LIFE12 ENV/ES/001163).

5 References

[1] "LIFE+ Project AG_UAS (LIFE09/ENV/ES/0456)." 2009.

[2] L. M. Goddijn and M. White, “Using a digital camera for water quality measurements in Galway Bay”, Estuar. Coast. Shelf Sci., vol. 66, no. 3–4, pp. 429–436, Feb. 2006.

[3] D. G. Bowers, D. Evans, D. N. Thomas, K. Ellis, and P. J. l. B. Williams, “Interpreting the colour of an estuary”, Estuar. Coast. Shelf Sci., vol. 59, no. 1, pp. 13–20, Jan. 2004.

[4] D. Doxaran, R. C. N. Cherukuru, and S. J. Lavender, “Use of reflectance band ratios to estimate suspended and dissolved matter concentrations in estuarine waters”, Int. J. Remote Sens., vol. 26, no. 8, pp. 1763–1769, Apr. 2005.

[5] R. N. Fraser, “Hyperspectral remote sensing of turbidity and chlorophyll a among Nebraska Sand Hills lakes”, Int. J. Remote Sens., no. March 2013, pp. 37–41, 2010.

[6] R. del’Papa Moreira Scafutto, and C.R. de Souza Filho, “Quantitative characterization of crude oils and fuels in mineral substrates using reflectance spectroscopy: Implications for remote sensing”, Int. J. Appl. Earth Obs. Geoinf., vol. 50, pp. 221-242, 2016.

[7] N. Saito, K. Takizawa, and T. Kurokawa, “Interference-enhanced imaging for detecting oil layer floating on the water”, Sensors Actuators A Phys., vol. 109, no. 3, pp. 195–201, Jan. 2004.

[8] M. Sensor, “Real-time Detection of Oil Slick Thickness Patterns with a Portable Multispectral Sensor”, Bureau of Safety and Environmental Enforcement – Department of the Interior, U.S., 2012.

[9] W. Yang, J. Nan, and D. Sun, “An online water quality monitoring and management system developed for the Liming River basin in Daqing, China”, J. Environ. Manage., vol. 88, no. 2, pp. 318–25, Jul. 2008.

[10] L. A. Dombrovsky, S. S. Sazhin, S. V Mikhalovsky, R. Wood, and M. R. Heikal, “Spectral properties of diesel fuel droplets”, Fuel, vol. 82, pp. 15–22, 2003.

[11] L. Suárez, P. J. Zarco-Tejada, G. Sepulcre-Cantó, O. Pérez-Priego, J. R. Miller, J. C. Jiménez-Muñoz, and J. Sobrino, “Assessing canopy PRI for water stress detection with diurnal airborne imagery”, Remote Sens. Environ., vol. 112, no. 2, pp. 560–575, Feb. 2008.

[12] Y. Kim, D. M. Glenn, J. Park, H. K. Ngugi, and B. L. Lehman, “Hyperspectral image analysis for water stress detection of apple trees”, Comput. Electron. Agric., vol. 77, no. 2, pp. 155–160, Jul. 2011.

[13] P. J. Zarco-Tejada, J. a. J. Berni, L. Suárez, G. Sepulcre-Cantó, F. Morales, and J. R. Miller, “Imaging chlorophyll fluorescence with an airborne narrow-band multispectral camera for vegetation stress detection”, Remote Sens. Environ., vol. 113, no. 6, pp. 1262–1275, Jun. 2009.

[14] P. J. Zarco-Tejada, V. González-Dugo, and J. a. J. Berni, “Fluorescence, temperature and narrow-band indices acquired from a UAV platform for water stress detection using a micro-hyperspectral imager and a thermal camera”, Remote Sens. Environ., vol. 117, pp. 322–337, Feb. 2012.

[15] J. A. J. Berni, P. J. Zarco-tejada, L. Suárez, V. González-dugo, and E. Fereres, “Remote sensing of vegetation from UAV platforms using lightweight multispectral and thermal imaging sensors”, International Society for Photogrammetry and Remote Sensing, Working Groups, 2009.

MLP Neural Network Based Approach for Facial Expression Analysis

Maryam Pourebadi[1], **Masume Pourebadi**[2]

[1] Department of Computer Science, Kent State University, Kent, Ohio, USA

[2] Department of Robotic Engineering, AU-TNB, Tehran, Iran

Abstract - *Human Facial Expression Recognition and Analysis has been an active topic in computer science for over two decades. In recent years, several different approaches have been proposed for developing methods of automatic facial expression analysis. However there are numerous researches on facial image analysis, the performance of expression recognition is still not acceptable due to the variety of human expression and enormous variations in facial images. In this paper an improved method for facial expression analysis by multi-layer perceptron (MLP) is presented. The proposed method combines modified facial action units for emotion recognition by MLP. The system also applies principle component analysis for dimension reduction. To evaluate, the proposed approach is tested on two databases which are, the Cohn-Kanade facial expression and the facial expression recognition FER-13 databases. Method has compared to the related works and experimental results clearly demonstrate the efficiency of the proposed algorithm.*

Keywords: Emotion analysis, Facial Expressions, FACS, Geometric features, Social robotics, Multi-layer perceptron (MLP).

1 Introduction

Facial expression analysis has become an important research area in the entertainment industry and the robotics. To understand human emotions, facial expressions play a major role along with both speech annotations (including silence durations and tone variations) and non-verbal communications such as hand gestures and head motions [33]. Facial expressions and the tone of speech are universal, and have innate characteristics [2]. During conversation, people scan the facial expressions of other persons to get a visual cue of their emotion. At the same, in social robotics, it is essential for robots to analyze facial expressions and express a subset of human emotions to have a meaningful human-robot interaction.

Based on psychological theories [3], human emotions are classified into six basic emotions: surprise, fear, disgust, anger, happiness, and sadness [9]. In addition, there are many composite emotions and transitions between emotions that require continuous facial-expression analysis. Facial expressions research and analysis has been used to simulate and identify human emotions.

An automatic facial expression recognition system generally comprises three crucial steps [4]: face detection, facial feature extraction, and facial expression classification. Face detection is a pre-processing stage to detect or locate the face regions in the input images [3]. Facial feature extraction attempts to find the most appropriate representation of facial images for recognition, and mainly there are two approaches: geometric features-based systems and appearance features-based systems. In this paper we use geometric feature-based method which extracts shapes and locations of facial components information including mouth, eyes, eyebrows, nose to form feature vectors. Nevertheless, the geometric features-based systems [7, 8] require the accurate and reliable facial feature detection.

In Facial expression classification step as the last step of a facial expression recognition system, a classifier is employed to identify different expressions based on the extracted facial features. The most known classifiers used for facial expression recognition are the K-Nearest Neighbor classifier, Template Matching classifier, Hidden Markov Models classifier, Adaboost algorithms classifier, Support Vectors Machines classifier and Neural Networks classifier.

In this study we use multi-layer perceptron (MLP) as a Neural Network classifier. The number of input neurons is equal to the size of related feature vectors and the number of output neurons is equal to the number of facial expressions to be recognized.

The rest of the paper is organized as follows: Section two presents the background of FACS, geometric features for emotion, principal component analysis (PCA), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Section three characterizes our new approach. Section Four describes the algorithm and the implementation. Section Five delineates the dataset and the results. Section six explains the related works. The last section concludes the paper, and presents the future work.

2 Background

2.1 Principal Component Analysis

Principle Component Analysis (PCA) is a dimension reduction technique that transforms the data-points to a new coordinate system using orthogonal linear transformation such that the transformed data-points lie with greatest variance on the first coordinate. Only the dimensions with major variations are chosen for further analysis reducing the feature space. It is based upon finding out the eigenvectors and eigenvalues [10].

For image analysis, image samples are trained to obtain the principal subspace composed of orthogonal basis vectors, then mapping the samples into the subspace to derive projection coefficient vectors as sample feature-vectors. Test images are mapped into the principal subspace to derive the corresponding feature-vectors of the test-images [25].

2.2 Facial Action Control System

Contractions of a subset of facial muscles generate human facial expressions. A set of 66 Action Units (AUs) [11] has been used to simulate the contractions of facial muscles. An action unit simulates the activities of one or several muscles in the face. Table I and II describe a relevant subset of action units needed for the simulation of the facial expressions for the six basic emotions [12].

Table I. Description of relevant facial action units

AU	FACS name	AU	FACS name
1	Inner brow raiser	12	Eye-Lid corner puller
2	Outer brow raiser	14	Dimpler
4	Brow lower	15	Lip corner depressor
5	Upper lip raiser	16	Lower lip depressor
6	Cheek raiser	17	Chin raiser
7	Lip tightener	20	Lip stretcher
9	Nose wrinkler	23	Lip tightener
10	Upper eye-lid raiser	26	Jaw drop

2.3 Associating AUs with Geometric Features

In order to use geometric features approach to find the most appropriate representation of facial images for recognition, we have to map the effect of AUs to the observable changes in the geometric features. In this section, we use the presentation of this mapping between AUs and the movement of geometric feature points provided by this research [26]. Table VI describes a mapping between AUs and the movement of geometric feature points.

Table II. Mapping of action units to geometric features

Action units ↔ Features				Action units ↔ Features			
AUs		Features		AUs		Features	
AU	Action	Id	Action	AU	Action	Id	Action
1	Up	br$_1$, bl$_1$	up	12	Pull	mr$_1$, ml$_1$	Stretch
2	Up	br$_3$, bl$_3$	up	14	Dimple	mr$_1$, ml$_1$	Stretch
4	Down	br$_1$, bl$_1$	down	16	Down	mm$_3$, mm$_4$	Down
5	Up	mm$_1$, mm$_2$	up	17	Up	mm$_3$, mm$_4$	Up
6	Up	mr$_1$, ml$_1$	stretch	20	Stretch	mr$_1$, ml$_1$	Stretch
7	Tight	mr$_1$, ml$_1$	tight	23	Tight	mr$_1$, ml$_1$	Tight
9	Wrinkl e	br$_1$, bl$_1$	down	26	Down	mr$_3$, mm$_3$, ml$_3$,	Down

The mapping shows that many muscle movements map to the same geometric features. For example, AU # 6, #12, and #14 all are involved in stretching mr1 and ml1; AU #4 and #9 pull down the inner brow point's br1 and bl1 down; and AU # 16 and #26 pull mm3 down. While, multiple emotions may map to the movement of same feature points, the magnitude of movement is different, and is derived by transformation matrix for the MLP training.

2.4 Modified Geometric Features in Face

Face-features can be modeled as a graph [14]. Face movement is a combination of all facial feature points, but some points have main role in facial expression. There are three types of nodes (feature-points): stable, passive and active. Stable feature-points are fixed, and do not make any perceptible movement. Passive feature-points do not have significant muscle movement associated with them. Active feature-points are most affected by muscle movements; the change in position of active-points results in the change of facial-expressions.

There is a modification in geometric model mentioned in [15] provided by this paper [26]. Modification is in number of feature points by reducing it from 62 points to 24 major points. These points are optimum number of points which are effective on muscle movements and facial expression. New modification improves computational efficiency. Here we use 6 points in eyebrows (br1…bl3), 8 points in eyes (er1…el4) and 10 points on mouth (mr1…ml3, mm1… mm4). The subset {er1, el1} represents stable points, the subset {er4, el4, mr2, ml2} represents passive points, and the subset {br1, br2, br3, bl1, bl2, bl3, er2, er3, el2, el3, mr1, ml1, mr3, ml3, mm1, mm2, mm3, mm4} represents the active-points.

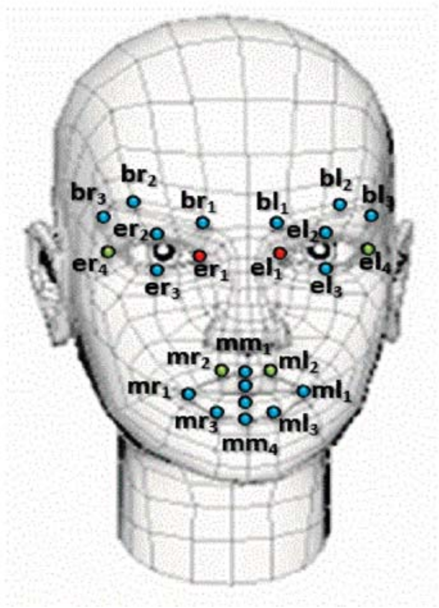


Figure 2. Feature-points in the geometric model of a face

Table III describes the deviations of various facial feature-points that are needed to simulate facial expressions. Various movements of facial feature-points are left, right, up, down, stretch and tighten.

Table III. Actions of feature-points in Figure 2.

Feature Points	Deviation
Brow points (br_1, br_2, br_3, bl_1, bl_2, bl_3)	up, down
Mid points of eyes (er_2, er_3, el_2, el_3)	up
Outer lip points (mr_1, ml_1)	stretch, tighten
Midpoint of upper lips (mm_1, mm_2)	up, down

The transformation matrix maps muscles and face movement to a formula and each feature point movement on face can be regarded as a combination of translation, rotation and scaling. This transformation is caused due to head-movements, and uses the change in coordinates of the fixed inner eye corners er1 with coordinate (x_r, y_r) and el_l with coordinate (x_l, y_l) for transformation as given in equations (1) thru (5). The abbreviations norm, Trans, rot, and sc denote normalize, transform, rotate and scale respectively.

$$\mathrm{norm}(x,y) = \mathrm{sc}(x,y) \times \mathrm{rot}(x,y) \times \mathrm{trans}(x,y) \quad (1)$$

$$\mathrm{trans}(x,y) = \begin{bmatrix} -\frac{x_l \ +x_r}{2} \\ -\frac{y_l+ \ y_r}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

Where (x_l, y_l) and (x_r, y_r) are the coordinates of left and right inner eye corners el1 and er1 respectively.

$$\mathrm{rot}(x,y) = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

Where θ is the angle between the intervals joining the inner eye corners and the horizontal x-axis.

$$\mathrm{sc}(x,y) = \frac{1}{2_{x_r}} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

$$\mathrm{sc}(x,y) = \frac{1}{2_{x_l}} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

Where x_r and x_l are the X-coordinates of right and left eyes respectively.

In order to separate the intensities of feature points for different emotions, the equation for cumulative difference is defined as follows:

$$\mathrm{diff} = \sum_{i=1}^{n-1}(E_{i+1} - E_i) - \sum_{i=1}^{n-1}(N_{i+1} - N_i) \quad (6)$$

Where E_i ($0 \leq i \leq. n - 1$) represents the feature point of an expressive face state and N_i ($0 \leq i \leq. n - 1$) represent feature point of a neutral face state respectively. The outcome diff > 0 means muscle-elongation and diff < 0 means muscle-contraction. Here about each image in database we have a vector that contains diff values and is related to a special state. Now we can train our system with these vectors.

2.5 Support Vector Machine

SVM have shown to be very effective classifiers for face recognition applications and provide the ability to generalize over imaging variants [27]. SVM provide an optimal decision hyperplane by employing kernel learning, projecting the data into a high-dimensional space [28].

To perform classification with a linear SVM, a labeled set of features {X_i, Y_i}, is constructed for all *r* features in the training data set. The class of feature C_i defined by Y_i= {1, −1}. If the data are assumed to be linearly separable, the SVM attempts to find a separating hyperplane with the largest margin. The margin is defined as the shortest distance from the separating hyperplane to the closest data point. [29]

2.6 Training Multi-Layer Perceptron

The multi-layer perceptron (MLP) is one of the most popular neural networks topologies based on back-propagation algorithm. With facial expression classes, a multi-layer perceptron is trained.

An MLP can be viewed as a logistic regression classifier where the input is first transformed using a learnt non-linear transformation. This transformation projects the input data into a space where it becomes linearly separable. This intermediate layer is referred to as a hidden layer. A single hidden layer is sufficient to make MLPs a universal approximation. However we know that there are substantial benefits to using many such hidden layers, i.e. the very premise of deep learning.

An MLP with a single hidden layer can be represented graphically as follows:

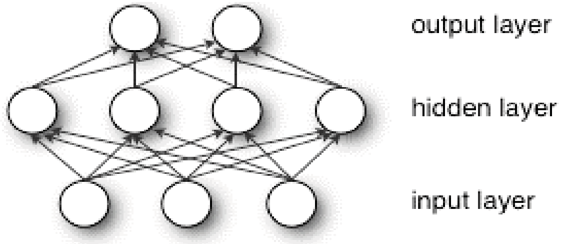


Figure 3. MLP with Single hidden layer

One of the difficulties of using the MLP neural networks is to determine the optimal number of hidden neurons before the training process. Many researchers focused their study on this part [30], [31]. Here, the number of neurons in input layer is equal to the size of related feature vector for each experiment and the number of output neurons is equal to the number of individual emotions the network is required to recognize. All input patterns are trained one by one, and then, we develop a structure of emotion detection system based on feed forward neural networks in order to detect emotions expression. The network is trained by using variants of the back propagation algorithm. The inputs vectors are applied to input units that have linear transfer functions. Others units have typically a sigmoid nonlinear function. The Back-propagation network undergoes a supervised learning process, and the output signal goes through an activation function.

In this paper, we use the artificial neural network to classify the face emotions, and in this way, a constructive training algorithm for MLP neural networks has been proposed to train data with one hidden layer by DeepLearnToolbox which is a Matlab toolbox for Deep Learning. [32]

3 Related Works

There are some issues in facial expression recognition such as accuracy and time efficiency and there are many researches in this area which try to find the better and higher accuracy beside their efficiency in time [34]. FACS system has been used for emotion generation by many researchers using AU based simulation [1, 7]. Many researchers use geometric model [13, 18, 23], and try to improve it. In some works, modified coding systems have been presented [16] and some others work to make an integration of them or to improve strategies and methods [17]. Many researchers developed the version of a computer vision system that are sensitive to subtle changes in the face [18, 19]. Some articles present a framework for recognition of facial action unit (AU) combinations by viewing the classification as a special representation problem [20, 21] and others present heuristic methods for achieving better performance [5, 20, 22, and 24]. However, to our knowledge, no one has tried to prune the search space by unifying geometric features and FACS. Our scheme significantly improves the performance while retaining the accuracy, and is suitable for real-time analysis of facial expressions and for real-time human-robot interaction.

4 Implementation

The Cohn-Kanade and the FER-13 databases have been used for the experiment. Input features sizes are high and a PCA (principle component analysis) was used to reduce image dimension. All experiments have been implemented using MATLAB. Figure 4 shows the process.

At the first step for each emotion category, 224 of the 653 images in the database were selected for training and scaling is applied on images as an image preprocessing techniques. Images of size 490*400 were transformed into 196000*1 dimensional column vectors. The remaining images were used for testing. PCA is used for dimension reduction and find component images and then we extracted facial regions from images by Viola-Jones face detector [6].

Canny filter is applied for edge extracting from PCA components. Then, we run implemented code to extract pixels around the geometric feature points from each images to find measures and differences, for MLP training by using Modified Geometric Features in Face algorithm. Finally, we used database images to train MLP with one hidden layer by DeepLearnToolbox which is a verified toolbox.

The inputs are some vectors that are related to each image in the database and extracted using AUs and the geometric feature model. The results has compared with related work which used SVM training system.

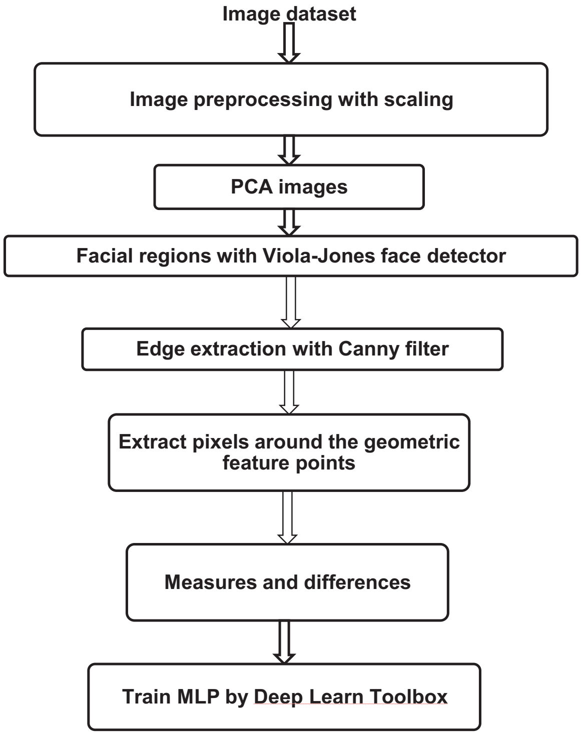


Image dataset
Image preprocessing with scaling
PCA images
Facial regions with Viola-Jones face detector
Edge extraction with Canny filter
Extract pixels around the geometric feature points
Measures and differences
Train MLP by Deep Learn Toolbox

Figure 4. Flow of the algorithm in a view

5 Experimental Results

We tested the method on Cohn-Kanade and the FER-13 data set and show the average results for both dataset here. Table IV to IX present the percentage of correctness for emotion recognition by traditional approach using SVM and MLP. The abbreviation NSur denotes "not surprise", NA denotes "not angry", NSad denotes "not sad", NH denotes "not happy", ND denotes "not disgust", and NF denotes "not fear". Table IV, V, VII and VIII show the correctness using normal and our integrated approach.

Table IV and V shows the accuracy percentage for with MLP and SVM with normal and modified action units respectively Cohn-Kanade database. It shows that in average the accuracy values are same but when we look at the table VI, Experimental results clearly demonstrate the efficiency of the proposed algorithm and shows that is more than 3 time efficient than normal method which use regular action unit system Cohn-Kanade database. The same result are shown for FER-13 database in tables VII , VIII and IX.

Table IV. Emotion recognition using MLP and SVM with normal AU method on Cohn-Kanade database.

Emotions	MLP	SVM
NSur	78%	85 %
NA	73%	83%
NSa	79%	86%
NH	75%	84%
ND	79%	89%
NF	80%	84%

Table V. Emotion recognition using MLP and SVM with modified AU method on Cohn-Kanade database.

Emotions	MLP	SVM
NSur	78%	83%
NA	79%	81%
NSa	74%	74%
NH	78%	82%
ND	75%	75%
NF	77%	78%

Table VI. Execution time efficiency on Cohn-Kanade database.

Emotions	SVM (Normal method)	MLP (normal Method)	SVM (modified method)	MLP (modified method)
NSur	7.5 ms	1.3ms	2.4 ms	.5 ms
NA	6.7 ms	1.3 ms	1.8 ms	.3 ms
NSa	7.3 ms	1.3 ms	2.3 ms	.4 ms
NH	6.7 ms	1.3 ms	1.6 ms	.2 ms
ND	7.4 ms	1.3 ms	2.2 ms	.4 ms
NF	7.5 ms	1.3 ms	2.1 ms	.3 ms

Table VII. Emotion recognition using MLP and SVM with normal AU method on FER-13 database.

Emotions	MLP	SVM
NSur	83%	87 %
NA	81%	87%
NSa	84%	88%
NH	83%	89%
ND	91%	93%
NF	92%	93%

Table VIII. Emotion recognition using MLP and SVM with modified AU method on FER-13 database.

Emotions	MLP	SVM
NSur	81%	84%
NA	82%	90%
NSa	84%	89%
NH	87%	92%
ND	88%	92%
NF	90%	93%

Table IX. Execution time efficiency on FER-13 database.

Emotions	SVM (Normal method)	MLP (normal Method)	SVM (modified method)	MLP (modified method)
NSur	4.3 ms	.7ms	.4 ms	.12 ms
NA	3.5 ms	.6 ms	.3 ms	.09 ms
NSa	4.4 ms	.7 ms	.4 ms	.1 ms
NH	3.7 ms	.7 ms	.3 ms	.08 ms
ND	4.4 ms	.7 ms	.4 ms	.11 ms
NF	4.5 ms	.7 ms	.4 ms	.12 ms

6 Conclusion and Future Work

In this paper an integrated method for facial expression detection has been presented. Our system maps the AUs for specific emotions to geometric feature point movements, and uses the characterizing feature points based upon AU mapping to prune the number of pixels being processed improving the execution time. The search space has been pruned significantly with loss of correctness. The improved performance makes it suitable for real-time robot-human interaction.

Currently, we are integrating this approach with our locality sensitive hashing technique developed by us [6]. Facial expression has not only diverse variations of facial images such as pose and illuminations but also inherent expression variation depending on subjects and emotional status. To overcome these difficulties, recent studies try to use deep learning techniques. In the future work we are going to use convolutional neural network for this purpose.

7 References

[1] M. S. Bartlett, G. Littlewort, M. G. Frank, C. Lainscsek, I. Fasel, and J. R. Movellan, “Recognizing facial expression: Machine learning and application to spontaneous behavior”, in Proc. IEEE Conf. Comput. Vis.Pattern Recognit., vol. 2. Jun. 2005, pp. 568- 573.

[2] J. M. Fellous and M.A.Arbib,”Who needs emotions? The brain meets the robots”, Oxford press, 2005.

[3] S. W. Chew, R. Rana, P. Lucey, S. Lucey, and S. Sridharan, “Sparse Temporal Representations for Facial Expression Recognition”, in Advances in Image and Video Technology Vol. 7088. New York, NY, USA: Springer-Verlag, 2012, pp.311-322.

[4] T. F. Cootes, G. J. Edwards and C. Taylor, “Active appearance models”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, Jun. 2001, pp. 681-685.

[5] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. Image Vis. Comput., 27(6): 803-816, 2009.

[6] M. Fratarcangeli,”Computational Models for Animating 3D Virtual Faces”, ISBN 978-91-7519-544-5, ISSN 0280-7971, 2013.

[7] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 10, Oct. 1999, pp. 974-989.

[8] F. Dornaika and F. Davoine, “Simultaneous facial action tracking and expression recognition in the presence of head motion”, Int. Journal Comput. Visualization., vol. 76, no. 3, 2008, pp. 257-281

[9] P. Ekman and W. Friesen, “Facial Action Coding System: A Technique for the Measurement of Facial Movement", Consulting Psychologists Press, 1978.

[10] K. Fukunaga, “Introduction to Statistical Pattern Recognition”, Academic Press, San Diego, 1990.

[11] C.J.C. Burges,"A Tutorial on Support Vector Machines for Pattern Recognition”, Bell Laboratories, 1998.

[12] L. Gang, L. Xiao-hua, Z. Ji-liu and G. Xiao-gang, “Geometric feature based facial expression recognition using multiclass support vector machines", IEEE International Conference on Granular Computing (GRC '09), 2009, pp. 318-321.

[13] K. Hong et al, “A Component Based Approach for Classifying the Seven Universal Facial Expressions of Emotion", in proc. of the IEEE Symposium on Computational Intelligence for Creativity and Affective Computing, 2013, pp. 1-8.

[14] K.E. KO and K. B. Sim, “Development of a Facial Emotion Recognition Method based on combining AAM with DBN", International Conference on Cyber worlds, 2010, pp. 87-91.

[15] I. Kotsia and I. Pitas, “Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines", IEEE Transaction on Image Processing, Vol. 16, No. 1, January 2007, pp. 172-187.

[16] J. J. Lien, T. Kanade, J. F. Cohn and C. Li, “Detection, tracking, and classification of action units in facial expression”, J. Robot. Auto. Syst., vol. 31, no. 3, 2000, pp. 131-146.

[17] M. H. Mahoor, M. Zhou, K. L. Veon, S. M. Mavadati and J. F. Cohn, “Facial action unit recognition with sparse representation”, =IEEE Int. Conf. Autom. Face Gesture Recognition, Mar. 2011, pp. 336-342.

[18] Y. Tian, T. Kanade and J. F. Cohn, “Recognizing action units for facial expression analysis”, IEEE Trans. Pattern Anal. Mach. Intell., vol.23, No. 2, Feb. 2001, pp. 97-115.

[19] M. Rogers and J. Graham, “Robust active shape model search”, in Proc. of the Eur. Conf. Comput. Vis., 2002, pp. 517-530.

[20] M. e and I. Patras, “Dynamics of facial expressions: Recognition of facial actions and their temporal segments from face profile image sequences”, IEEE

Trans. Syst., Man, Cybern. B, Cybern. vol. 36, no. 2, Apr. 2006, pp. 433-449.

[21] C. Shan, S. Gong and P. W. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study”, Image Vis. Comput., vol. 27, no. 6, 2009, pp. 803-816.

[22] Y. Tong, W. Liao and Q. Ji, “Facial action unit recognition by exploiting their dynamic and semantic relationships”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 10, Oct. 2007, pp.1683-1699.

[23] M. Rogers and J. Graham, Robust active shape model search, Proceedings of the Eur. Conf. Comput. Vis, 517-530, 2002.

[24] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression, (CVPR4HB), San Francisco, USA, 94-101, 2010.

[25] M.Ghayoumi,A.Bansal, An Integrated Approach for Efficient Analysis of Facial Expressions, SIGMAP 2014.

[26] M.Ghayoumi, A.Bansal, Unifying Geometric Features and Facial Action Units for Improved Performance of Facial Expression, CSSCC 2015.

[27] Heisele, B., Ho, P., Poggio, T., 2001. Face recognition with support vector machines: global versus componentbased approach. Proceedings of the 8th IEEE International Conference on Computer Vision, Vol. 2, pp. 688 - 694.

[28] Vapnik, V., 1995. The nature of statistical learning theory. , Springer, Berlin.

[29] H. Abrishami Moghaddam and M. Ghayoumi "Facial Image Feature Extraction Using Support Vector Machines" Proc. VISAPP 2006, Setubal, Portugal.

[30] D. Liu, C. Tsu-Shuan, and Y.Zhang, “A constructive algorithm for feedforward neural networks with incremental training,” Transactions on Circuits and Syst-Fundamental Theory and Applications, vol. 49, no. 12, pp. 1876-1879, 2002.

[31] S. Masmoudi, M. Frikha, M. Chtourou, and A. Hamida, “Efficient mlp constructive training algorithm using a neuron recruiting approach for isolated word recognition system,” International Journal of Speech Technology, vol. 14.

[32] R. Palm, Prediction as a candidate for learning deep hierarchical models of data, 2012.

[33] Mehdi Ghayoumi, A Review of Multimodal Biometric Systems Fusion Methods and Its Applications ICIS, USA, 2015.

[34] H. Abrishami Moghaddam and M. Ghayoumi "Facial Image Feature Extraction Using Support Vector Machines" Proc. VISAPP 2006, Setubal, Portugal.

SESSION

THREE DIMENSIONAL IMAGING SCIENCE AND APPLICATIONS

Chair(s)

TBA

Improving Image Memorability of Global Traveling with Intel RealSense 3D Camera

Dan Ye[1], Shih-Wei Liao[1]

[1]National Taiwan University, Department of Computer Science and Information Engineering

Abstract—Travel the world and take pictures as proof of global journeys by just sitting back at home and posing in front of Intel RealSense 3D Camera with any favorite backgrounds. The main contribution is to apply a novel method on captured images by Intel 3D camera for creating memorability maps for each image and to provide a concrete method to perform image memorability manipulation. Experiment results demonstrate it can robustly estimate the memorability of images from many different classes, positioning memorability and deep memorability features as prime candidates to improve memorability prediction of images.

Keywords: Memorability, Intel RealSense 3D camera, Background replacement.

1 Introduction

The Intel® RealSense™ SDK includes a User Segmentation module that generates a segmented image on each frame that can be used to remove or change the background behind the user's body. This paper shows that how to use this sample app's method for replacing the background with the selected static global traveling image.

This paper applies a novel framework that can predict face memorability and image memorability to evaluate the memorability of global traveling which captured by Intel® RealSense™ 3D Camera. To our best knowledge, predicting image memorability, using deep learning and LaMem, a novel diverse dataset, initiates a novel method to achieve unprecedented performance at estimating the memorability ranks of images, and evaluate memorability maps. New visual materials could be enhanced using the memorability maps approach, to reinforce forgettable aspects of an image while also maintaining memorable ones. In general, consistently identifying which images and which parts of an image are memorable or forgettable could be used as a proxy for identifying visual data useful for people, concisely representing information, and allowing people to consume information more efficiently.

The reminder of paper is organized as follows. Section II describes a comprehensive overview of current key technical solutions for background replacement, automatic memorability predictor. Section III introduces visualizing what makes an image memorable. Section IV depicts that the methodology of predicting memorability. Implementation on real experiment environment is introduced in section V. Simulation results on the previous discussed solutions are presented in section VI. Finally, conclusions are reiterated in section VII.

2 System Design

2.1 Background replacement

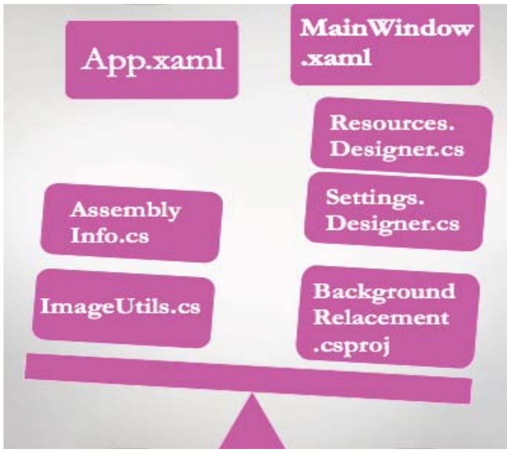


Fig.1. System Architecture

Fig.1 shows the source code design architecture in background replacement module. It implements in C#.

2.2 Feature Selection

It becomes crucial that feature selection algorithm determine a set of distinct characteristics that make an image memorable. Employing an information-theoretic approach to feature selection allows us to select a compact set of non-redundant features and calibrate features based on they contain. Selecting features that maximize mutual information with memorability, such that the total number of bits required describing an image using the selected features does not exceed B.

$$\text{F*=argmax I(F;M) s.t. } C(F) \le B \tag{1}$$

where F is a subset of the features, I(F;M) is the mutual information between F and memorability M, B is the budget (in bits), and C(F) is the total number of bits required to encode F. we assume that each feature is encoded independently,

$$C(F) = \sum_{i=1}^{n} C(f_i), f_i \in F \tag{2}$$

where $C(f_i)$ is the number of bits required to encode feature f_i, computed as $H(f_i)$, the entropy of feature f_i across the training images.

The algorithm selects features with the maximum ratio of improvement in mutual information to their cost, while the total cost of the features does not exceed the allotted budget. The set of features that provides the higher mutual information is retained.

$$I\hat{G}(f) = \min_j (I(f_j \cup f_j; M) - I(f_j; M)), f_j \in F \tag{3}$$

The ratio of this approximation to the cost of the feature is used as the score to evaluate the usefulness of features during greedy selection. This ensures that feature selected at each iteration maximizes the per-bit minimal gain in mutual information over each of individual features selected. To maximize the mutual information beyond the greedy algorithm, employing multiple passes on the feature set. Given a budget B, we first greedily add features using a budget of 2B, and then remove features （reduce the mutual information）until we fall within the allotted budget B. This allows for features that were added early on in the forward pass. These forward and backward passes are repeated 4 times each. At each pass, objective function cannot decrease, and final solution is still guaranteed to have a total cost within the allotted budget B. Feature selection within realm of a predictive model allows us to better capture features that achieve a concrete and practical measure of performance: “which set of features allows us to make the best predictions about an image’s memorability? While selecting such features would be computationally expensive to do over all our 923 features, using a pruned set of features obtained via information-theoretic selection makes this feasible. We employ a support vector regress (SVR) as our predictive model. The submodularity ratio of a function is the best predictor of how well a greedy algorithm performs. Regression performance has a high submodularity ratio.

2.3 Automatic memorability predictor

This section applies a probabilistic framework that models how and which local regions from an image may be forgotten using a data-driven approach that combines local and global images features. The model automatically discovers memorability maps of individual images without any human annotation. We incorporate multiple image region attributes in this algorithm, leading to improved memorability prediction of images.

Made predictions on the basis of a suite of global image features pixel histograms, GIST, SIFT, HOG, SSIM. Running the same methods on current 2/3 data splits achieves $\rho = 0.468$. Do better by using our selected features as an abstraction layer between raw images and memorability. We trained a suite of SVRs to predict annotations from images, and another SVR to predict memorability from these predicted annotations. For annotation types, we used the feature types selected by our 100-bit predictive selection on 2/3 training sets. To predict the annotations for each image in our training set, we spilt the training set in half and predicted annotations for one half by training on the other half, and vice versa, covering both halves with predictions. We then trained a final SVR to predict memorability on the test set in three ways: 1) using only image features (Direct), 2) using only predicted annotations (Indirect), and 3) using both (Direct + Indirect). Combining indirect predictions with direct predictions performed best $\rho = 0.479$, slightly outperforming the direct prediction method $\rho = 0.468$.

3 Visualizing the memorable image

Since object content appears to be important in determining whether or not an image will be remembered, visualizing object-based "memory maps" for each image. These maps shade each object according to how much the object adds to, or subtracts from, the image's predicted memorability. More precisely, to quantify the contribution of object i to an image, employ a prediction function, f, that maps object features to memorability scores and calculate how its prediction m changes when zero features associated with object i from the current image's feature vector, $(a_1,...,a_n)$. This gives a score s_i for each object in a given image:

$$m_1 = f(a_1,...,a_i,...,a_n) \tag{4}$$

$$m_2 = f(a_1,...,0,...,a_n) \tag{5}$$

$$s_i = m_1 - m_2 \tag{6}$$

For the prediction function f, the current method uses S VR on Labeled Multiscale Object Areas, trained as abo ve. Thus, these maps show predictions as to what will make a novel image either remembered or not remembe red. The validity of these maps is supported by SVR th at can be used to generate Labeled Multiscale Object A reas regression makes predictions that correlate relatively well with measured memory scores$\rho = 0.48$.

This visualization gives a sense of how objects contribute to the memorability of particular images. Estimated an object's overall contribution as its contribution per image, calculated as above, averaged across all test set images in which it appears with substantial size (covers over 4000 pixels). This method sorts objects into an intuitive ordering: people, interiors, foregrounds, and human-scale objects tend to contribute positively to memorability; exteriors, wide angle vistas, backgrounds, and natural scenes tend to contribute negatively to memorability.

4 Predicting Memorability of face

As we make changes to a face, this section introduces a model to reliably predict its memorability; if we cannot predict memorability, then we cannot hope to modify it in a predictable way. Thus, in this section, we explore various features for predicting face memorability and apply a robust memorability metric to significantly improve face memorability prediction. We also note that the task of automatically predicting the memorability of faces using computer vision features.

Faces tend to have a significantly higher false alarm rate than scenes. Rather than being memorable (with high correct detections), these faces are in fact "familiar" - people are more likely to report having seen them, leading to both correct detections and false alarms. To account for this effect, this paper applies a slight modification to the method of computing the memorability score.

For predicting memorability, dense global features such as HOG and SIFT significantly outperform landmark-based features such as 'shape' by about 0.15 rank correlation. This implies that it is essential to use these features in our face modification algorithm to robustly predict memorability after making modifications to a face. While powerful for prediction, the dense global features tend to be computationally expensive to extract, as compared to shape. Shape is used in this algorithm to parameterize faces so it essentially has zero cost of extraction for modified faces.

5 Implement

Development environment:
Windows 10 on MacBook Air
Microsoft Visual studio professional 2015
Inter realsense camera F200
Intel realsense SDK R5
SDK Runtime Distributable
F200 Camera Driver:Depth Camer Manger
Utility PRO 4.6.3

5.1 Intel RealSense F200

The Intel® RealSense™ 3D Camera (F200) houses both a 640x480 resolution IR camera and a 1080p RGB color camera. Combining data from both the IR depth and color cameras has the advantage of overcoming many of the issues associated with 2D camera background subtraction, such as a dynamic backgrounds (e.g., people or objects moving in the background), shadows, and varying lighting conditions.

5.2 Global traveling

The selected background image is automatically scaled by the app to the pixel dimensions of the color and depth streams (640x480). On each frame the background and segmented images are stitched together and rendered in a WPF Image control.

5.3 Improving memorablity

Pre-trained CNN using Caffe deep learning toolbox, it provides the network deploy file, the trained network model, and the train-val file which can be loaded using Caffe. Extract CaffeNet / AlexNet features using the Caffe utility. MemNet trained on first train/test split of LaMem, with FA, rank correlation with 10 crops per image 0.64.

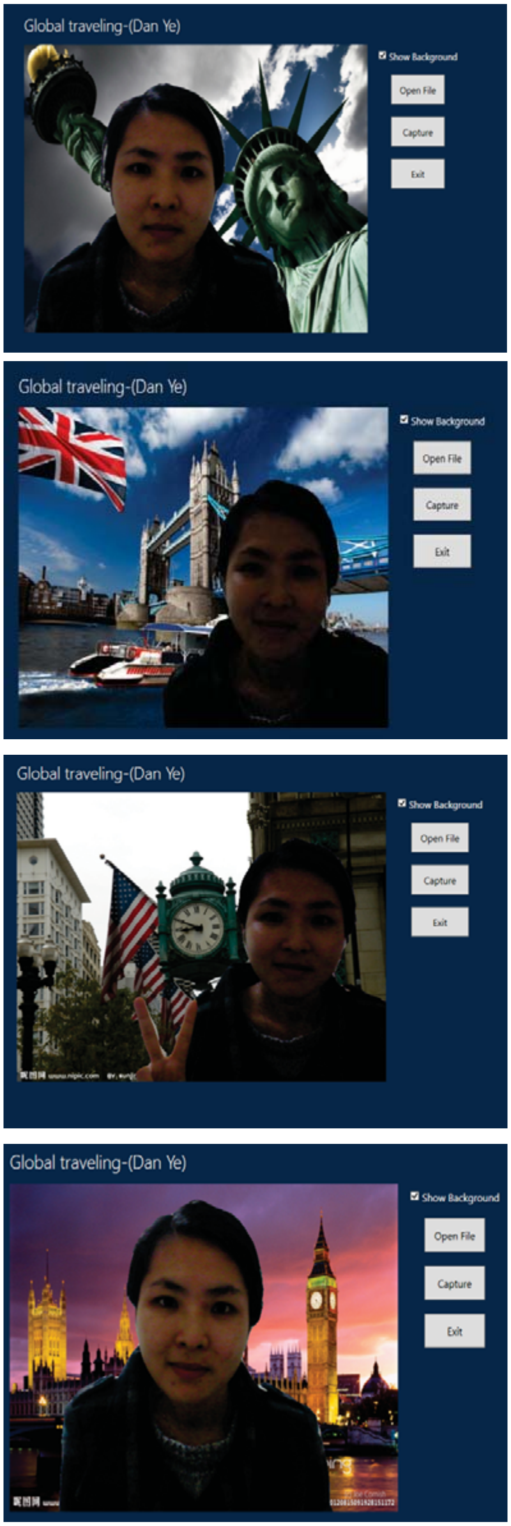
Firstly, conduct memorability experiments using Amazon’s Mechanical Turk (AMT) on the 60, 000 target images obtained by sampling the various datasets. Each task lasted about 4.5 minutes consisting of a total of 186 images divided into 66 targets, 30 fillers, and 12 vigilance repeats. It can be obtained 80 scores per image on average, resulting in a total of about 5 million data points. The evaluation is to use rank correlation to measure consistency.

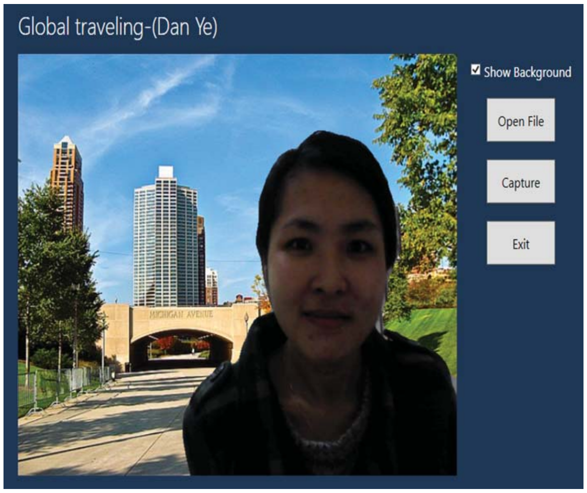
Then randomly select 500 images from their dataset, and collected 80 scores per image. After applying this algorithm to correct the memorability scores, it can obtain a within-dataset human rank correlation of 0.77 (averaged over 25 random splits). Furthermore, it can obtain a rank correlation of 0.76 when comparing the independently obtained scores from the two methods. This shows that this method is well suited for collecting memorability scores.

To account for the wrong correct detections, simply subtract the false alarms from the hit count to get an estimate of the true hit count of an image. To show that this metric is robust, we apply it to both the face and scene memorability datasets. We observe that human consistency remains largely the same in both cases. This is expected as the false alarms tend to be consistent across participants in the study. Importantly, we observe that there is a significant increase in the prediction performance from rank correlation of 0.33 to 0.51 for face memorability. By using this new metric, we have effectively decreased noise in the prediction labels (memorability scores) caused by inflated memorability scores of familiar images. This allows the learning algorithm to better model the statistics of the data that best describe memorability.

6 Results

Fig.2. Global background replacement





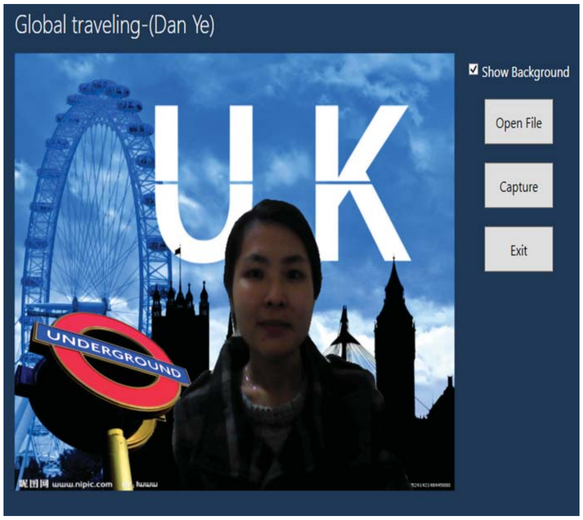
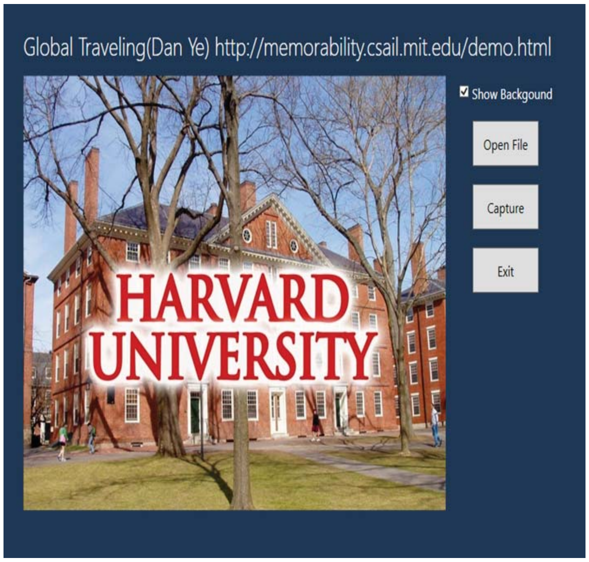
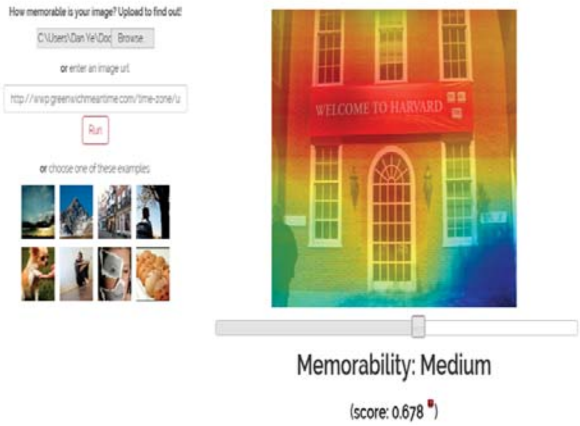
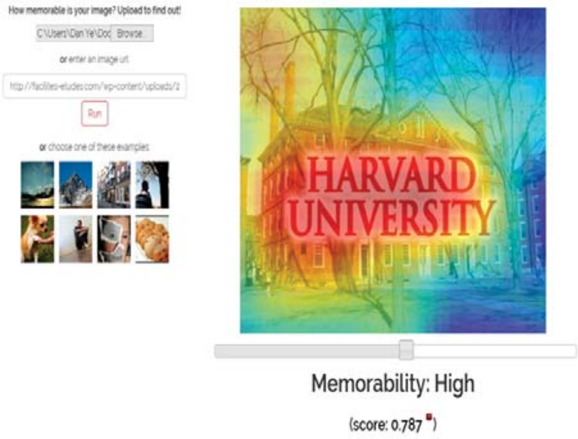


Fig.3. Predict memorability


Global Traveling(Dan Ye) http://memorability.csail.mit.edu/demo.html
☑ Show Backgound
Open File
Capture
Exit
HARVARD UNIVERSITY


How memorable is your image? Upload to find out!
C:\Users\Dan Ye\Doc Browse
or enter an image url
http://www.greenwichmeantime.com/time-zone/u
Run
or choose one of these examples
WELCOME TO HARVARD
Memorability: Medium
(score: 0.678)


How memorable is your image? Upload to find out!
C:\Users\Dan Ye\Doc Browse
or enter an image url
http://facilities-etudes.com/wp-content/uploads/2
Run
or choose one of these examples
HARVARD UNIVERSITY
Memorability: High
(score: 0.787)

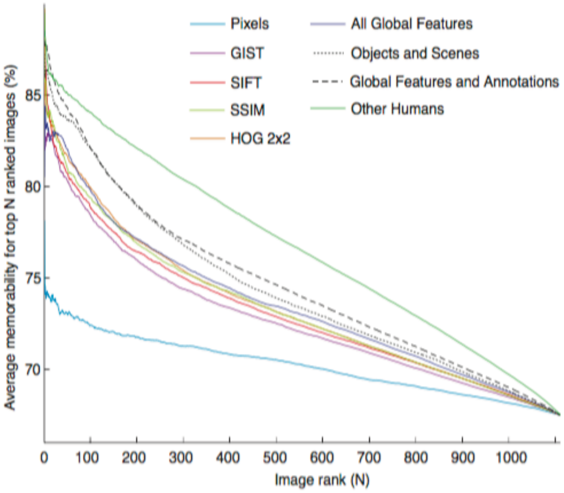


Fig.4 Regressions results

Fig.2 shows global background replacement results from New York to London. Fig.3 depicts the GUI of global traveling and different levels of predicting memorability results. Fig.4 elaborates comparison of regressions results averaged across 25 random split half trials. Images are ranked by predicted memorability and plotted against the cumulative average of empirically measured memorability scores. Error bars omitted for clarity. Fig.5 plots showing the relationship of memorability and various image attributes. For each curve, the images are sorted independently using ground-truth memorability scores. As each curve may contain a different number of images in emotion, the image index above has been normalized to be from 0 to 1. Memorability scores of global traveling images for the three settings are shown in Fig. 6. Note that the scores for *low*, *medium* and *high* are independently sorted.

7 Conclusions

This paper applies background replacement module and touchless controller module, as well as the memorability predictor in the images captured by Intel RealSense 3D Camera. It introduces the current state-of-art technical solutions for face and image memorability. The key contribution is to apply a novel framework in predicting memorability of face photographs and modifying memorability. Moreover, this paper applies computer vision techniques to extract memorability automatically. It defined an image's memorability score as the probability that a viewer will detect a repeat of the image within a stream of pictures. Results show that there is a large degree of consistency among different viewers, and that some images are more memorable than others even when there are no familiar elements (such as relatives or famous monuments). This work is a further attempt to quantify this useful quality of individual images.

8 References

[1].Aditya Khosla, Jianxiong Xiao, Phillip Isola, Antonio Torralba, Aude Oliva. Image Memorability and Visual Inception, In SIGGRAPH Asia, 2012.

[2].A. Khosla, A. S. Raju, A. Torralba and A. Oliva, Understanding and Predicting Image Memorability at a Large Scale, International Conference on Computer Vision (ICCV), 2015.

[3].Aditya Khosla, Wilma A. Bainbridge, Antonio Torralba and Aude Oliva, Modifying the Memorability of Face Photographs, International Conference on Computer Vision (ICCV), 2013.

[4].Wilma A. Bainbridge, Phillip Isola, Aude Oliva. The Intrinsic Memorability of Face Photographs, In Journal of Experimental Psychology: General (JEPG), 2013.

[5].Aditya Khosla, Jianxiong Xiao, Antonio Torralba, Aude Oliva. Memorability of Image Regions, In Neural Information Processing Systems (NIPS), 2012.

[6].Phillip Isola, Devi Parikh, Antonio Torralba, Aude Oliva. Understanding the Intrinsic Memorability of Images, In Advances in Neural Information Processing Systems (NIPS), 2011.

[7].Phillip Isola, Jianxiong Xiao, Antonio Torralba, Aude Oliva. What makes an image memorable?, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[8].R. Dubey, J. Peterson, A. Khosla, M.-H. Yang, and B. Ghanem. What makes an object memorable? In International Conference on Computer Vision (ICCV), 2015.

[9].B. Celikkale, A. T. Erdem, and E. Erdem. Visual attention-driven spatial pooling for image memorability. In CVPR Workshop. IEEE, 2013.

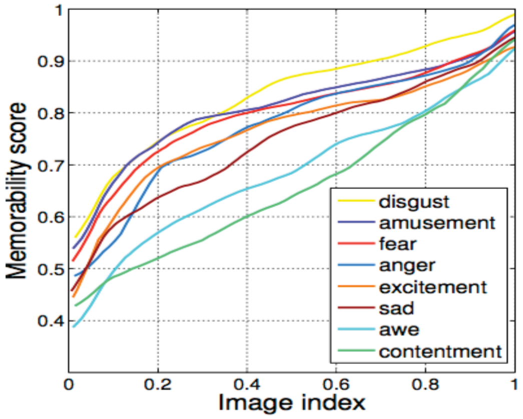


Fig.5 Memorability vs emotions

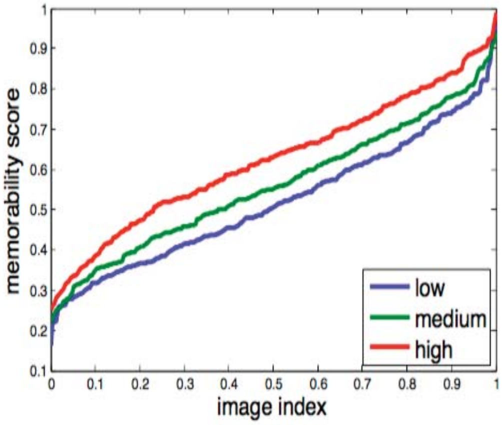


Fig.6 Memorability scores

Error reduction of the absolute phase recovered from three sets of fringe patterns with selected wavelengths

Jiale Long [1], Jiangtao Xi[2*], Jianmin Zhang[1], Yi Ding[3]
[1]School of Information Engineering, Wuyi University, Jiangmen, Guangdong, China
[2] School of Electrical Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW, Australia
[3]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei, China
*Corresponding author Jiangtao Xi: jiangtao@uow.edu.au

Abstract *-In a recent published work we proposed a technique to recover the absolute phase maps of three sets of fringe patterns with selected wavelengths. It is demonstrated that the absolute phase maps can be unwrapped from the wrapped phase maps by the three-step phase-shifting profilometry (PSP), using only 9 fringe patterns. However, few incorrect values still remain in the recovered absolute phase. In this paper, we develop a method to correct and remove the errors in the recovered absolute phase. The experiments validate the effectiveness of our proposed method.*

Keywords: Phase Shift; Phase Unwrapping; Three-dimension Measurement; Error Reduction

1 Introduction

Fringe projection profilometry (FPP) is one of the most promising technologies for non-contact 3D shape measurement. In these systems, fringe patterns are measured by digital devices such as CCD cameras, and then wrapped phase is retrieved using phase shift techniques. As the retrieved phase is falling in $(-\pi,\pi)$, phase unwrapping should be implemented to unfold the principal value to the unbounded phase [1-3]. Over the years, many phase unwrapping algorithms have been developed [4-14]. In a recent published work we proposed a technique to recover the absolute phase maps of three sets of fringe patterns with flexible selection of fringe wavelengths [14]. It is demonstrated that the absolute phase maps can be unwrapped from the wrapped phase maps by the three-step phase-shifting profilometry (PSP), using only 9 fringe patterns. Ideally, according to the theoretical analysis, the recovered absolute phase should be characterized by monotonically increasing. However, we still find few incorrect unwrapped values in the experimental results in [14]. Therefore, further work is needed to develop approaches to detect and eliminate them. In this paper, we demonstrate that the number of pixels with error values is so small that these errors could be corrected and discarded. In this method we take two steps to reduce the errors. The first step is error correcting. Since the absolute phase varies monotonically, the algorithm of linear interpolation is a good choice to replace the error pixels. The second step is using a designed quality template to discard the remaining error pixels from further three-dimension reconstruction when the first step fails in error correction sometimes.

The paper is organized as follows. In Section 2 we give a brief review of the technique in [14]. Section 3 presents the method of error reduction, and experiments and results are presented to validate the proposed method. Section 4 concludes the paper.

2 The technique proposed in [14]

With the approach proposed in [14], three sinusoidal fringe patterns are projected onto the surface of an object, which are reflected and captured by a camera. The three fringe patterns are with different spatial wavelengths, denoted by λ_1 , λ_2 and λ_3 respectively, whose intensity varies in a sinusoidal manner vertically (i.e., in the y direction). It should be noted that wavelength as defined here represents the total number of pixels per fringe period. After analyzing the one-to-one correlation between the image captured by the camera and the projected image, we have six useful inequalities:

$$-\lambda_2 < \frac{\lambda_1\varphi_1(y)-\lambda_2\varphi_2(y)}{2\pi} < \lambda_1,$$

$$\text{i.e., } -\lambda_2 < \left[m_2(y)\lambda_2 - m_1(y)\lambda_1\right] < \lambda_1 \qquad (1)$$

$$-\lambda_3 < \frac{\lambda_1\varphi_1(y)-\lambda_3\varphi_3(y)}{2\pi} < \lambda_1,$$

$$\text{i.e., } -\lambda_3 < \left[m_3(y)\lambda_3 - m_1(y)\lambda_1\right] < \lambda_1 \qquad (2)$$

$$-\lambda_3 < \frac{\lambda_2\varphi_2(y)-\lambda_3\varphi_3(y)}{2\pi} < \lambda_2 ,$$

$$\text{i.e., } -\lambda_3 < \left[m_3(y)\lambda_3 - m_2(y)\lambda_2\right] < \lambda_2 \qquad (3)$$

$$0 \le m_1(y) < R/\lambda_1 \qquad (4)$$

$$0 \le m_2(y) < R/\lambda_2 \qquad (5)$$

$$0 \le m_3(y) < R/\lambda_3 \qquad (6)$$

With inequalities (1)~(6) above, an unique mapping from $\left[\lambda_1\varphi_1(y)-\lambda_2\varphi_2(y)\right]/2\pi$, $\left[\lambda_1\varphi_1(y)-\lambda_3\varphi_3(y)\right]/2\pi$, $\left[\lambda_2\varphi_2(y)-\lambda_3\varphi_3(y)\right]/2\pi$ to $m_1(y)$, $m_2(y)$ and $m_3(y)$ can be identified, where $\varphi_1(y)$, $\varphi_2(y)$ and $\varphi_3(y)$ are wrapped phases which can be obtained by phase-shifting profilometry (PSP) from the deformed fringe patterns captured by the camera, and $m_1(y)$, $m_2(y)$, $m_3(y)$ are the fringe orders. A look-up table could be constructed by the unique mapping which is used to recover absolute phase map. However, to unwrap the absolute phase, the wavelengths of the three fringe patterns should satisfy the constraint expressed as follows:

$$R \le \lambda_1\lambda_2\lambda_3 / \left(k^2 k_1 k_2 k_3\right) \qquad (7)$$

where R is the resolution of projector, k is the greatest common measure (g.c.m.) of λ_1 , λ_2 and λ_3 . When $\lambda_1 = kg_1$, $\lambda_2 = kg_2$, $\lambda_3 = kg_3$, k_1 is the g.c.m. of g_1 and g_2 , k_2 is the g.c.m. of g_2 and g_3 , and k_3 is the g.c.m. of g_1 and g_3 . The allowable phase error of this method is:

$$0 \le \Delta\varphi_{\max} < \text{median}\left(\frac{kk_1\pi}{\lambda_1+\lambda_2}, \frac{kk_2\pi}{\lambda_1+\lambda_3}, \frac{kk_3\pi}{\lambda_2+\lambda_3}\right) \qquad (8)$$

Let us consider an example where $R = 768$ and $(\lambda_1,\lambda_2,\lambda_3) = (80,64,48)$. Since $768 \le 80\times64\times48/16^2$, the selection of wavelengths meets the requirement (7). From Eq. (8) we can get the upper bound of the allowable phase error is $\pi/8$. Since the maximal phase error on the wrapped phase maps obtained from three-step PSP is $\pi/10$, the absolute phase maps could be successfully recovered.

The look-up table which demonstrates the unique mapping is constructed in Table 1. Table 1 covers all the possible values of $m_1(y)$, $m_2(y)$ and $m_3(y)$, and the last three columns meet the requirements of the desired range (1)~(3) without repetition.

Table 1. Mapping from $m_2(y)\lambda_2 - m_1(y)\lambda_1$, $m_3(y)\lambda_3 - m_1(y)\lambda_1$ and $m_3(y)\lambda_3 - m_2(y)\lambda_2$ to $m_1(y)$, $m_2(y)$, $m_3(y)$ when $\lambda_1 = 80$, $\lambda_2 = 64$, $\lambda_3 = 48$ and $R = 768$

$m_1(y)$	$m_2(y)$	$m_3(y)$	$m_2(y)\lambda_2 - m_1(y)\lambda_1$	$m_3(y)\lambda_3 - m_1(y)\lambda_1$	$m_3(y)\lambda_3 - m_2(y)\lambda_2$
3	3	5	-48	0	48
7	8	11	-48	-32	16
2	2	3	-32	-16	16
6	7	10	-32	0	32
1	1	1	-16	-32	-16
1	1	2	-16	16	32
5	6	8	-16	-16	0
5	6	9	-16	32	48
9	11	15	-16	0	16
0	0	0	0	0	0
0	0	1	0	48	48
4	5	6	0	-32	-32
4	5	7	0	16	16
8	10	13	0	-16	-16
8	10	14	0	32	32
3	4	5	16	0	-16
3	4	6	16	48	32
7	9	12	16	16	0
7	9	13	16	64	48
2	3	4	32	32	0
6	8	10	32	0	-32
6	8	11	32	48	16
1	2	2	48	16	-32
1	2	3	48	64	16
5	7	9	48	32	-16
9	12	16	48	48	0
0	1	1	64	48	-16
4	6	8	64	64	0
8	11	14	64	32	-32

In experiment, the resolution of the projector is 1024× 768. The three fringe patterns with spatial wavelengths $(\lambda_1,\lambda_2,\lambda_3) = (80,64,48)$ are projected onto a toy model, as shown in Fig.1(a), 1(b) and 1(c). Since the width of the toy model is about half of the width of projection area, only about half number of the fringes can be seen on the toy. The wrapped phase maps obtained from three-step PSP are shown in Fig.1(d), 1(e) and 1(f).

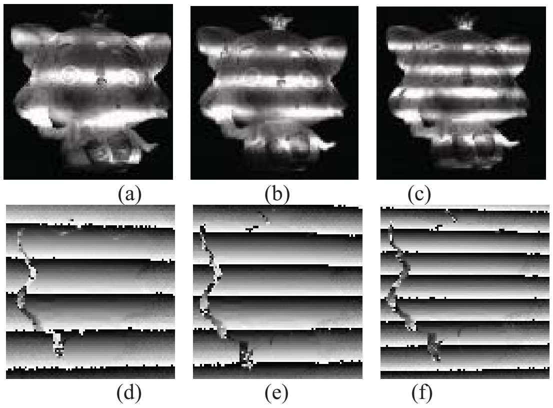


Fig.1 Experiment results when $(\lambda_1,\lambda_2,\lambda_3) = (80,64,48)$. (a), (b) and (c) are the deformed fringe patterns; (d), (e) and (f) are the wrapped phase maps get by three-step PSP.

In order to make the results more clearly, a shadow noise filter [15,16] is employed, so the shadow-noised regions are discarded from further processing. Fig. 2(a) showed the filter.

Fig. 2(b) is the unwrapped phase map of $\lambda = 80$. According to the method in [14], the unwrapped phase map should be characterized by monotonic variance over the areas of smooth shape change on the model, but unfortunately, there are still few incorrect unwrapped values shown in Fig.2 (b).

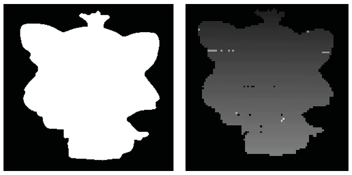

(a) (b)

Fig.2(a) is the designed shadow noise filter; (b) is the recovered absolute phase map.

Fig.3 give the sections of recovered absolute phase map showed in Fig.2 (b) on $x = 500, 667$ respectively, where x is the horizontal pixel index. Apparently, the incorrectly unwrapped pixels are clearly observed by the discontinuities and sharp changes.

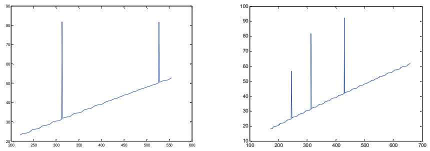

(a) (b)

Fig.3 Recovered absolute phase map on sections. (a) is on the section $x = 500$; (b) is on the section $x = 667$.

3 Error reductions

Temporal phase unwrapping approaches use multiple fringe patterns, which recover the absolute phase on pixel-by-pixel basis. Since the correct recovered absolute phase is monotonic along the direction perpendicular to the fringe patterns, this property could be used to detect the incorrectly unwrapped points pixel-by-pixel. Thus the monotonicity of absolute phases can be expressed as:

$$\Phi(x, y) < \Phi(x, y+1) \qquad (9)$$

where $\Phi(x, y)$ is the absolute phase. Therefore, any phase value which does not meet this requirement should be regarded as an incorrect unwrapped point. Therefore we have two steps to handling these errors. The first step is replacing the error points by the algorithm of data interpolation, but if there are some successive error points, the algorithm of data interpolation may fail in error correction. Hence, in the second step, a quality template is used to discard the remaining error pixels which have not been corrected in the first step.

The experimental results in section 2 are used to demonstrate the effectiveness of our proposed error correction method. For many application scenarios, linear or quadratic function is sufficient to approximate the local changes of the absolute phase. Based on this observation, we test several interpolation strategies and find the linear interpolation is the best one. Fig.2(b) shows the absolute phase before correction and Fig.3 shows the sections of Fig.2(b) containing discontinuities and sharp changes (on the section $x = 500$ and 667). Fig.4(a) shows the absolute phase after correction by the linear interpolation and Fig.4(b) and 4(c) show the sections of Fig.4(a) on $x = 500$ and $x = 667$ which are characterized by monotonic variance, hence the revised absolute phase is considered as correct.

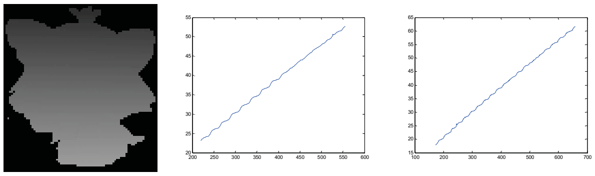
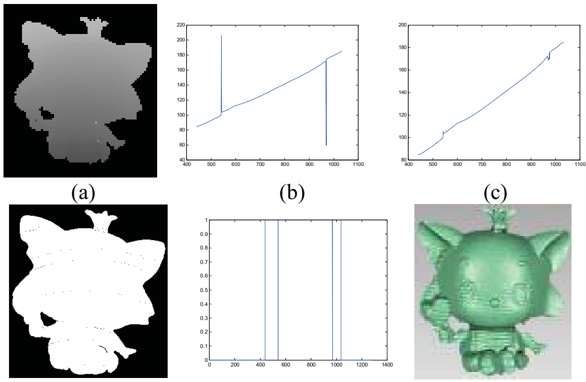

(a) (b) (c)

Fig.4 (a) is the absolute phase map after incorrect points replacing; (b) is the section of (a) on $x = 500$; (c) is the section of (a) on $x = 667$.

The experiment result in [14] where $(\lambda_1, \lambda_2, \lambda_3) = (25, 30, 35)$ is used to demonstrate the error discarding. Fig.5(a) shows the absolute phase before error reduction, Fig.5(b) shows the section on $x = 420$ of Fig.5(a). Linear interpolation is employed at the first step, and Fig.5(c) shows the section on $x = 420$ of the absolute phase after correction which still leaves some jumps. Hence, the second step is needed. The quality template shown in Fig.5(d) is designed according to the Eq.(9), the template selects the points that their absolute phase values do not increase monotonically as the position with errors. Fig.5(e) shows the quality template on section $x = 420$ finds the errors correctly. Note that the outermost two lines of Fig.5(d) are the edges of shadow noise filter, not the position of errors.


(a) (b) (c)

(d) (e) (f)

Fig.5 (a) is the absolute phase map before incorrect points discarded; (b) is the section on $x = 420$ of (a); (c) is the section on $x = 420$ of absolute phase map after incorrect points replacing; (d) is the designed quality template of (a); (e) is the section on $x = 420$ of (d); (f) is the reconstructed 3-D view after incorrect points discarded.

In this experiment, the resolution of the camera is 1280×1024, the amount of pixels should be processed is only part of them due to the use of shadow noise filter which has discarded the shadow-noised regions. The number of pixels which we unwrapped is 226136, the number of incorrect unwrapped pixels is 736, and the error rate is about 0.0033, i.e., 0.33%. The small number of the incorrectly unwrapped pixels could be corrected and discarded while rarely effect on the three-dimension reconstruction. Fig.5(f) is the reconstructed 3-D view after incorrect points discarded. As the point cloud data retrieved by absolute phase is so huge that discarding few pixels would not degrade the quality of three-dimension reconstruction of reserved points, which can be observed from Fig.5(f).

Although the pixels with wrong absolute phase values are few, the wrong absolute phase values would lead to obvious errors in three-dimensional reconstruction results. This method could eliminate and correct these wrong absolute phase values, which will improve the quality of reconstruction results significantly.

4 Conclusions

In this paper, we develop a method to correct and discard the errors in the recovered absolute phase from [14]. Since the correct absolute phase is monotonic along the direction perpendicular to the fringe patterns, the method of linear interpolation is used to correct the errors at the first step. A quality template is designed to discard the remaining errors when the first step fails. The measurement errors have been significantly reduced. The method proposed in this paper could be used to correct the errors from other temporal phase unwrapping techniques.

5 Acknowledgements

This work was partially supported by the Science Foundation for Young Teachers of Wuyi University under Grant number 2013zk06; Foundation for Distinguished Young Talents in Higher Education of Guangdong, China under Grant number 2015KQNCX167.

6 References

[1] S.Zhang, "Digital multiple wavelength phase shifting algorithm," Proc. SPIE 7432, 74320N-1(2009).

[2] Towers, D. P., Jones, J. D. C., and Towers, C. E., “Optimum frequency selection in multi-frequency interferometry,” Opt. Let. 28, 1–3(2003).

[3] Towers, C. E., Towers, D. P., and Jones, J. D. C., “Absolute fringe order calculation using optimised multi-frequency selection in full-field profilometry,” Opt. Laser Eng. 43, 788–800(2005).

[4] S. Zhang, P. S. Huang, "High-resolution, real-time three-dimensional shape measurement," Opt. Eng. 45:123601-123608 (2006).

[5] S. Zhang, X. Li, and S. T. Yau, "Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction," Appl. Opt. 46(1), 50-57(2007).

[6] H. J. Chen, J. Zhang, D. J. Lv, and J. Fang, “3-D shape measurement by composite pattern projection and hybrid processing,” Opt. Express. 15, 12318–12330 (2007).

[7] J. Li, L.G, Hassebrook, and C.Guan, "Optimized two-frequency phase-measuring profilometry light-sensor temporal-noise sensitivity," J. Opt. Soc. Am. A 20, 106–115(2003).

[8] K. Liu, Y. Wang, D. L. Lau, Q. Hao, and L. G. Hassebrook, “Dual-frequency pattern scheme for high-speed 3-D shape measurement,” Opt. Express 18, 5229–5244 (2010).

[9] C. E. Towers, D. P. Towers, and J. D. C. Jones, “Absolute fringe order calculation using optimisied multifrequency selection in full-filed profilometry,” Opt. Lasers Eng. 43, 788–800 (2005).

[10] Y. Ding, J. Xi, Y. Yu, and J. Chicharo, "Recovering the absolute phase maps of two fringe patterns with selected frequencies," Opt. Let. 36, 2518-2520(2011).

[11] Y. Ding, J. Xi, Y. Yu, W. Q. Cheng, S. Wang, and J. Chicharo, "Frequency selection in absolute phase maps recovery with two frequency projection fringes," Opt. Express 20, 13238-13251(2012).

[12] Satoshi Tomioka, Shusuke Nishiyama,”Phase unwrapping for noisy phase map using localized compensator,” Appl. Opt, 51 (21), 4984-4994(2012).

[13] J. Long, J. Xi , M. Zhu, W. Cheng, R. Cheng, Z. Li and Y. Shi, “Absolute phase map recovery of two fringe patterns with flexible selection of fringe wavelengths,” Appl. Opt. 53 (9),1794-1801(2014)

[14] J. Long, J. Xi, J. Zhang, M. Zhu, W. Cheng, Z. Li and Y. Shi, “Recovery of absolute phases for the fringe patterns of three selected wavelengths with improved anti-error capability,” Journal of Modern Optics, in print

[15] S. Zhang, “Phase unwrapping error reduction framework for a multiple-wavelength phase-shifting algorithm,” Opt. Eng. 48(10), 105601 (2009).

[16] H. J. Tiziani, “Heterodyne Interferometry using two wavelengths for dimensional measurements,’’ Proc. SPIE, 1553, 490-501(1991).

A TIN-based classification approach for buildings or vegetation extraction

Shijun Tang
School of Engineering, Computer Science and Mathematics
West Texas A&M University
Canyon, TX, USA 79016
stang@wtamu.edu

Rajan Alex
School of Engineering, Computer Science and Mathematics
West Texas A&M University
Canyon, TX, USA 79016
ralex@wtamu.edu

Abstract—**In this paper, we propose a new TIN-based classification approach for feature extraction from Light Detection and Ranging (LiDAR) data point clouds. The method builds TIN first and then computes variance (standard deviation) of normal vectors at each node to classify objects from raw LiDAR data point clouds. Experimental results indicate that our method can effectively classify buildings or vegetation via choosing different threshold values of variance (or standard deviation) of normal vectors.**

Keywords— LiDAR, variance of normal vector, triangulated irregular network, classification

I. INTRODUCTION

Classification and detection of vegetation or buildings are important in land use planning as well as environment monitoring. Feature extraction from LiDAR point clouds has been playing the key role in classification and detection of vegetation or buildings. It is convenient that the Triangulated Irregular Network (TIN) model has been employed to describe and operate the 3D sub-randomly spatial distributed LiDAR points and neighboring relation of spatial discrete points. A TIN model has been constructed via the planar Delaunay triangulation network. The Triangulated Irregular Network model (TIN) can directly reflect and consider the points' neighboring relation in 3D LiDAR point clouds.

For filtering and classifying the LiDAR data, many methods based on the TIN model have been proposed. The main approaches include the region growing [1], and the least square interpolation method [2]. Akel *et al* (2003) [3] provided that, for each triangle in the TIN model, the normal direction and mean height of the three points that compose it are calculated. Then, for each two triangles, if they have similar normal directions and heights, these triangles are merged into a region using the region growing approach. Although this method is for dense raw LiDAR data, there exists many misclassifications because they determine if two neighboring triangles are similar only by computing their normal direction and the height of the neighboring triangle.

Zeng *et al* [4] designed an assistant plane and made classifications according to the neighbor's number and height difference of every point based on a TIN model. But, the height difference of every point on TIN doesn't completely reflect the characteristics of objects (vegetation and buildings) from the physical nature.

Belkhouche *et al* [5] proposed a surface flatness method based on a TIN model for detection of vegetation from raw LiDAR data. The flatness of a 3D object modeled by a set of points is defined as its volumetric surface divided by its surface projected on 2D. Vegetation tends to have higher flatness values than other objects. However, some objects (buildings with no flat tops) also have higher flatness values.

The above methods have their own advantages and disadvantages. In order to make classification better, we combine the variance of normal vectors distribution with height interval [7, 8, 9] and propose a novel method to perform classification of aerial LiDAR data into buildings or vegetation based on a TIN model, which effectively uses the information of LiDAR data cloud points.

II. METHODOLOGY

A. Triangulated Irregular Network (TIN)

Triangulated Irregular Network (TIN) comprises a triangular network of vertices with associated coordinates in three dimensions connected by edges. Three-dimensional visualizations are readily created by rendering of the triangular facets. The TIN model has typically been constructed based on the Delaunay triangulation network. The TIN is widely used for representation of the physical land surface or sea bottom, made up of irregularly distributed nodes and lines with three dimensional coordinates (x, y, and z). The TIN model has often been employed to operate the 3D spatial distributed LiDAR points.

B. Digital Terrain Model

Since LiDAR data is collected from measuring the time delay between transmission of a pulse and detection of the reflected signal, the elevation and surface information of objects might be extracted from LiDAR data. Digital

Elevation Model (DEM) is created using elevation of bare earth points, and Digital Surface Model (DSM) is based on the actual surface, including vegetation and buildings. A normalized digital surface model (nDSM) represents the height of ground features. The nDSM is obtained from the aerial LiDAR data (DSM) subtracting standard (DEM), that is, nDSM=DSM-DTM.

C. LiDAR Data Set

Light Detection and Ranging (LiDAR) is an optical remote sensing technology that measures properties of scattered light to find the range and/or other information of a distant target. Our method does not require other information but LiDAR data. In this paper we use raw LiDAR data provided by the state of Louisiana [6]. The dataset was collected at a high emission rate of about 15,000 to 30,000 pulses per second. The resolution of the LiDAR data set is about 0.18 pt/m^2.

D. Variance of Normal Vectors Based on TIN Model

We use the new TIN-based classification approach—the variance of normal vectors at each node on TIN (Triangulated Irregular Network) to find the characteristics and detect vegetation or buildings from LiDAR raw data at urban areas. The below algorithms were implemented in Matlab.

Algorithm

```
Step01: Input the preprocessed LiDAR data (i.e., nDSM=DSM-DTM)

Step02: Use function Delaunary from Matlab Library to obtain a set of triangles

Step03: Find all triangles associated with the vertex p for each vertex p

Step04: Calculate and normalize each normal vector at the vertex p by using two vectors at two sides of each triangle

Step05: Count the number of normal vectors at the vertex p

Step06: Get the average normal vector $N_p$ at each vertex

Step07: Calculate the distribution of normal vectors (Standard Deviation Std or Variance Var) at the selected point p

Step08: Apply the criterion of Std or Var to determine if the point belongs to a building or vegetation

Step09: If Std ( or Var ) < threshold value
            Classify flat, e.g. building roof
        else
            Classify uneven, e.g. vegetation
        End if
```

Figure 1 shows the diagram for representation of a surface TIN and normal vectors distribution at node *p* on a surface TIN. The vectors ***A*** (or ***B***) are from two edges of a triangle starting from node *p*. ***N*** is the vector normal to the plane formed by vectors ***A*** and ***B***. The vectors ***N*** can be expressed as:

$$\boldsymbol{N}=\boldsymbol{A}\times\boldsymbol{B}= N_x\,\boldsymbol{i}+ N_y\boldsymbol{j}+ N_z\,\boldsymbol{k} \qquad (1)$$

As shown in Figure 1, three components of variance of normals are $var\_x=\sum_{i=1}^{m}\frac{(\overline{N_x}-N_{xi})^2}{m}$, $var\_y=\sum_{i=1}^{m}\frac{(\overline{N_y}-N_{yi})^2}{m}$ and $var\_z=\sum_{i=1}^{m}\frac{(\overline{N_z}-N_{zi})^2}{m}$, respectively. The number of normal vectors at the vertex *p* is expressed as *m*. Three components of the standard deviation of normal are $std\_x=\sqrt{\sum_{i=1}^{m}\frac{(\overline{N_x}-N_{xi})^2}{m}}$, $std\_y=\sqrt{\sum_{i=1}^{m}\frac{(\overline{N_y}-N_{yi})^2}{m}}$ and $std\_z=\sqrt{\sum_{i=1}^{m}\frac{(\overline{N_z}-N_{zi})^2}{m}}$, respectively.

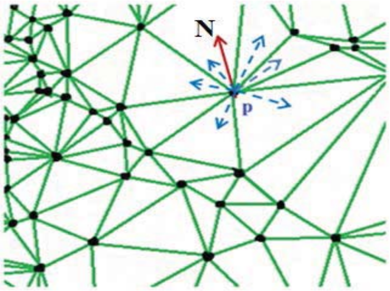


Fig.1. *TIN Representation of a surface TIN and normal vectors distribution at node p on a surface TIN*

III. Experimental Results And Analyses

In this paper, we preprocessed the raw LiDAR data to get normalized height. Due to the characteristics of the urban region, we employed the methods of height interval (separated levels) to reduce the errors of computing normal vectors. We only took the data when height *h*>9.14m in this paper (not including classes such as cars, shrubs, grass and roads, etc. under the height *h*<9.14m).

Figures 2~3 give the classification results using our method. Here we employed the proposed normal vector variation at each node on TIN to classify objects buildings /vegetation. We took the same threshold value for three components of standard deviation of normal vectors in this paper. The threshold we selected may be adjusted according to the different density of raw LiDAR data. The blue represents buildings. The green represents vegetation.

From figures 2~3, we find that the main buildings have been recognized, although there exists some green dots on the surface of buildings. In figure 3, the green dots obviously decrease on the roof of buildings when the threshold value of ***std*** is taken as 0.782.

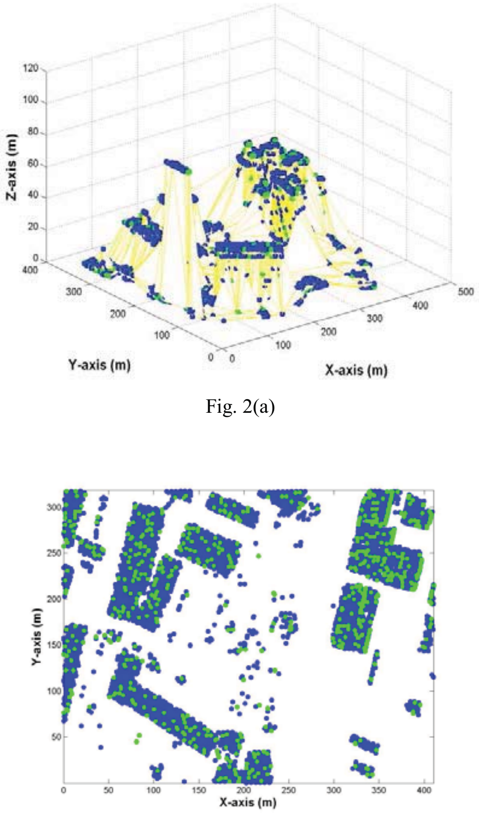


Fig. 2(a)

Fig. 2(b)

Fig.2. Classification results (the yellow represents lines for Triangulated Irregular Network, the blue represents buildings, and the green represents vegetation) (a) 3D classified result using our TIN-based classification approach at *h*>9.14m and *std* <0.482, (b) 2D classified result at *h*>9.14m and *std* <0.482

In our TIN-based classification approach, the category of determining each node depends on the distribution of normal vectors on the node which relates with its neighboring triangles. During the computation of variation of normal vectors, each normal vector at each node relates to two vectors on the two sides of each triangle. The distribution of triangles around the node and the relative height of the point greatly affect the classifying result. Thus, for the sparse LiDAR data at the urban areas, the height distribution of the roof of a building easily mixes with the distribution of normal vectors around the node which belongs to vegetation.

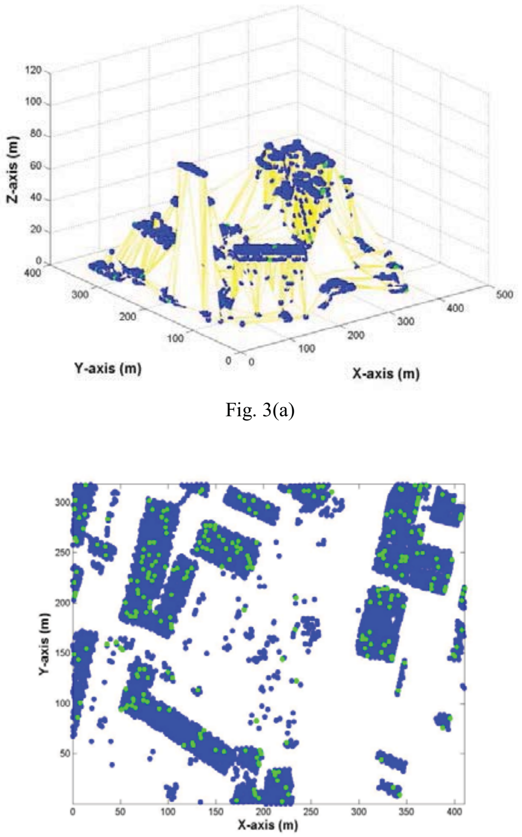


Fig. 3(a)

Fig. 3(b)

Fig.3. Classification results (the yellow represents lines for Triangulated Irregular Network, the blue represents buildings, and the green represents vegetation) (a) 3D classified result using our TIN-based classification approach at *h*>9.14m and *std* <0.782, (b) 2D classified result at *h*>9.14m and *std* <0.782

Although there exists little misclassification in the classified categories, as an effective feature extraction method combining with the properties of TIN, this method can be used for vegetation or buildings extraction via choosing the proper threshold of normal distribution. This method is more suitable for raw LiDAR data at large areas due to its simple algorithm and quick computation.

IV. Conclusions

In this paper, we present a method for vegetation/buildings extraction using low resolution LiDAR raw data. The contribution is to correctly extract vegetation/buildings from the raw LiDAR data set. We successfully used the fact that vegetation tends to have a larger derivation of normal vectors distribution than those of other objects.

Also, we have effectively classified the sparse LiDAR data point clouds using the new TIN-based method. The advantage of the proposed method includes that it can be employed for buildings or vegetation extraction via choosing the threshold value of normal vectors distribution at each node *p,* and that the proposed method reflects the neighbors' relationship of each node and reduces computational time since the proposed method only involves the direction (normal vector) distribution of triangular facets at each node of TIN. The further exploration and comparison of the proposed techniques will be completed in future work.

References

[1] G. Sithole, Filtering of Laser Altimetry Data Using a Slope Adaptive Filter, IAPRS, XXXIV (Pt.3/W4), p.203-210, 2001

[2] H-G. Maas, Least squares matching with airborne lasers canning data in a TIN structure, IAPRS, 33 (3) , p.548-555, 2000

[3] N. A. Akel, O. Zilberstein, and Y. Doytsher, Automatic DTM extraction from dense raw LiDAR data in urban areas, Proceedings of FIG working week, April 13-17, Paris, France, 10, 2003

[4] Q. Zeng, J. Mao, X. Li and X. Liu, LiDAR data filtering and classification with TIN and assistant plane, Geoinformatics 2007: Remotely Sensed Data and Information, edited by Weimin Ju, Shuhe Zhao, Proc. of SPIE Vol. 6752, 675206, 2007

[5] M. Y. Belkhouche, B. P. Buckles and L. J. Steinberg, " Vegetation Extraction from LiDAR raw points using surface flatness", Signal and Image Processing (SIP 2009), Honolulu, Hawaii, USA, August 17–19, 2009

[6] Data distributed by, "Atlas: The louisiana statewide GIS," LSU CADGIS Research Laboratory, Baton Rouge, LA, 109, http://atlas.lsu.edu

[7] S. Tang, P. Dong and B. P. Buckles, Three-dimensional surface reconstruction of tree canopy from LiDAR point clouds using a region-based level set method, International Journal of Remote Sensing, Vol. 34, No. 4, p1373-1385, 2013

[8] S. Tang, P. Dong and B. P. Buckles, A new method for extracting trees and buildings from sparse LiDAR data in urban areas, Remote Sensing Letter, 3, 3 p211–219, 2012

[9] S. Tang, P. Dong, B. P. Buckles, Comparison of two classification methods for feature extraction from LiDAR data in urban areas. The 2010 International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV'10), Vol. II, p553, 2010

Face detection in 3D images with more than one person

Juan Paduano, Marcelo Romero and Rosa María Valdovinos
{jpaduanos, mromeroh, rvaldovinosr}@uaemex.mx

Facultad de Ingeniería,
Universidad Autónoma del Estado de México,
Toluca, Estado de México, México

Abstract— *In this paper, we present an experimental analysis on the face detection problem using 3D face data. The novelty of the method presented is the automatic detection and localisation of every face in a 3D image with more than one person. Then, our method consists of four main steps: The first step, was to remove planar areas in the 3D images by the MSAC algorithm. In the second step, curvature analysis was used to detect points into convex elliptical areas. The third step, was to collect nose-tip candidate vertices using spin-images with an optimized neighbourhood search implementation based on the KD-Tree and KNN algorithms. Finally, in the fourth step, a classifier was used to get the best nose-tip candidate, using PCA and KNN. For testing, 3D images were collected using the Kinect One™ sensor, varying positions from one to eight different persons in each 3D image. Our method successfully detected 85% of faces in the 3D images. Additionally, our approach was tested in state of the art databases, FRGC & CurtinFaces, with a detection success of 98% and 97% respectively.*

Keywords: 3D Face detection, 3D Face processing, 3D feature descriptors

1. Introduction

Face detection is the first step in almost every face processing application, the face is localised and then extracted from an income image prior to specific analysis. Face detection is one of the visual tasks that humans do without effort. However, in computer vision, this task is not easy. Growing needs for automatic facial recognition systems and applications for automatic facial processing has encouraged the development of appropriate face detection algorithms.

Although 2D detection algorithms have reached an acceptable level, most of them work in impractical and controlled conditions, e.g. face recognition [1], [2], face tracking [3], pose estimation [4], facial expressions and facial gestures recognition [5]. The 2D images have a lot of difficulties when they are analyzed by computers [6], [7], [2], because the 3D world is projected down onto a 2D image, creating ambiguity and losing depth information. For example; Is it possible to recognize the same object from different viewpoints? or How do we deal with ambiguity between object size and distance from the camera? Additionally, How do we deal with illumination variation, that can make that the same object appears quite different when is imaged [7]. This research proposes a suitable face detection process by using only 3D data, because they provide explicit shape information and they are considered robust to illumination and pose variations [6].

The goal of 3D face detection could be defined as: given an arbitrary 3D image, we need to know if there are faces on it, regardless of number of people and position respect to the sensor. If at least one face in the image exist, must return its location coordinates. Unfortunately, face detection is a complicated problem because of different factors: position of the subject, type of 3D capturing image sensor, number of vertices of the facial surface, 3D image type and facial expression [6], [7], [8], [2].

Chellappa et al. [9], Bowyer et al. [6] and Kumar et al. [2] have discussed this factors in face detection. They analyzed related topics as segmentation and feature extraction, and classified the face detection algorithms in: feature-based detection and holistic face detection.

Although face detection is essential for many face processing applications, in literature it has received little attention, especially when using 3D data. Then, our overall research is aim to provide the state of the art algorithms for face detection in 3D data.

1.1 Related Work

Colombo et. al. [10] use curvature analysis, he started with a range image, i.e. an image where for each location (i,j) the coordinates (x,y,z) of the 3D scene are expressed with respect to the camera reference system [10]. He did calculated the mean (H) and Gaussian (K) curvatures in the range images and uses HK classification, divides the segmented regions into four types: convex, concave, and two types of saddle regions. The output of the processing step may contain any number of candidate facial features. If no nose or less than two candidate eyes are detected, they assume that no faces are present in the acquired scene, while there are no upper bounds on the number of features that can be detected and further processed. The final output of the procedure is a list containing the location and extension of each detected face.

Mian et al. [11] propose a nose tip detection method on horizontally sliced image contours. For each slice contour a triangle of maximum altitude is found using a circle whose center travels a long the slice contour. The vertices of the triangle are all on the slice contour with one vertex coinciding with the center of the circle and the other two vertices being the two intersections of the circle with the slice contour. The nose tip candidate for this slice contour is regarded as the center of the circle.Random Sample Consensus (RANSAC) is used to further remove outliers.Of the remaining nose tip candidates,the one that has the maximum confidence is taken as the nose tip. Finally extracted the face using a ratio.

Segundo et al. [12] present a method to extract the face region from an input range image containing only one subject. Their face segmentation algorithm is composed basically by two main stages: locating homogeneous regions in the input image by using clustering combined with edge data and identifying candidate regions that belong to the face region by an ellipse detection method based on the Hough Transform. They apply the K-Means algorithm by setting $k = 3$ to segment the image in three main regions: background, body, and face. However, this step alone is not enough to correctly extract the face region,then they apply edge detection. After performing region and edge detection, which can be made in parallel, they combine the two resulting images by using an AND operation for combined image and obtained the facial region.

Nair et al. [13] presented an accurate and robust framework based on the fitting of a facial model for face detection and segmentation, landmark localization and fine registration of face meshes. This model is based on a 3-D Point Distribution Model (PDM) that is fitted without relying on texture, pose, or orientation information. Fitting is initialized using candidate locations on the mesh, which are extracted from low-level curvature-based feature maps. Face detection is performed by classifying the transformations between model points and candidate vertices based on the upper-bound of the deviation of the parameters from the mean model. The performance of face detection is evaluated on a database of faces and non-face objects where they achieve an accuracy of 99.6%. They also demonstrate face detection and segmentation on objects with different scale and pose.

Maes et al. [14] presented a SIFT algorithm adapted for 3D surfaces (called meshSIFT) and its applications to 3D face pose normalisation and recognition. The algorithm allows reliable detection of scale space extrema as local feature locations. The meshSIFT algorithm then describes the neighbourhood of every scale space extremum in a feature vector consisting of concatenated histograms of shape indices and slant angles. The feature vectors are reliably matched by comparing the angle in feature space.

Earlier, we presented and experimental analysis on the face detection problem using 3D face data [15], we have identified three key published papers that use curvature analysis, a slicing approach and a segmentation technique. Then, we have defined an experimental procedure to investigate those papers using the Face Recognition Grand Challenge database for a performance comparison and analysis.

Additionally, other research that indirectly perform face detection in 3D images are identified. For example, [16] proposed a nose tip detection method that has the following characteristics. First, it does not require training and does not rely on any particular model. Second, it can deal with both frontal and non-frontal poses. Finally, it is quite fast, requiring only seconds to process an image of 100 to 200 pixels (in both x and y dimensions). In [17], thresholds are set for two curvature maps to search for the nose tip, the two eyes,and the nose ridge. [18] use an hybrid approach using ICP and ASM fitting for non-rigid registration of a dense surface model on 3D faces. This method does not require texture, it imposes constraints on the orientation of the face and is not scale invariant. Finally, Nanni et al.[19] presents a face detector based on Viola Jones algorithm.

The rest of this paper is divided as follows. Section 2 describes the data bases used for this research. Section 3 introduces our face detection procedure in 3D images with more than one person and finally, Section 4 concludes this paper.

2. 3D Face Data Base

For this investigation, we are using two state of the art databases: Face Recognition Grand Challenge [20] and CurtinFaces [21]. In recent years the database FRGC has traditionally been the most widely used set of biometric data research. However, nowadays the Microsoft 3D sensor has made available an economic solution. It has attracted the research community for working with 3D data and to create databases using this sensor as the CurtinFaces database.

From the FRGC database, we are using all sets: Spring 2003, Fall 2003 and Spring 2004 consisting of 4950 2D/3D face images from different subjects acquired under different conditions. For this publication we are reporting performance over the complete set.

The CurtinFaces database consists of 97 images from 52 different persons. This data was collected using a Kinect 360™ sensor. Images where captured with variations in head pose, illumination, facial expressions and occlusion. For this paper, we have used only 1437 3D images from persons in front pose and not wearing glasses.

2.1 Experimental data acquisition

As we know, state of the art 3D databases (as FRGC and CurtinFaces) have only one person per image and they were collected in controlled conditions as, illumination, mostly neutral facial expressions and front pose.

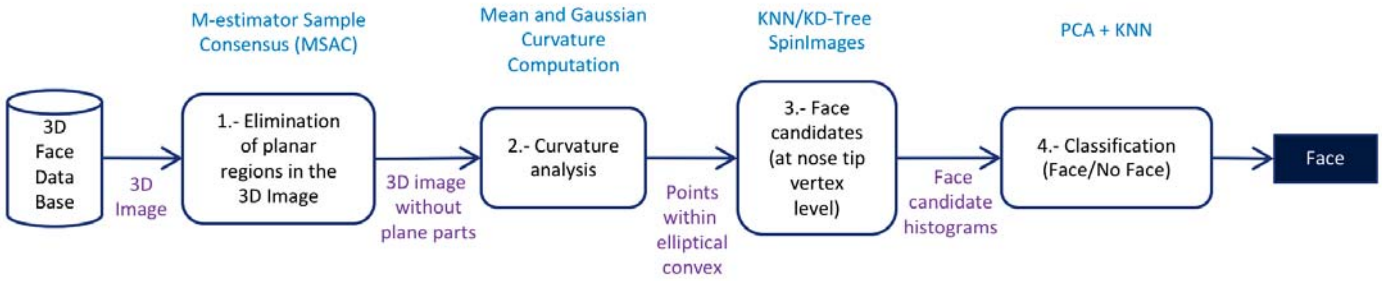

M-estimator Sample Consensus (MSAC)
Mean and Gaussian Curvature Computation
KNN/KD-Tree SpinImages
PCA + KNN
3D Face Data Base
3D Image
1.- Elimination of planar regions in the 3D Image
3D image without plane parts
2.- Curvature analysis
Points within elliptical convex
3.- Face candidates (at nose tip vertex level)
Face candidate histograms
4.- Classification (Face/No Face)
Face

Fig. 1: Experimental procedure to detect every face in a 3D image consisting of four main steps.

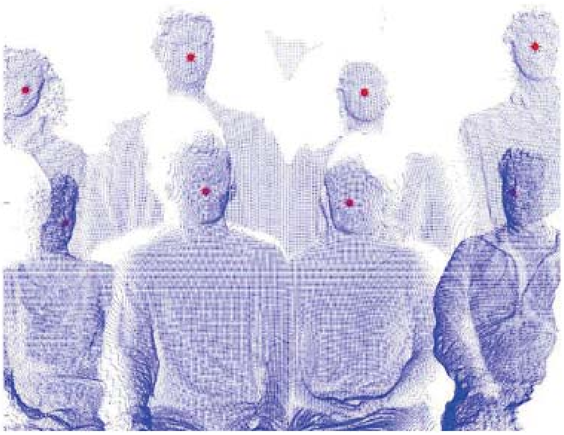


Fig. 2: Ground truth landmarking, in this figure every red-dot correspond to the vertex manually selected as the nose of a face.

Then, a capture session was performed using Kinect One™ sensor, to collect 2D/3D images by varying the number and position of eight subjects in the scene. Figure 4 shows how people were placed within the field of view of the 3D sensor.

Our first data collection is limited by using at maximum eight persons in each scene because the Kinect's ™ field of view is not able to capture more people without reducing the number of vertices per face. This is well known as a challenge in 3D face processing [6].

Given these reasons, four sets of images were collected varying in each scene the number and the position of subjects, and it was possible to get 2^8 different positions, counting 1020 images in total without empty cases.

For training and performance evaluation it was necessary to collect ground truth values. We assumed that a face exists into an image if its nose is present as well. Therefore, meticulously, we collect ground truth values at nose tip vertex level, by manually identifying the best candidate in a Matlab 3D plot, by selecting the nose as the most prominent part in a face (see Figure 2).

3. Face detection in 3D images with more than one person

As the experimental procedures for face detection in literature [15], we have designed and evaluated a novel approach for face detection in 3D images. It consist of four main steps as illustrated in Figure 1. In general terms, the first step is to eliminate planar regions in the 3D image using M-estimator sample consensus (MSAC) algorithm. The second step curvature analysis was used for detecting convex elliptical points. The third step is nose tip candidate selection using SpinImages with KD-Tree and KNN. The fourth step is a Face/No-Face classification using PCA and our centroids technique. Finally, we extracted the face using an 8cm radius sphere centred at the selected nose tip candidate.

To test our experimental procedure to detect any face in a 3D image, our set of 1020 depth images was used. We defined different training and testing sets as indicated in Table 1.

Table 1: Training and testing sets of depth images with one to eight different people.

Set	Number of images	Number of faces
Training	100	200
Testing	920	3896
Total	1020	4096

Using the experimental data set shown in Table 1, we evaluated our experimental face detection procedure using 3D images containing more than one person in the scene. Our experimental procedure is as follows:

1) For every 3D image, planar areas were eliminated using M-estimator Sample Consensus (MSAC), MSAC is a modification of the RANSAC algorithm, where the aim is to adjust planes in the point cloud using a distance d, when the plane is set, the distance of the points is calculated to it in order to obtain two sets of points, inliers and outliers, where inliers are the

vertices that fit the calculated plane and the outliers are those whose distance to the plane is greater that d. The steps for implementation of the MSAC technique are: The input of the MSAC algorithm is a point cloud and distance d. A plane is adjusted to a part of the 3D image with the operation (2), using the equation of the plane:

$$P1\cdot X + P2\cdot Y + P3\cdot Z + P4 = 0 \tag{1}$$

$$\begin{bmatrix} X1 & Y1 & Z1 & 1 \\ X2 & Y2 & Z2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ Xn & Yn & Zn & 1 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix} = A\bar{p} \tag{2}$$

The optimal parameters are calculated. We defined the optimal parameters as the minimum values of plane adjustment, shown in next equation:

$$\bar{p}opt = argmin||A\bar{p}||^2 \tag{3}$$

Calculate the estimation error of the distance of each point to the plane:

$$e^2 = \frac{([x\ y\ z\ 1]\bar{p}opt)^2}{P_1^2 + P_2^2 + P_3^2} \tag{4}$$

Select the inliers using a distance d. If the distance of a point is less or equal than the value of d it is considered inliers, otherwise are outliers.

We selected the outlier because technically they are points that are not planar regions and them could be a face. The Figure 5 shows the result of this step.

2) Then, we used curvature analysis for detecting convex elliptical points, the steps for implementation of the curvature analysis are:

We started with a range image that is a representation of the coordinates (x,y,z) of the 3D image in a location (i,j). The Depth images are computed from respective 3D point cloud using equations (5) and (6).

$$f \rightarrow U\ defined\ on\ an\ open\ set\ U \subseteq R^2 \tag{5}$$

$$S = (x, y, z)|(x, y) \in U; z \in R; f(x, y) = z \tag{6}$$

For every point on the depth map, Mean (7) and Gaussian (8) curvature are obtained by calculating the first (9) and second derivatives as [22].

$$H(x,y) = \frac{(1+f_y^2)f_{xx} - 2f_xf_yf_{xy} + (1+f_x^2)f_{yy}}{2(1+f_x^2+f_y^2)^{\frac{3}{2}}} \tag{7}$$

$$K(x,y) = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1+f_x^2+f_y^2)^2} \tag{8}$$

where $f_x, f_y, f_{xy}, f_{xx}, f_{yy}$ are the first and second derivatives of f in (x, y). A face is initially represented by a range image of points. Since we have only a discrete representation of S, estimate the partial derivatives. For each point the depth map, we considered the equations (9) and (10):

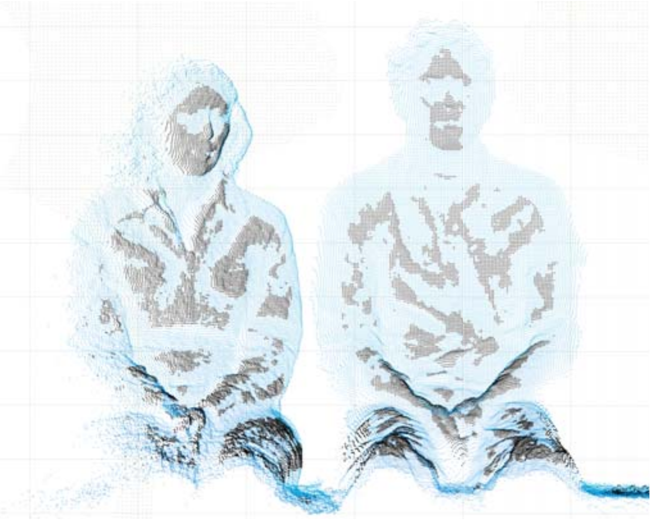


Fig. 3: The black vertices are convex elliptical points

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} + O(h^2) \tag{9}$$

$$f''(x_0) = \frac{f(x_0+h) - 2f(x_0) + 2f(x_0-h)}{h^2} + O(h^2) \tag{10}$$

where x_0 is (x, y) in the range image, h is an integer greater than zero, and $0(h^2)$ is the *truncation error*, caused by stopping the polynomial approximation to second order, which tends to zero.

Then, by analyzing the signs of the mean and the Gaussian curvature, we perform what is called an HK classification [23].

Finally, we have only selected the convex elliptical points (see Figure 3).

3) At this point, we mostly get points around the subjects body and head.

4) For every convex elliptical point in the 3D point cloud, *Spin Images* was used, as prescribed by Johnson and Hebert [24] in a training set of 200 different face images. The steps for implementation of the *spin Images* technique are:

Select a point P in the point cloud. Calculate the normal of P using a radius r to select neighbours P. α and β are calculated by using (11) and (12) equations respectively.

$$\alpha = \sqrt{||x-p||^2 - (n\cdot(x-p))^2} \tag{11}$$

$$\beta = (n\cdot(x-p)) \tag{12}$$

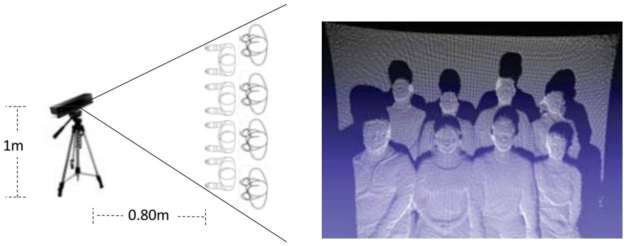


Fig. 4: A one-depth image with eight people in the scene

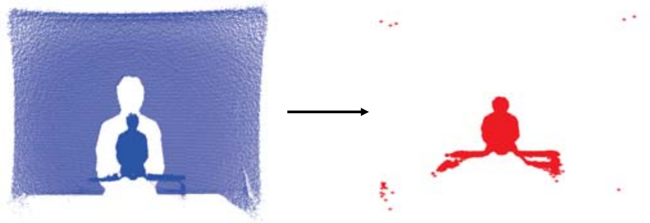


Fig. 5: Planar areas elimination using an MSAC algorithm implementation.

Values i and j are calculated using a ***bin***, where ***bin*** is a constant, this research is equal to 0.2:

$$i = [\frac{\beta max - \beta}{bin}] \quad , \quad j = [\frac{\alpha}{bin}] \tag{13}$$

To speed up the selection process for spin image histogram creation, we used *Nearest Neighbour* and *KD-Tree*. Figure 6 shows some 3D point cloud faces and spin image histogram samples. Spin images was trained with 200 faces. In testing, an spin image histogram for each vertex of the point cloud is calculated.

5) Face no-face classification was done using PCA on local maximum vertices.
6) Finally, face extraction was done by using an 8cm radius.

4. Preliminary results

Using our face data set containing from one to eight persons in each scene we have experimentally evaluated our face detection procedure. In this case, from the resulting spin image classification we find first the local maximum vertex for every cluster of candidates. Then, using different thresholds we compare the localisation errors from every vertex to its respective ground truth value. Figure 7 shows a classification result, while the Tables 2 and 3, summarises our localisation procedure's performance. As we can see, we are successfully detecting 85% of faces among our experimental face images using a 16mm threshold and 98% of the FRGC and 97% of the CurtinFaces databases respectively (see Figure 8).

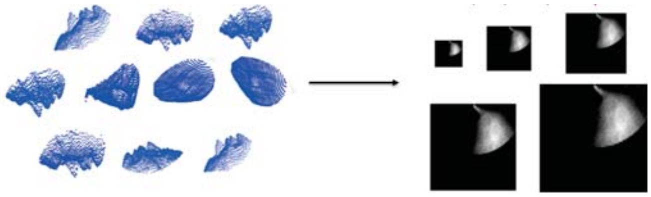


Fig. 6: Face 3D point cloud and spin images histogram samples.

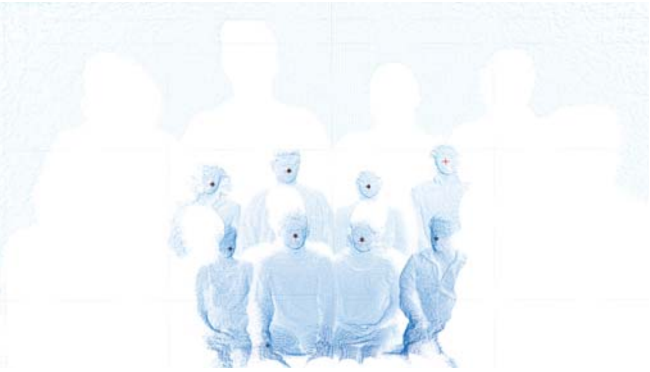


Fig. 7: In this point cloud image, eight people were captured. Our face detection approach is successfully detecting seven faces and one is missed.

Table 2: This table shows some techniques for face detection, used databases and the obtained results.

Technique	DataBase	Images	Performance [%]
Curvature analysis [10]	DISCo	150	96.85%
Slicing analysis [11]	FRGC	4500	98.30%
Face segmentation [12]	FRGC	4950	99.00%
Face-fits [13]	GavabDB + NTU	827	99.60%
Our approach	FRGC	4950	98.00%
	CurtinFaces	1237	97.00%

5. Conclusions

In this paper we have presented a novel experimental procedure for face detection, based on four main steps, that is able to successfully detect every contained face in one thousand twenty 3D images. Those images have been collected using one to eight different subjects at different locations in each scene. Additionally, we have tested our face detection procedure with state of the art databases (FRGC and CurtinFaces) that were collected, considering only one

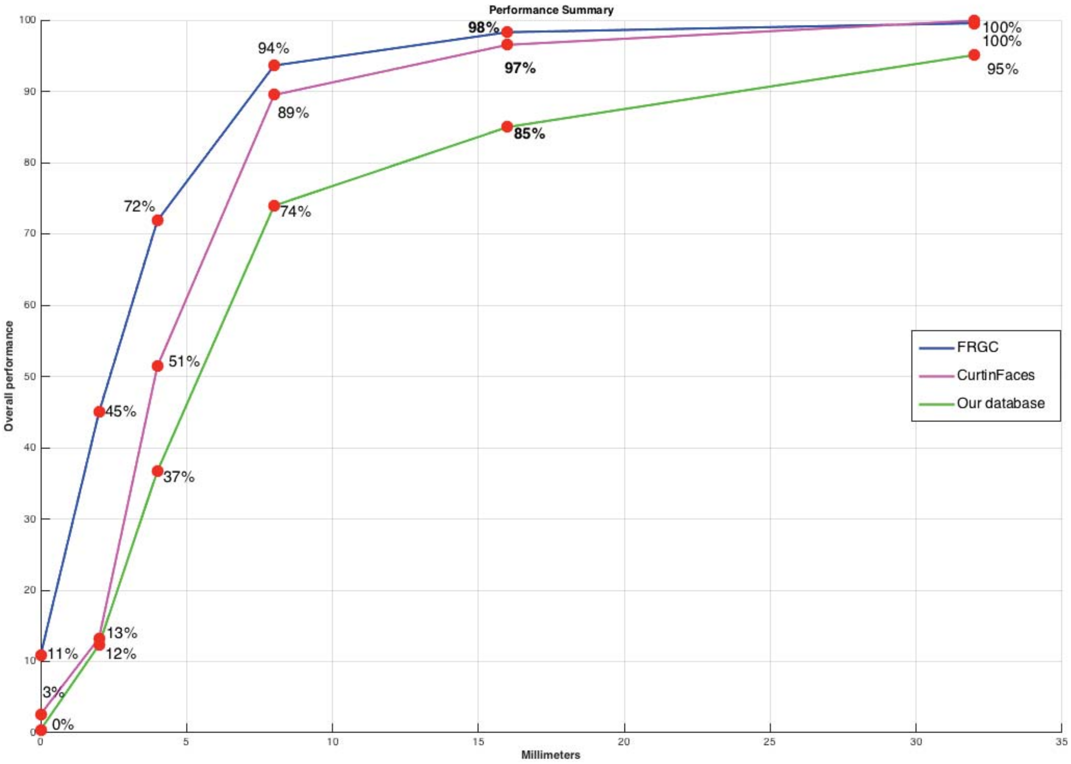


Fig. 8: Performance Summary.

person per image and we have confirmed our detection procedure success.

For the three datasets experimented in this paper: (1) Our database with more than one subject, (2) FRGC, and (3) CurtinFaces, we have gathered location errors by computing the Euclidean distance from the selected face and their respective ground-truth at the nose tip level. Results illustrated in Table 3 shows that 98%, 97% and 85% of the faces in every image of the FRGC, CurtinFaces and our dataset with more than one subject, respectively are located between 0 to 16 mm.

Table 3: Face detection summary using FRGC, CurtinFaces and our point cloud images containing from one to eight subjects at 16mm.

Data Bases	Performance [%]
FRGC	98%
CurtinFaces	97%
Our DataBase	85%

Those results are encouraging our research and allow us to draw possible venues for future work. In the mean time we are collecting more experimental images with more than one person in the scene to further research to confirm our results. Also, we are including more powerful classifiers than PCA within our experimental procedure.

Acknowledgments

Authors thank to the Research Department of the Autonomous University of the State of Mexico (UAEMex) for their financial support, grant 3720/2014/CID. Also, our gratitude goes to CONACYT for Mr. Paduano′s scholarship 634473.

References

[1] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, “Toward a practical face recognition system: Robust alignment and illumination by sparse representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 2, pp. 372–386, 2012.

[2] A. Kumar, M. Datta, and P. Kumar, *Face Detection and Recognition: Theory and Practice*. Chapman and Hall/CRC, 2015.

[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1442–1468, 2014.

[4] P. Wohlhart and V. Lepetit, “Learning descriptors for object recognition and 3d pose estimation,” in *Computer Science: Computer Vision and Pattern Recognition*, feb 2015. [Online]. Available: http://adsabs.harvard.edu/abs/2015arXiv150205908W

[5] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer, “Meta-analysis of the first facial expression recognition challenge,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 966–979, 2012.

[6] K. W. Bowyer, K. Chang, and P. Flynn, “A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition,” *Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1–15, 1 2006.

[7] N. Pears, Y. Liu, and P. Bunting, *3D Imaging, Analysis and Applications*, 1st ed., P. B. Nick Pears, Yonghuai Liu, Ed. Springer-Verlag London, 2012, vol. 1, no. 2012939510.

[8] M. Yang, Z. Feng, S. C. K. Shiu, and L. Zhang, “Fast and robust face recognition via coding residual map learning based adaptive masking,” *Pattern Recognition*, vol. 47, no. 2, pp. 535–543, 2 2014.

[9] R. Chellappa, C. Wilson, and S. Sirohey, “Human and machine recognition of faces: a survey,” *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, May 1995.

[10] A. Colombo, C. Cusano, and R. Schettini, “3d face detection using curvature analysis,” *Pattern Recognition*, vol. 39, no. 3, pp. 444 – 455, Mar 2006.

[11] A. Mian, M. Bennamoun, and R. Owens, “Automatic 3d face detection, normalization and recognition,” in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, 2006, pp. 735–742.

[12] M. P. Segundo, C. Queirolo, O. R. P. Bellon, and L. Silva, “Automatic 3d facial segmentation and landmark detection,” in *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, 2007, pp. 431–436.

[13] P. Nair and A. Cavallaro, “3-d face detection, landmark localization, and registration using a point distribution model,” *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 611–623, June 2009.

[14] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, “Feature detection on 3d face surfaces for pose normalisation and recognition,” in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, Sept 2010, pp. 1–6.

[15] J. Paduano, M. Romero, and M. V., “Toward face detection in 3d data,” in *International Conference on Image Processing, Computer Vision, & Patter recognition*, vol. 15, Jul 2015, pp. 473–479. [Online]. Available: http://worldcomp-proceedings.com/proc/proc2015/ipcv.html

[16] X. Peng, M. Bennamoun, and M. A. S., “A training-free nose tip detection method from face range images,” *Pattern Recognition*, vol. 44, no. 3, pp. 544 – 558, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320310004553

[17] K. I. Chang, K. W. Bowyer, and P. J. Flynn, “Multiple nose region matching for 3d face recognition under varying facial expression,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1695–1700, Oct 2006.

[18] T. J. Hutton, B. F. Buxton, and P. Hammond, “Automated registration of 3d faces using dense surface models,” in *In Proc. British Machine Vision Conference*, 2003, pp. 439–448.

[19] L. Nanni, A. Lumini, F. Dominio, and P. Zanuttigh, “Effective and precise face detection based on color and depth data,” *Applied Computing and Informatics*, vol. 10, no. 1Ð2, pp. 1 – 13, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221083271400009X

[20] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, “Overview of the face recognition grand challenge,” *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 947–954 vol. 1, 2005.

[21] B. Y. L. Li, A. S. Mian, W. Liu, and A. Krishna, “Using kinect for face recognition under varying poses, expressions, illumination and disguise,” in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, Jan 2013, pp. 186–192.

[22] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Illustrated, Ed. Prentice Hall, 1998.

[23] P. J. Besl and R. C. Jain, “Invariant surface characteristics for 3d object recognition in range images,” *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 33 – 80, 1986.

[24] A. E. Johnson and M. Hebert, “Surface matching for object recognition in complex three-dimensional scenes,” *Image and Vision Computing*, vol. 16, no. 9–10, pp. 635–651, 7 1998.

FaceDNA: Intelligent Face Recognition System with Intel RealSense 3D Camera

Dan Ye[1], Shih-Wei Liao[1]

[1]National Taiwan University, Department of Computer Science and Information Engineering

Abstract—This paper develops an intelligent face recognition system which has been applied to Intel Realsense 3D camera. The key components include computer vision application, face tracking, personal attributes, FaceDNA, emotion detection, video application modules. Computer vision application can extract rich information from images to categorize and process visual data. FaceDNA is consist of face verification, face identification, face detection, face matching, face grouping. It can detect human faces and compare similar ones, organize people into groups according to visual similarity, and identify previously tagged people in images. Emotion application can analyze faces to detect a range of feelings and personalize your responses. Intelligent video processing stable video output, detect motion, creates intelligent thumbnails, and detects and tracks faces.

Keywords: FaceDNA, Intel RealSense 3D camera, face tracking and recognition.

1 Introduction

The Intel® RealSense™ SDK includes a face tracking module provides a suite of the following face algorithms: Face detection locates a face (or multiple faces) from an image or a video sequence, and returns the face location in a rectangle. You can use this feature to count how many faces are in the picture and find their general locations. Landmark detection further identifies the feature points (eyes, mouth, etc.) for a given face rectangle. The eye location is of a particular interest for applications that change display perspectives based on where on the screen users are looking. Pose detection estimates the face orientation where the user's face is looking. Expression detection calculates the scores for a few supported facial expressions such as eye-closed and eye-brow turning up. Face recognition feature compares the current face with a set of reference pictures in the recognition database to determine the user's identification. Pulse estimation tracks subtle change in face skin color over time and estimates the person's pulse rate. Gaze tracking traces human eye movement and provide estimated eye gaze location on the display screen and angles from the origin.

This paper constructs a novel face recognition system that integrates computer vision application, face tracking, personal attributes, FaceDNA, emotion detection, video application modules. FaceDNA can predict past, future face photo using current face photo which captured by Intel® RealSense™ 3D Camera. According to the changing facial feature various with growing up on same person, infer his or her photoes at different ages. FaceDNA can deduce blood relationship. Analyze family relationship through feature exaction of multiple faces based on human genetic inheritance, FaceDNA can infer relatives among farther, mother, brother or sister. Without the detection of DNA, matching faces of people will return their internal relationship. Through the face photo of parents, FaceDNA can infer photo of their child using image morphing. The attribute festures of face photo can infer the basic information of people such as Name, Age , Gender, race, identity, clothes, gesture, event, memorability. Various emtions can demonstrate their confidence values in each facial expression. Recognize the emotion of face image, record the changes of emtion in the short video, and capture emotion values in real-time.

To our best knowledge, predicting image memorability [1, 2, 3], using deep learning and LaMem, a novel diverse dataset, initiates a novel method to achieve unprecedented performance at estimating the memorability ranks of images, and evaluate memorability maps. New visual materials could be enhanced using the memorability maps approach, to reinforce forgettable aspects of an image while also maintaining memorable ones [4].

The reminder of paper is organized as follows. Section II describes a comprehensive overview of intelligent face recognition system design with face tracker and detection as well as face matching, face verification, face grouping, automatic memorability predictor. Section III depicts that the methodology of predicting face memorability. Implementation on real experiment environment is introduced in section IV. Experiment results on the previous discussed features are presented in section V. Finally, conclusions are reiterated in section VI.

2 System Design

2.1 FaceDNA

Detect one or more human faces in an image and get back face rectangles for where in the image the faces are, along with face attributes which contain machine learning-based predictions [5, 6] of facial features. After detecting faces, you can take the face rectangle. The face attribute features available are: Age, Gender, Pose, Smile, and Facial Hair along with 27 landmarks for each face in the image.

2.2 Face Verification[7]

Check the likelihood that two faces belong to the same person. The API will return a confidence score about how likely it is that the two faces belong to one person. For this application, if providing photos, please use images which contain only a single face.

2.3 Face Identification[8]

Search and identify faces. Tag people and groups with user-provided data and then search those for a match with previously unseen faces.

2.4 Similar Face Searching

Easily find similar-looking faces. Given a collection of faces and a new face as a query, this API will return a collection of similar faces.

2.5 Face Grouping

Organize many unidentified faces together into groups, based on their visual similarity.

2.6 Face relationship[9]

Analyze family relationship through feature exaction of multiple faces based on human genetic inheritance, inference relatives among father, mother, brother or sister. Just like the detection of DNA, matching faces of people will return their internal relationship.

2.7 Analyze attributes

This feature returns information about visual content found in an image. Use tagging, descriptions and domain-specific models to identify content and label it with confidence. Apply the adult/racy settings to enable automated restriction of adult content. Identify image types and color schemes in pictures. Please try vision feature analysis by uploading a local image, or providing an image URL.

2.8 Recognize celebrities

The Celebrity Model is an example of Domain Specific Models. Our new celebrity recognition model recognizes 200K celebrities from business, politics, sports and entertainment around the World. Domain-specific models is a continuously evolving feature.

2.9 Read text in images

Optical Character Recognition (OCR) detects text in an image and extracts the recognized words into a machine-readable character stream. Analyze images to detect embedded text, generate character streams and enable searching. Allow users to take photos of text instead of copying to save time and effort. Please try vision optical character recognition by uploading a local image, or providing an image URL.

2.10 Recognize Emotions in Images

The Emotion application takes an facial expression in an image as an input, and returns the confidence across a set of emotions for each face in the image, as well as bounding box for the face. A user can submit the face rectangle as an optional input. The emotions detected are anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. These emotions are understood to be cross-culturally and universally communicated with particular facial expressions. Emotion application uses world-class machine learning techniques [10] to provide these results. You can also click the open image button or drag-and-drop to upload your own images, or input a URL for a remote image.

2.11 Recognize Emotions in Video

The Emotion application for Video [11] recognizes the facial expressions of people in a video, and returns an aggregate summary of their emotions. You can use this application to track how a person or a crowd responds to your content over time. The emotions detected are anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise.

2.12 Automatic memorability predictor

This section applies a probabilistic framework that models how and which local regions from an image may be forgotten using a data-driven approach that combines local and global images features. The model automatically discovers memorability maps of individual images without any human annotation. We incorporate multiple image region attributes in this algorithm, leading to improved memorability prediction of images.

Made predictions on the basis of a suite of global image features pixel histograms, GIST, SIFT, HOG, SSIM. Running the same methods on current 2/3 data splits. Do better by using our selected features as an abstraction layer between raw images and memorability. We trained a suite of SVRs to predict annotations from images, and another SVR to predict memorability from these predicted annotations. For annotation types, we used the feature types selected by our 100-bit predictive selection on 2/3 training sets. To predict the annotations for each image in our training set, we spilt the training set in half and predicted annotations for one half by training on the other half, and vice versa, covering both halves with predictions. We then trained a final SVR to predict memorability on the test set in three ways: 1) using only image features (Direct), 2) using only predicted annotations (Indirect), and 3) using both (Direct + Indirect). Combining indirect predictions with direct predictions performed best, slightly outperforming the direct prediction method.

3 Predicting Memorability of face

In this section, we explore various features for predicting face memorability and apply a robust memorability metric to significantly improve face memorability prediction. We also note that the task of automatically predicting the memorability of faces using computer vision features.

For predicting memorability, dense global features such as HOG and SIFT significantly outperform landmark-based features such as 'shape' by about 0.15 rank correlation. This implies that it is essential to use these features in our face modification algorithm to robustly predict memorability after making modifications to a face. While powerful for prediction, the dense global features tend to be computationally expensive to extract, as compared to shape. Shape is used in this algorithm to parameterize faces so it essentially has zero cost of extraction for modified faces. However, as compared to memorability, the gap in performance between using shape features and dense features is not as large for other attributes. Hence, we use landmark-based features instead of dense global features for the modification of facial attributes.

4 Experiment Implement

Development environment: Windows 10 on MacBook Air, Microsoft Visual studio professional 2015, Inter realsense camera F200, Intel realsense SDK 2016 R1, SDK Runtime Distributable, F200 Camera Driver (Depth Camera Manger), Microsoft Cognitive Services.

Pre-trained CNN using Caffe deep learning toolbox, it provides the network deploy file, the trained network model, and the train-val file which can be loaded using Caffe. Extract CaffeNet / AlexNet features using the Caffe utility. MemNet trained on first train/test split of LaMem, with FA, rank correlation with 10 crops per image 0.64. Then randomly select 500 different images in emotion from their dataset [12], and collected 80 scores per image. After applying this algorithm to correct the memorability scores, it can obtain a within-dataset human rank correlation of 0.77 (averaged over 25 random splits). Furthermore, it can obtain a rank correlation of 0.76.

5 Experiment Result

Fig.1 demonstrates emotion and gaze detection, face tracking by Intel Realsense 3D camera. Fig.2 indicates FaceDNA extracts various attributes of face photos. FaceDNA includes face detection, verification, face matching, and face grouping modules. Fig.3 depicts emotion detection recognizes the expression of face image, record the changes of emotion in the video, and capture emotion values in real-time. Fig.4 plots the relationship of memorability scores and various emotion image attributes.

6 Conclusions

This paper builds an intelligent face recognition system, which can be applied into Intel Realsense 3D camera. FaceDNA is the new feature in analyzing the face images captured by Intel Realsense 3D Camera. It covers the most advanced technical solutions for face tracking and recognition. The key contribution is to apply a novel face recognition system in realizing cognitive services. Moreover, this system applies computer vision techniques to extract memorability automatically.

7 References

[1].A. Khosla, A. S. Raju, A. Torralba and A. Oliva, Understanding and Predicting Image Memorability at a Large Scale, International Conference on Computer Vision (ICCV), 2015.

[2].Aditya Khosla, Wilma A. Bainbridge, Antonio Torralba and Aude Oliva, Modifying the Memorability of Face Photographs, International Conference on Computer Vision (ICCV), 2013.

[3].Phillip Isola, Jianxiong Xiao, Antonio Torralba, Aude Oliva. What makes an image memorable?, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[4].R. Dubey, J. Peterson, A. Khosla, M.-H. Yang, and B. Ghanem. What makes an object memorable? In International Conference on Computer Vision (ICCV), 2015.

[5].Shengcai Liao, Anil K. Jain, and Stan Z. Li, "A Fast and Accurate Unconstrained Face Detector," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.

[6].X. Wang and X. Tang. Random sampling for subspace face recognition. International Journal of Computer Vision, 70(1), 2006.

[7].N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile classifiers for face verification. In Proc. IEEE International Conference on Computer Vision (ICCV), 2009.

[8].M. Guillaumin, J. Verbeek, C. Schmid, I. LEAR, and L. Kuntzmann. Is that you? Metric learning approaches for face identification. IEEE International Conference on Computer Vision (ICCV), 2009.

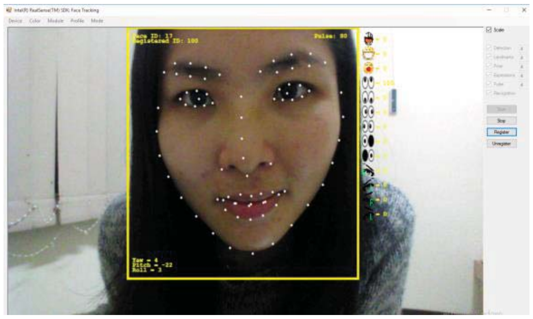
[9].Y. Su, S. Shan, X. Chen, and W. Gao. Adaptive generic learning for face recognition from a single sample per person. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

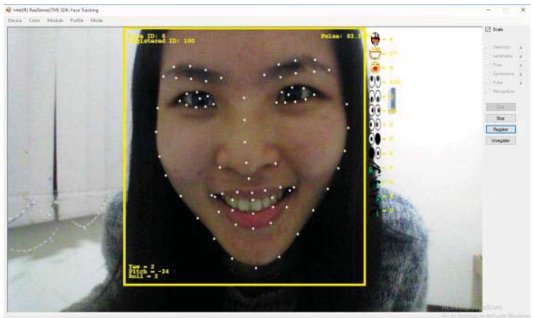
[10].Z. Cao, Q. Yin, J. Sun, and X. Tang. Face recognition with Learning-based Descriptor. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

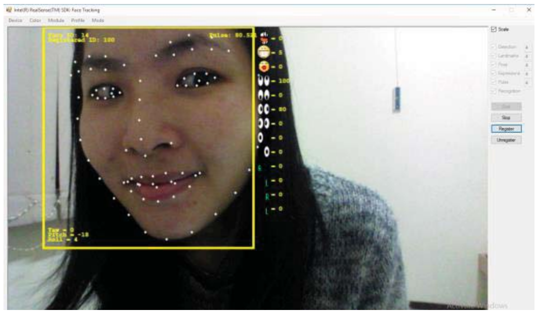
[11].F. De la Torre, W.S. Chu, X. Xiong, F. Vicente, X. Ding, and J. F. Cohn. IntraFace, IEEE International Conference on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia. , 2015.

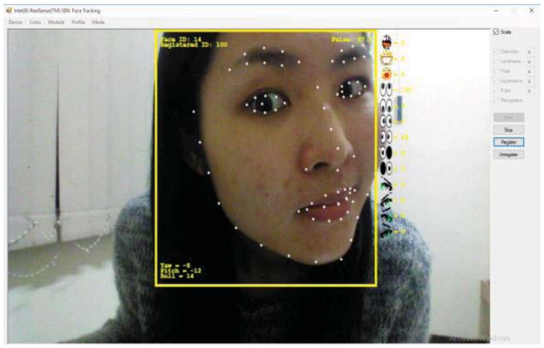
[12].N.Pinto,J.DiCarlo,and D.Cox.How far can you get with a modern face recognition test set using only simple features. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2009.

Fig.1. Face tracker









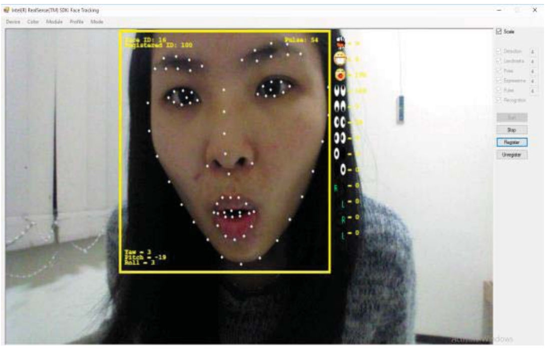
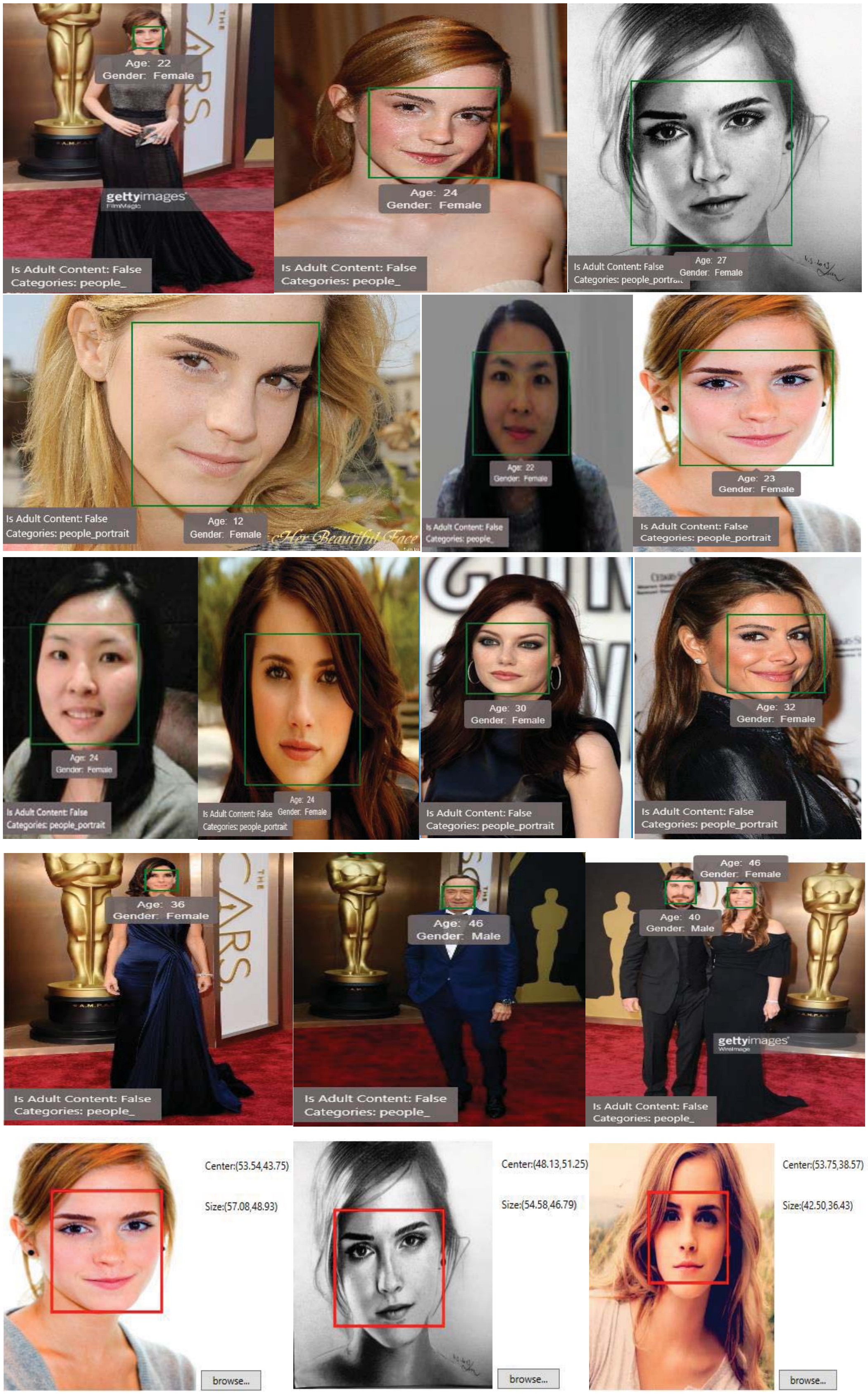


Fig.2.FaceDNA



Age: 22 Gender: Female

Is Adult Content: False Categories: people_

Age: 24 Gender: Female

Is Adult Content: False Categories: people_

Is Adult Content: False Categories: people_portrai

Age: 27 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 12 Gender: Female

Age: 22 Gender: Female

Is Adult Content: False Categories: people_

Age: 23 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 24 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 24 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 30 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 32 Gender: Female

Is Adult Content: False Categories: people_portrait

Age: 36 Gender: Female

Is Adult Content: False Categories: people_

Age: 46 Gender: Male

Is Adult Content: False Categories: people_

Age: 46 Gender: Female

Age: 40 Gender: Male

Is Adult Content: False Categories: people_

Center:(53.54,43.75)

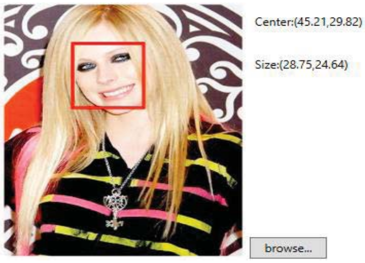
Size:(57.08,48.93)

Center:(48.13,51.25)

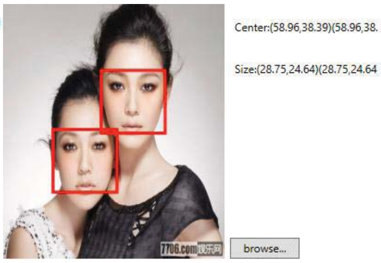
Size:(54.58,46.79)

Center:(53.75,38.57)

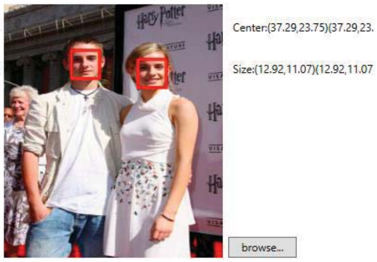
Size:(42.50,36.43)


Center:(45.21,29.82)

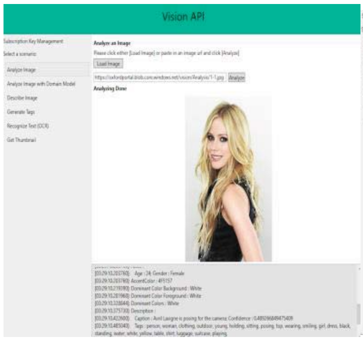
Size:(28.75,24.64)


Center:(58.96,38.39)(58.96,38.

Size:(28.75,24.64)(28.75,24.64


Center:(37.29,23.75)(37.29,23.

Size:(12.92,11.07)(12.92,11.07



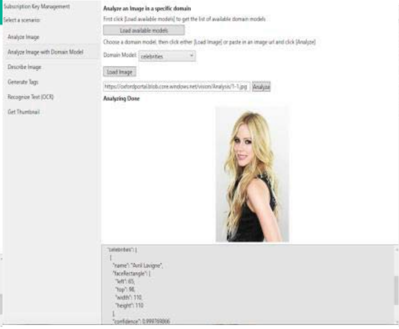
Analyze an Image

Please click either [Load Image] or paste in an image url and click [Analyze]

https://oxfordportal.blob.core.windows.net/vision/Analysis/1-1.jpg

Analyzing Done

```
[03:29:10.203760]:    Age : 24, Gender : Female
[03:29:10.203780]: AccentColor : 4F5157
[03:29:10.219390]: Dominant Color Background : White
[03:29:10.281960]: Dominant Color Foreground : White
[03:29:10.328644]: Dominant Colors : White
[03:29:10.375730]: Description :
[03:29:10.422600]:    Caption : Avril Lavigne is posing for the camera; Confidence : 0.489266849475408
[03:29:10.485040]:    Tags : person, woman, clothing, outdoor, young, holding, sitting, posing, top, wearing, smiling, girl, dress, black, standing, water, white, yellow, table, shirt, luggage, suitcase, playing
```



Analyze an Image in a specific domain

First click [Load available models] to get the list of available domain models

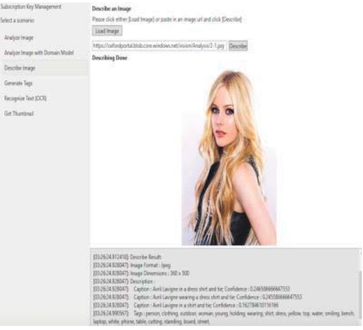
Choose a domain model, then click either [Load Image] or paste in an image url and click [Analyze]

Domain Model: celebrities

https://oxfordportal.blob.core.windows.net/vision/Analysis/1-1.jpg

Analyzing Done

```
"celebrities": [
  {
    "name": "Avril Lavigne",
    "faceRectangle": {
      "left": 65,
      "top": 98,
      "width": 110,
      "height": 110
    },
    "confidence": 0.999769866
```



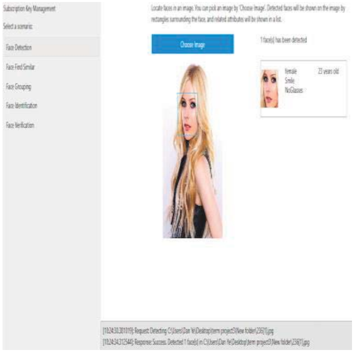
Describe an Image

Please click either [Load Image] or paste in an image url and click [Describe]

https://oxfordportal.blob.core.windows.net/vision/Analysis/2-1.jpg

Describing Done

```
[03:26:24.912410]: Describe Result:
[03:26:24.928047]: Image Format : Jpeg
[03:26:24.928047]: Image Dimensions : 360 x 500
[03:26:24.928047]: Description :
[03:26:24.928047]:    Caption : Avril Lavigne in a dress shirt and tie; Confidence : 0.246558666667553
[03:26:24.928047]:    Caption : Avril Lavigne wearing a dress shirt and tie; Confidence : 0.245558666667553
[03:26:24.928047]:    Caption : Avril Lavigne in a shirt and tie; Confidence : 0.162784610116166
[03:26:24.990567]:    Tags : person, clothing, outdoor, woman, young, holding, wearing, shirt, dress, yellow, top, water, smiling, bench, laptop, white, phone, table, cutting, standing, board, street
```

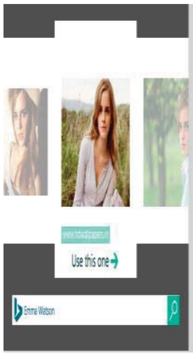


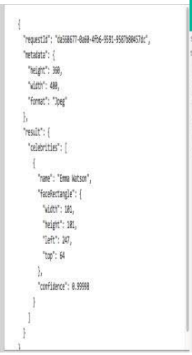
Locate faces in an image. You can pick an image by 'Choose Image'. Detected faces will be shown on the image by rectangles surrounding the face, and related attributes will be shown in a list.

1 face(s) has been detected

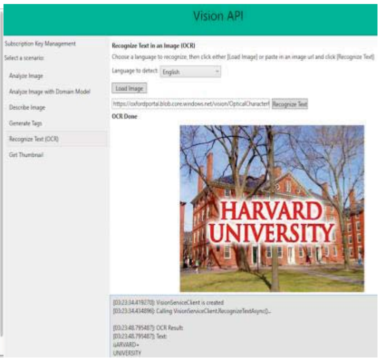
female 23 years old
Smile
NoGlasses

```
[10:24:30.301019]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
[10:24:34.312544]: Response: Success. Detected 1 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
```





```
{
  "requestId": "da568677-0a60-4fb6-9591-9587b80457dc",
  "metadata": {
    "height": 360,
    "width": 480,
    "format": "Jpeg"
  },
  "result": {
    "celebrities": [
      {
        "name": "Emma Watson",
        "faceRectangle": {
          "width": 101,
          "height": 101,
          "left": 247,
          "top": 64
        },
        "confidence": 0.99998
      }
    ]
  }
}
```



Recognize Text in an Image (OCR)

Choose a language to recognize, then click either [Load Image] or paste in an image url and click [Recognize Text]

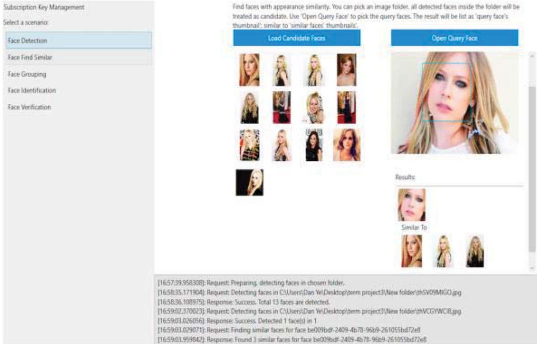
Language to detect: English

https://oxfordportal.blob.core.windows.net/vision/OpticalCharacterR

OCR Done

```
[03:23:34.419270]: VisionServiceClient is created
[03:23:34.434896]: Calling VisionServiceClient.RecognizeTextAsync()...

[03:23:48.795487]: OCR Result:
[03:23:48.795487]: Text:
iARVARD+
UNIVERSITY
```

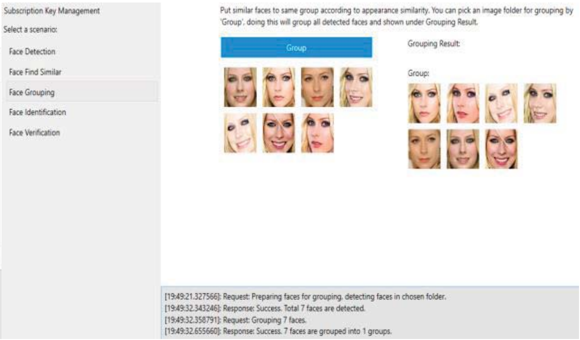


Find faces with appearance similarity. You can pick an image folder, all detected faces inside the folder will be treated as candidate. Use 'Open Query Face' to pick the query faces. The result will be list as 'query face's thumbnail'; similar to 'similar faces' thumbnails'.

Results:

Similar To

```
[16:57:39.958308]: Request: Preparing, detecting faces in chosen folder.
[16:58:35.171904]: Request: Detecting faces in C:\Users\Dan Ye\Desktop\term project3\New folder\thSV09MIGO.jpg
[16:58:36.108975]: Response: Success. Total 13 faces are detected.
[16:59:02.370023]: Request: Detecting faces in C:\Users\Dan Ye\Desktop\term project3\New folder\thVCGYWCI8.jpg
[16:59:03.026056]: Response: Success. Detected 1 face(s) in 1
[16:59:03.029071]: Request: Finding similar faces for face be009bdf-2409-4b78-96b9-261055bd72e8
[16:59:03.959842]: Response: Found 3 similar faces for face be009bdf-2409-4b78-96b9-261055bd72e8
```

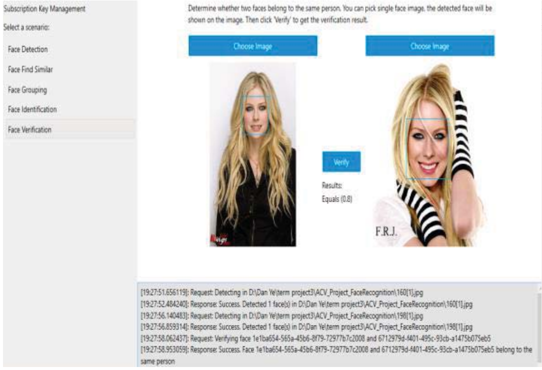


Put similar faces to same group according to appearance similarity. You can pick an image folder for grouping by 'Group', doing this will group all detected faces and shown under Grouping Result.

Grouping Result:

Group:

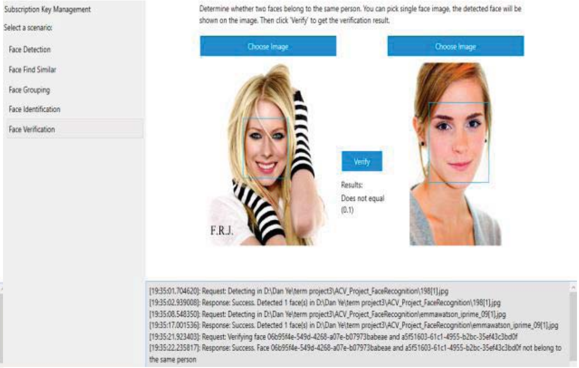
```
[19:49:21.327566]: Request: Preparing faces for grouping, detecting faces in chosen folder.
[19:49:32.343246]: Response: Success. Total 7 faces are detected.
[19:49:32.358791]: Request: Grouping 7 faces.
[19:49:32.655660]: Response: Success. 7 faces are grouped into 1 groups.
```



Determine whether two faces belong to the same person. You can pick single face image, the detected face will be shown on the image. Then click 'Verify' to get the verification result.

Results:
Equals (0.8)

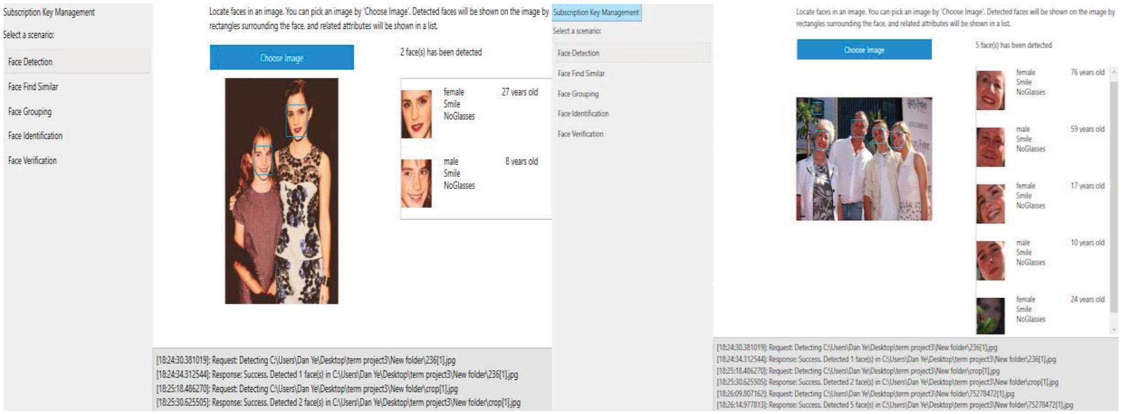
```
[19:27:51.656119]: Request: Detecting in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\160[1].jpg
[19:27:52.484240]: Response: Success. Detected 1 face(s) in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\160[1].jpg
[19:27:56.140483]: Request: Detecting in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\198[1].jpg
[19:27:56.859314]: Response: Success. Detected 1 face(s) in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\198[1].jpg
[19:27:58.062437]: Request: Verifying face 1e1ba654-565a-45b6-8f79-72977b7c2008 and 6712979d-f401-495c-93cb-a1475b075eb5
[19:27:58.953059]: Response: Success. Face 1e1ba654-565a-45b6-8f79-72977b7c2008 and 6712979d-f401-495c-93cb-a1475b075eb5 belong to the same person
```



Determine whether two faces belong to the same person. You can pick single face image, the detected face will be shown on the image. Then click 'Verify' to get the verification result.

Results:
Does not equal
(0.1)

```
[19:35:01.704620]: Request: Detecting in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\198[1].jpg
[19:35:02.939008]: Response: Success. Detected 1 face(s) in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\198[1].jpg
[19:35:08.548350]: Request: Detecting in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\emmawatson_iprime_09[1].jpg
[19:35:17.001536]: Response: Success. Detected 1 face(s) in D:\Dan Ye\term project3\ACV_Project_FaceRecognition\emmawatson_iprime_09[1].jpg
[19:35:21.923403]: Request: Verifying face 06b95f4e-549d-4268-a07e-b07973babeae and a5f51603-61c1-4955-b2bc-35ef43c3bd0f
[19:35:22.235817]: Response: Success. Face 06b95f4e-549d-4268-a07e-b07973babeae and a5f51603-61c1-4955-b2bc-35ef43c3bd0f not belong to the same person
```



Locate faces in an image. You can pick an image by 'Choose Image'. Detected faces will be shown on the image by rectangles surrounding the face, and related attributes will be shown in a list.

2 face(s) has been detected

female Smile NoGlasses 27 years old

male Smile NoGlasses 8 years old

```
[18:24:30.381019]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
[18:24:34.312544]: Response: Success. Detected 1 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
[18:25:18.486270]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\crop[1].jpg
[18:25:30.625505]: Response: Success. Detected 2 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\crop[1].jpg
```

Locate faces in an image. You can pick an image by 'Choose Image'. Detected faces will be shown on the image by rectangles surrounding the face, and related attributes will be shown in a list.

5 face(s) has been detected

female Smile NoGlasses 76 years old

male Smile NoGlasses 59 years old

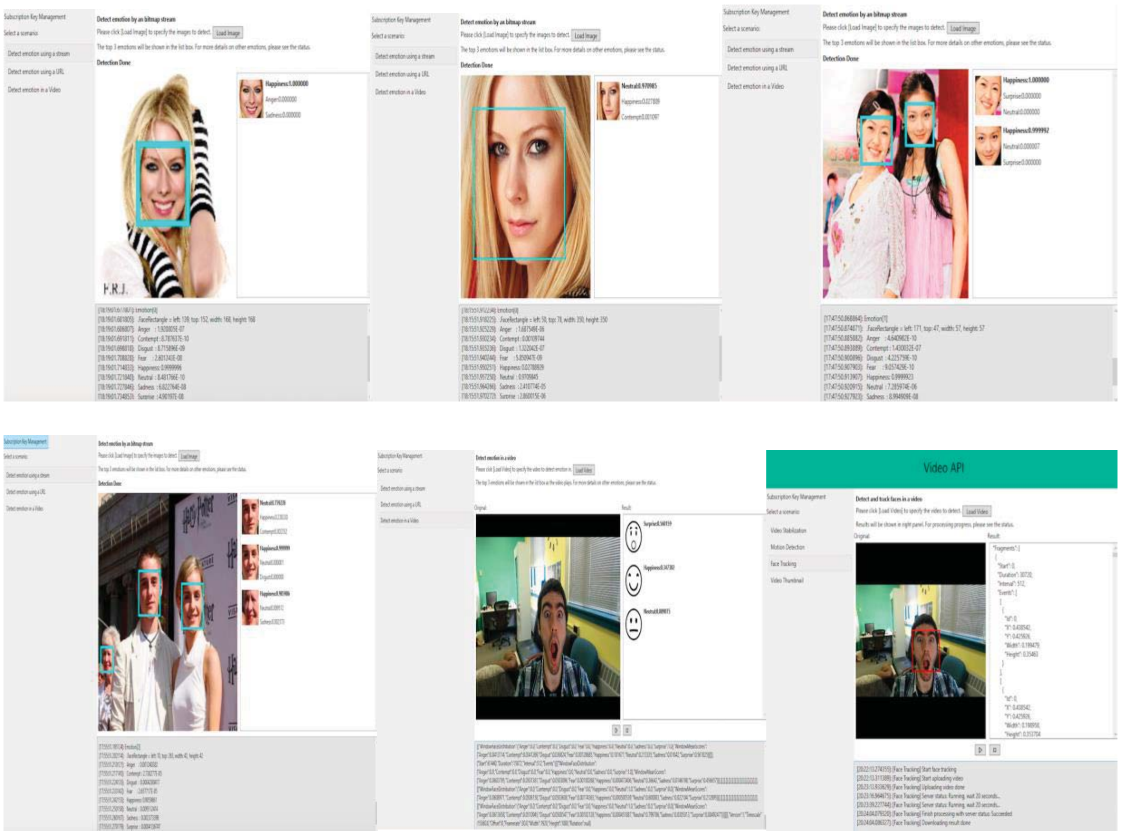
female Smile NoGlasses 17 years old

male Smile NoGlasses 10 years old

female Smile NoGlasses 24 years old

```
[18:24:30.381019]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
[18:24:34.312544]: Response: Success. Detected 1 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\236[1].jpg
[18:25:18.486270]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\crop[1].jpg
[18:25:30.625505]: Response: Success. Detected 2 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\crop[1].jpg
[18:26:09.807162]: Request: Detecting C:\Users\Dan Ye\Desktop\term project3\New folder\75278472[1].jpg
[18:26:14.977813]: Response: Success. Detected 5 face(s) in C:\Users\Dan Ye\Desktop\term project3\New folder\75278472[1].jpg
```

Fig.3. Emotion detection



Detect emotion by an bitmap stream

Please click [Load Image] to specify the images to detect.

The top 3 emotions will be shown in the list box. For more details on other emotions, please see the status.

Detection Done

Happiness:1.000000
Anger:0.000000
Sadness:0.000000

```
[18:19:01.6?1801]: Emotion[0]
[18:19:01.681805]:   .FaceRectangle = left: 139, top: 152, width: 168, height: 168
[18:19:01.686807]:   Anger    : 1.928805E-07
[18:19:01.691811]:   Contempt : 8.787637E-10
[18:19:01.698818]:   Disgust  : 8.715896E-09
[18:19:01.708828]:   Fear     : 2.801343E-08
[18:19:01.714833]:   Happiness: 0.9999996
[18:19:01.721840]:   Neutral  : 8.481766E-10
[18:19:01.727846]:   Sadness  : 6.822764E-08
[18:19:01.734850]:   Surprise : 4.90197E-08
```

Neutral:0.970985
Happiness:0.027889
Contempt:0.001097

```
[18:15:51.912234]: Emotion[0]
[18:15:51.918225]:   .FaceRectangle = left: 50, top: 78, width: 350, height: 350
[18:15:51.925229]:   Anger    : 1.687549E-06
[18:15:51.930234]:   Contempt : 0.00109744
[18:15:51.935236]:   Disgust  : 1.322042E-07
[18:15:51.940244]:   Fear     : 5.850947E-09
[18:15:51.950251]:   Happiness: 0.02788829
[18:15:51.957258]:   Neutral  : 0.9709845
[18:15:51.964266]:   Sadness  : 2.410774E-05
[18:15:51.970272]:   Surprise : 2.860015E-06
```

Happiness:1.000000
Surprise:0.000000
Neutral:0.000000

Happiness:0.999992
Neutral:0.000007
Surprise:0.000000

```
[17:47:50.868864]: Emotion[1]
[17:47:50.874071]:   .FaceRectangle = left: 171, top: 47, width: 57, height: 57
[17:47:50.885882]:   Anger    : 4.640982E-10
[17:47:50.893889]:   Contempt : 1.430032E-07
[17:47:50.900896]:   Disgust  : 4.225759E-10
[17:47:50.907903]:   Fear     : 9.057429E-10
[17:47:50.913907]:   Happiness: 0.9999923
[17:47:50.920915]:   Neutral  : 7.285974E-06
[17:47:50.927923]:   Sadness  : 8.994909E-08
```

Detect emotion in a video

Please click [Load Video] to specify the video to detect emotion in.

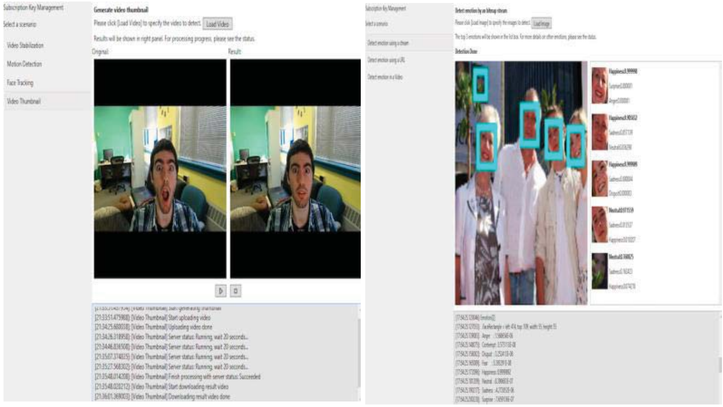
The top 3 emotions will be shown in the list box as the video plays. For more details on other emotions, please see the status.

Detect and track faces in a video

Please click [Load Video] to specify the video to detect.

Results will be shown in right panel. For processing progress, please see the status.

```
[20:23:13.833629]: [Face Tracking] Uploading video done
[20:23:16.964675]: [Face Tracking] Server status: Running, wait 20 seconds...
[20:23:39.227744]: [Face Tracking] Server status: Running, wait 20 seconds...
[20:24:04.079328]: [Face Tracking] Finish processing with server status: Succeeded
[20:24:04.086327]: [Face Tracking] Downloading result done
```


Subscription Key Management
Select a scenario:
Video Stabilization
Motion Detection
Face Tracking
Video Thumbnail
Load Video
Load Image
Detect emotion using a stream
Detect emotion using a URL
Detect emotion in a Video

Generate video thumbnail

Please click [Load Video] to specify the video to detect.

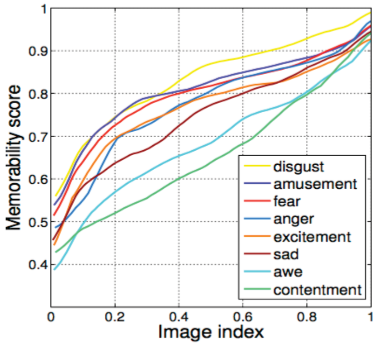
Results will be shown in right panel. For processing progress, please see the status.

Detect emotion by an bitmap stream

Please click [Load Image] to specify the images to detect.

Detection Done

Fig.4. Memorability vs emotions



SESSION

PATTERN RECOGNITION: OBJECTS AND BEHAVIORAL + APPLICATIONS

Chair(s)

TBA

Simulation-based Visual Analysis of Individual and Group Dynamic Behavior

Pawel Gasiorowski [1]**, Vassil Vassilev**[2] **and Karim Ouazzane**[2]
[1]The Vinyl Factory, London, UK
[2]School of Computing, London Metropolitan University, London, UK

Abstract *The article presents a new framework for individual and group dynamic behavior analysis with wide applicability to video surveillance and security, accidents and safety management, customer insight and computer games. It combines graphical multi-agent simulation and motion pattern recognition for performing visual data analysis using an object-centric approach. The article describes the simulation model used for modeling the individual and group dynamics which is based on the analytical description of dynamic trajectories in closed micro-worlds and the individual and group behavior patterns exhibited by the agents in the visual scene. The simulator is implemented using 3D graphics tools and supports real-time event log analysis for pattern recognition and classification of the individual and group agent's behavior.*

Keywords: Visual analytics, Behavior pattern recognition, Individual and group dynamics, Agent-based simulation, Ghosting, Ray casting

1 Introduction

Behaviour analysis of individual and group dynamics in closed micro-worlds is an area of extensive research in both academia and industry due to its wide applicability to various areas - video surveillance and security, accidents and safety management, business customer insight and video games programming. Despite the recent advances in the use of various methods for visual behavior data analysis (i.e., Markov models, statistical pattern recognition, qualitative physics, etc. - see [2-5] for some recent research in analysis of individual dynamics and [6-8] in group dynamics) and the availability of some powerful tools for video data analysis in the market such as 3VR Video Intelligence Platform, savVI Real-Time Event Detection, PureTechSystems Video Analytics, IndigoVision Advanced Analytics, IBM Intelligent Video Analytics, etc. (see [23-27] for more details on these products) the problem remains difficult. Two main factors impact the real-time performance here: the enormous volume of data, which has to be processed in real-time, and the need to combine video data processing with complex analytical symbolic data processing. While the first problem can be addressed entirely by the technological development, the second one requires model-driven behavior pattern analysis which cannot be implemented by the visual data processing methods alone.

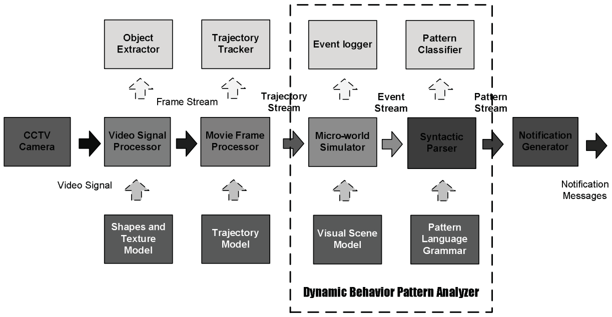


Fig. 1 General workflow of the framework

We are specifically interested in analyzing the dynamic behavior of individuals and groups of individuals moving at a walking speed within enclosed spaces (rooms, corridors, staircases, floors and open spaces) of big buildings, such as shopping malls and transport stations, as well as in large transport vehicles, such as cruiser ships. Our approach for tackling the complexity of this task is to eliminate the need for analyzing the entire video stream and replace it with the analysis of simulated data which approximates the actual video stream. The framework presented here combines two complementary methods for data analytics: visual trajectory analysis based on 3D simulation, and dynamic pattern classification based on agent's behavior logic. It forms a central part of the research program within the Cyber Security Research Group of London Metropolitan University which is dedicated to machine processing of video surveillance data in real time. It includes visual scene extraction, trajectory reconstruction, dynamic simulation and behaviour analysis for online processing of live video signals from CCTV cameras (see Fig. 1). This article reports the core of the framework, the simulation and behaviour analysis platform which has been implemented in Java using jMonkey engine [14].

2 Visual simulation as a basis for behaviour analysis

In their comprehensive book Xiang and Gong [1] classify the approaches for the development of behavior's representation model into four groups: *object-based*, *part-based*, *pixel-based* and *event based*. In this part of the research we combine the object-based approach with the event-driven approach, which allows us to streamline the video data processing from the physical camera input to the logical notification output in real-time. Our approach to dynamic behavior analysis fits within the tradition of agent-based simulations [9-12] which is widely used in game programming. The starting point of the simulation is the reconstructed trajectories of individual objects in the visual scene [16]. As a result of the simulation the annotated live video signal is enhanced with additional information which is used for dynamic behavior analysis in accordance with the behavior pattern description. The output is an asynchronous notification corresponding to the identified pattern – i.e., calling the fire brigade, calling the ambulance, calling the police or the bomb squad.

2.1 The trajectory data

There are three separate types of input data used by the simulator – static visual scene information, dynamic trajectories of the moving objects and asynchronous event notifications. The simulator performs initial setup of the visual scene which can be updated later in the case of synchronous or asynchronous changes in the scene (e.g. appearance of a new agent as a result of moving inside the scope of the camera, appearance of a new object as a result of changing the viewing angle, disappearance of an objects from the visual scene, changing the viewing angle, receiving an additional signal, etc.). A sample XML of the visual scene is shown below:

```
<scene id="vc#00000BF>
  <camera id="cctv#0000XX">
    <frustrum>
      <viewAngle>
        <quaternionW>0.1739063</quaternionW>
        <quaternionX>-0.7228111</quaternionX>
        <quaternionY>0.19664077</quaternionY>
        <quaternionZ>0.63924426</quaternionZ>
      </viewAngle>
      <viewDirection>
        <vectorX>-0.7644838</vectorX>
        <vectorY>-0.6443617</vectorY>
        <vectorZ>-0.019037962</vectorZ>
      </viewDirection>
      <aspectRatio>
        <width>1280</width>
        <height>1024</height>
      </aspectRatio>
    </frustrum>
  </camera>
  <objects total="8">
    <object id="obj#00001 type="dynamic"
                          shape="humanoid">
      <objectsFeatures>
        <proximityTrigger>3.00f</proximityTrigger>
        <sightRangeTrigger>10.0f</sightRangeTrigger>
        <velocity>2.25f</velocity>
        ...
      </objectsFeatures>
    </object>
    ...
  </objects>
</scene>
```

Between the changes in the visual scene the simulation is driven entirely by data from the reconstructed trajectories of individual objects [16]. The trajectories are described analytically in a standard vector notation for representing location and motion. This description is based on the quaternions theory instead of purely trigonometric equations, in order to represent more complex movements involving rotation, changing the direction and twisting while moving [13]. While the trajectories provide information about the location and movements of individual agents only, the simulation generates additional information which allow the analysis of the dynamic group behavior as well. In our experiments the trajectories are simulated but the actual information will be provided by the module responsible for reconstructing the trajectories, currently under development.

2.2 The simulation entities

In the simulation we adopt an agent approach similar to the toy world which is widely used in AI for controlling intelligent robots and is also endorsed by the 3D games programming community [13]. The main entities used to build the simulation are:

Agent: an abstraction of humans or any other entities capable of some sort of movements (i.e., shopper in the shopping mall, passenger in a vehicle, traveler at the station, etc.). Their behavior is essentially either dynamic or active but always autonomous.

Pair: two agents involved in dynamic interaction (i.e., handshaking, hugging, pushing, punching or kicking each other)

Group: several agents sharing some common behaviors, allowing to treat them as one entity (i.e. flow of people moving into the direction of an open door, people climbing the same stairs, people walking in the same room, etc.). The groups exhibit both external dynamic (relative to the scene) and internal dynamic (relative to the included individual agents).

Object: an object that is part of the scene and with which the agents can interact during their physical movements (i.e., doors, stairs, floor, shelves, etc.). Typically they do not change their relative position within the scene and remain static for a period of time.

Scene: well-defined boundaries where each agent can move (i.e. a room in a building or a compartment in a transport vehicle). It provides the basis for coordinates of the restricted micro-world observed by a video camera.

The dynamics of the scene is analyzed through recognition and classification of various *events*, *activities* and *situations* which are observed within the visual scene. They correspond to the real-life dynamics observed by the CCTV cameras.

2.3 Changing the location

The key aspect of online simulation is the execution of agent's trajectories in real time. However, depending on the purpose of the simulation, the trajectory information does not have to come from a camera. With the possibility to use gravity and incoming data on agent's movements arriving at a constant rate, it is possible to calculate the movements of an agent with a relatively high precision, absolutely sufficient for visual analysis of the dynamic behavior. Calculating the positions in the next frame, when moving horizontally along a straight line, is implemented easily using vector calculus. The position is determined on the basis of the current location, the relative velocity of movement and the forward direction vector [14]. But when the agent moves on a curved path, its position needs to be calculated through combining the motion formula with some kind of rotation. Our algorithm is inspired by the ideas of Reynolds [19], which are especially appropriate for real-time simulations due to the fact that the speed of movement is not relative to the visual frames and thus, it does not depend on the speed of the simulation. The new position is calculated using quaternions, while the actual position of the agent is used only for smoothening of the trajectory. An example of such a curved trajectory is shown in Fig. 2. The calculations in this case are relatively simple and can be done in real-time.

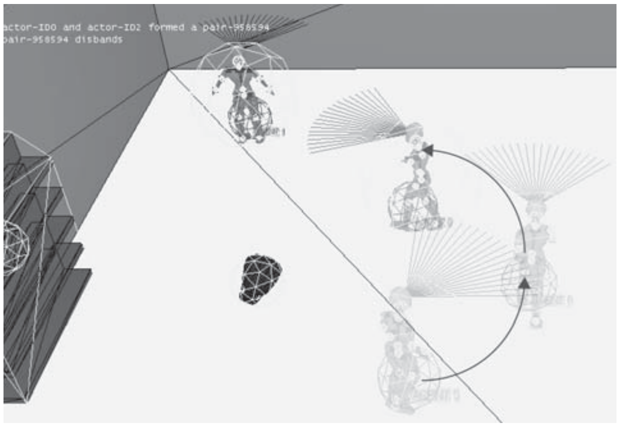


Fig. 2. An agent moving along curved path.

2.4 Gathering of agents and grouping

Unlike the statistical approach to simulation used in crowd behavior, which performs well on a macro level but do not give much on a micro level [1,6-8], we base the behavior analysis on the individual agent's behavior. It still allows group behavior analysis for small groups in enclosed spaces, or "mini-crowds", which is within our scope of interest.

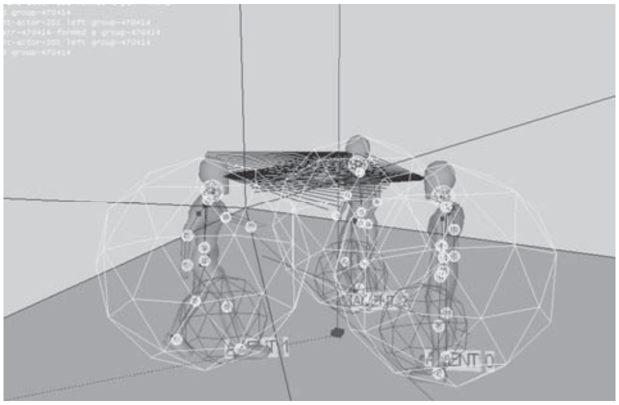

group-470414
t-actor-ID1 left group-470414
air-470414 formed a group-470414
t-actor-ID1 left group-470414
group-470414
AGENT 1
AGENT 0

Fig. 3. Gathering of multiple agents in the visual scene

In our approach the individual and group dynamics are linked logically and not statistically. This is a critical feature of our approach to group behavior analysis since our main aim is to analyze the individual and group dynamics from the point of view of the individual interactions and interrelations between the agents in the scene. It is essential, for example, to be able to establish when a group is being formed but to continue tracking both the individual members of the group as well as the group as a whole, because the individual behavior of the agents within the group are superimposed. Fig. 3 shows a slightly more "crowded" scene with a number of agents wandering around while being in a group. It is also important to be able to analyze the group dynamics in relation to other groups which may exist in the same scene. In our case this is possible thanks to the logical approach adopted to grouping.

3 Events on the visual scene

The simulation plays a dual role in our framework. On one hand, it is used for formation of the dynamic patterns of behavior. On the other hand, it allows generating additional information relevant to the agent's behavior which is based on the laws of physics and the logics of the visual scene.

During the construction of our simulator we have incorporated a number of techniques widely used in game programming [17] and robot control systems [19]. The most important of them are the invisible bounding box volumes surrounding the agents and the ray casting [21]. The API of jMonkey we used for development utilizes these concepts in the form of listeners known as "ghosts" and "Line-of-Sights" leads (LOS). The "ghosts" in combination with LOS can be used for further enhancement of the control over agent's dynamic behavior:

- to estimate the spatial dimensions of entities within their existing space.
- to extrapolate the trajectories beyond the scene of visibility.
- to calculate the distances between objects on the path of movement or on the line of sight.
- to induce logically new relations between objects, like detecting obstacles in front/sides of the agents, preventing collisions with objects and predicting reactions.

3.1 Identification of the physical space occupied by the objects using "ghosts"

Physical entities within the focus of the camera occupy certain space and their "bounding boxes" are the starting point of the model dynamics built into our framework. The bounding boxes in the visual scene represent objects that have been successfully recognized and delivered as an input to the simulator. With the knowledge of physical boundaries of the objects, we start outlining a set of rules for the event logging strategy such as taking into account the relationships between objects based on proximity values and overlapping of their mutual spaces. These volumes or "ghosts" have been highlighted with yellow wireframes in the simulation to visually evaluate their accuracy at runtime as it is shown in Fig. 4.

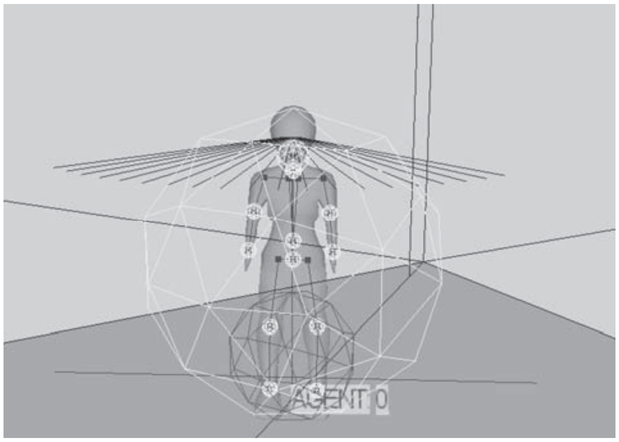


Fig. 4. Ghosts attached to an autonomous agent and its limbs.

Using bounding boxes has the advantage; the boxing does not cause any slowdowns of the simulation since the boxes are not participating in any physical collision calculations. Any recognized object that is passed to the simulator can be equipped with its own "ghost" to support better logging, but the obvious drawback is in the limited area of coverage. This problem is addressed by another technique in 3D graphics programming known as *ray casting*.

3.2 Estimating the physical dimensions using ray casting techniques

The ray casting is a technique that is based on the idea of casting a ray from one point in a specified direction and checking if any geometry comes into contact with it. This will enable us to establish the existence of geometries in a particular area of the visual space (Fig.5). This method is also often called the Line-of-Sight (LOS) and it determines whether two geometries in the environment can "see" each other with respect to another that can cause an occlusion [20]. In our simulator the ray casting technique has twofold usage - firstly it allows us to equip each agent with a "sight sense" and secondly, it enhances the event logging by scanning each agent's surroundings.

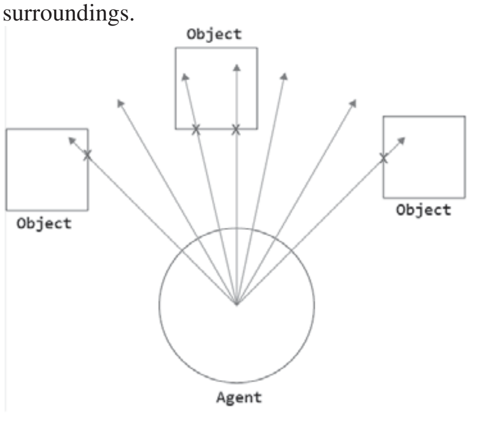
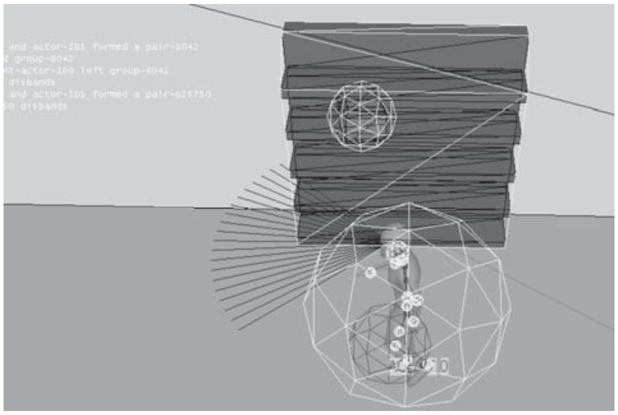


Fig. 5. Ray casting by an agent relative to movement direction

3.3 Detecting obstacles by "sighting" the agent

The full sight sense of an agent has been developed with the use of several rays casted from head position over an arc. The main difference in our approach compared to the case of a single ray casting used in robot motion control [18] lies in the positioning of rays. Each line is being re-rendered with a ¼ π * 0.1 angular offset from the previous one at each frame of the simulation. Because LOS technique has been applied to every agent, it is possible to determine any obstacle that is exactly in front of it. The way we have implemented it approximates the human perceptions, accounting the rules of peripheral vision so that it is possible to deduce agent's focus at a specific time. However, this imposes certain limitations on the way the information about the nearby environment is being gathered since all "sight" rays are being casted towards similar directions, covering a limited front area of the agent only as illustrated in Fig. 6.



```
 and actor-I01 formed a pair-6042
d group-6042
nt-actor-I00 left group-6042
 disbands
 and actor-I01 formed a pair-615750
50 disbands
```

Fig. 6. The sight sense in a form of several rays casted in a viewing direction is insufficient to detect side objects

To eliminate the above limitation, we have to log additional information obtained from other source of probing. In our implementation we have attached special "ghosts" not only to the agents, but also to any object which has been recognized on the visual scene. The difference between the two types of ghosts is that unlike the agent's ghosts, the object ghosts do not cast rays relative to the direction of movement but "reflect" rays along their surface. This allows the agents to move within close proximity without "touching" the objects.

3.4 Establishing relations between agents and the surrounding using ray casting

The main reason for the introduction of ray casting in the simulation is to capture the physical placement of the objects in relation to agent's location by collecting data on entities that come into contact with rays. Through knowing the actor's forward viewing vector, it is possible to calculate its left and right directions, cast rays and gather information on the static entities that are on a side as shown in Fig. 7**.**

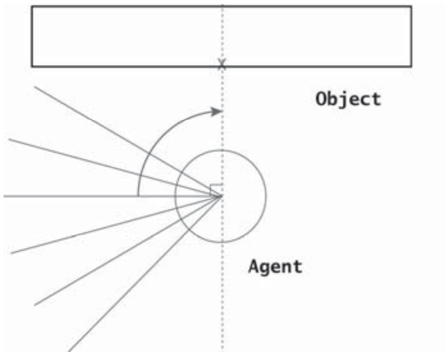
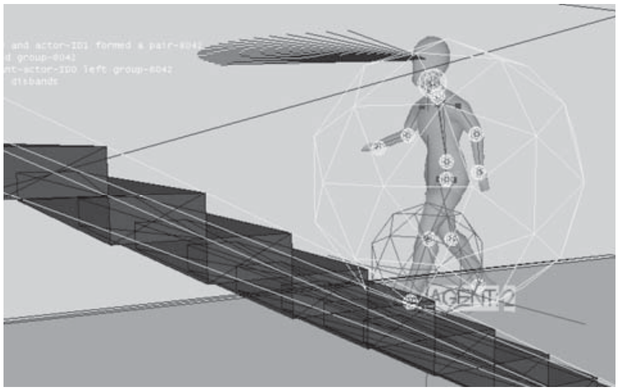


Fig. 7. Rays being casted on each of the agent's sides allow to detect any previously recognized static objects

This simple procedure executed at specific time intervals allows us to store the data in a data structure, sample it separately and potentially report it in the log. By developing this idea further, it becomes possible to recognize when an autonomous entity finds itself "on top of" or "below" a static object. In this case, instead of casting the rays along one axis only, we have to do this along two axes as depicted in Fig.8.



```
 and actor-ID1 formed a pair-8042
d group-8042
nt-actor-ID0 left group-8042
 disbands
```

Fig. 8. Casting rays along axes relative to viewing direction to detect changing altitude

We have implemented a number of different single and multi-ray casting algorithms as part of the simulator [14]. At a later stage we plan to analyse the dependence of the rays density from the complexity of the scene in order to estimate the computational power required for the simulation.

4 Social life of the agents

An essential advantage of the agent-based simulation is that it allows analyzing the group behavior using the same mechanisms used to analyze the individual behavior. Our simulator is capable of capturing the agent's "social life", which opens an unlimited opportunity for digging further into the group behavior analysis.

4.1 "Attraction" between agents and coupling

Using the simple concepts of *coupling* and *grouping* of the agents it is possible to develop a sound foundation for analyzing group dynamics in a purely object-oriented manner using efficient algorithms. The fundamental mechanism for simulating group behavior is based on the concept of "attraction" between agents. When two agents detect each other they may become "attracted" and form a pair (Fig. 9)**.**

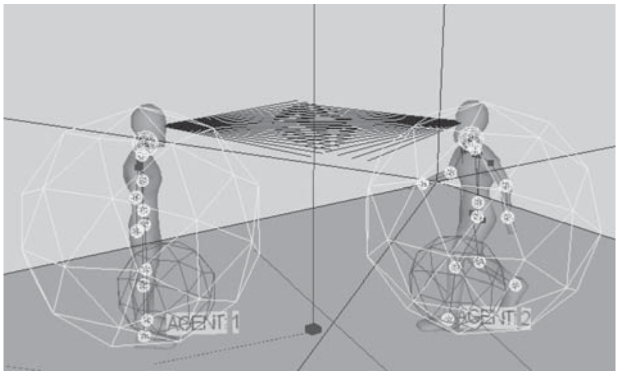


Fig. 9. The moment of "attraction" between agents

This can be used to analyze the group behavior. The main task is to formulate computationally tractable criteria of attraction. In the current version of our framework we are considering the distance between the agents only – the attraction is maximal if the agent's capsules intersect and decreases with the distance between them, which can be easily detected by the agent's ghosts (Fig. 10). In future research we intend to account more complex criteria of "attraction" which include other factors of interest such as direction of movement, as well as non-dynamic factors, such as behavioral attitude.

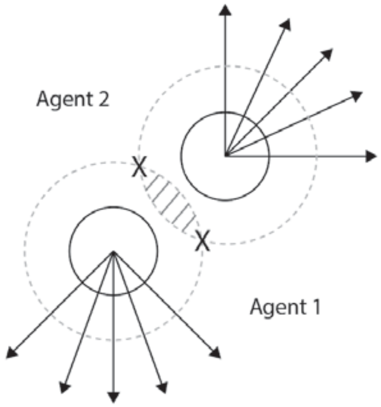


Fig. 10. Detecting coupling through ghost interaction

4.2 Group formation

Our approach to group formation is based on measuring the distances between agents to establish if they are in close proximity. For this purpose, we are calculating a median point out of the physical locations. The reference point allows treating the group as a single entity and by superposition of the individual behavior it is possible to establish a group behavior. It is important to stress that tracking group behavior does not seize tracking the individual activity, so that it is still possible to identify the individual activities in parallel.

4.3 Joining and leaving the crowd

The agents may join or leave already established groups (Fig. 11) which can be formed by gathering individual agents, by merging pairs or by joining a pair of agents by an individual agent. In the current version of the simulator the grouping override the coupling, i.e, the couples are treated as separate individuals within the group and they leave the groups individually, not pairwise. The current version of the simulator assumes that for an agent to join a group, he must first find himself within proximity to a member of an existing group. If, while coupling it is realized that the other agent already belongs to a group, the first agent joins them, but if he himself belongs to a group, the two groups merge, forming a "crowd". Analogically, to state that an agent is leaving the group, the distance from the other members of the group needs to cross certain threshold which is a parameter of the simulation.

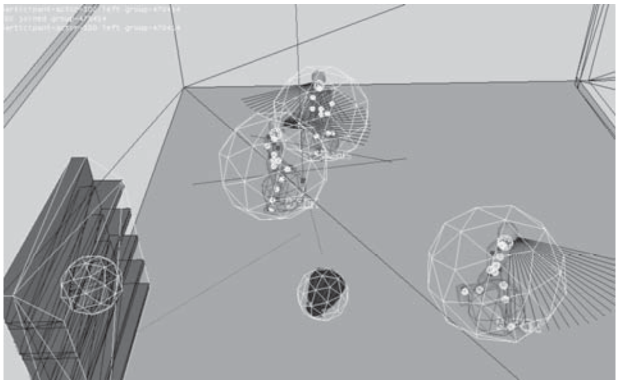


Fig. 11. Joining and leaving a group by individual agents

In our model the groups consist of 3 or more agents and can be formed by gathering of individual agents, by merging pairs or by joining a pair by an individual agent. In the current version of the simulator the grouping override the coupling, i.e, the couples are treated as separate individuals within the group and they leave the groups individually, not pairwise.

5 Implementation

The 3D simulator is written entirely in Java and uses the open source engine jMonkey [14]. This software is widely used in game programming and has been chosen for implementing the platform because it allows modeling of the physical constraints of the micro-world such as gravity and supports additional control mechanisms. We have utilized it for detecting obstacles of the agent's path, collision prevention and navigation control.

5.1 Agents

The model of a humanoid agent was developed using Blender open source modeler [15]. Each agent has an associated "ghost" with it which is equipped with ray casting algorithm for probing the space in order to detect obstacles, navigate through the space and interact with the environment and other agents. Although the agent model allows the use of the full body armature, in the current version of the simulator the gestures are not accounted. However, this will be exploited further in the next version of the simulator.

5.2 Movements

The physical movements of the agents within the visual scene are modelled using methods of standard vector algebra with the addition of quaternions theory to model rotations and movements along curved trajectories. Currently, the movements are calculated on the basis of the position, the direction of movement and the velocity. Although this method is approximate, due to the frequent recalculations, the deviation from the actual trajectories is insignificant for the purpose of the pattern matching and does not affect the quality of the analysis. This allows the analysis to be performed in real-time using entirely simulated data rather than using actual data from the video stream.

5.3 Simulator loop

The event logger works in a loop. At the beginning of each iteration, it updates the current state of the visual scene and then logs all events generated by the individual observers during the simulation. The loop is initialized when a new configuration is introduced as a result of an internal or external asynchronous event in the visual scene.

5.4 Scene changes

At each update the simulator records potential collisions caused by ghosts overlapping or rays piercing physical geometries of the objects. Some of the calculations that are needed for this update can be appropriately timed to reduce the potential frame rate drops and to ensure the data is not coming in too fast to be synchronized. During the experiments it was observed that the delay is not causing any major frame rate drops, but with the increase of the number of objects and "overcrowding" of the scene more substantial computational power may be required to keep the frame drop rate low.

5.5 Event logging

The major role of the simulation is to generate an informative log of the events occurring within the visual scene so that they can be analysed further by pattern matching techniques. In its current version, the simulator generates a log file with time-stamped entries describing each captured event:

```
...
09:39:41 :: Agent ID0 LeftLowArm touches Stairs ID1
09:39:41 :: Agent ID0 LeftHand touches Stairs ID1
09:39:41 :: Agent ID0 RightFoot touches Stairs ID1
09:39:41 :: Agent ID0 LeftFoot touches Stairs ID1
09:39:51 :: Agent ID2 moves towards Bookshelf ID2
09:39:53 :: Agent ID2 moves towards Bookshelf ID2
09:39:56 :: Agent ID2 moves along Bookshelf ID2 on left
09:39:56 :: Agent ID2 moves away from Bookshelf ID2
09:39:57 :: Agent ID2 moves towards Bookshelf ID2
09:39:59 :: Agent ID2 moves along Bookshelf ID2 on right
09:40:41 :: Agent ID2 climbs Stairs ID1 up
...
```

The event logger is implemented with architecture of an "observer", attaching a separate individual logger to each object within the visual scene. The individual observers log all events related to the observed. This allows further extension of the logging module without changing the existing code of the simulator. In the next version of our simulator we plan to incorporate fine grained event logging which account not only for the body motion of the agents but also for their gestures.

5.6 Pattern classification and beyond

The simulator log is parsed for recognizing and classification of the behaviour patterns according to the grammar of its language [22]. After the simulator the behaviour analysis can be continued solely based on the logs, while the original video data can be used to increase the precision of approximation. This approach gives the opportunity to incorporate purely symbolic techniques for behaviour analysis. In a forthcoming article we will report the visual dynamics ontology developed for this purpose. It forms another part of our research program which will be based entirely on semantic technologies.

6 Conclusion

This article introduces a new framework for real-time video data processing for the purpose of individual and group dynamic behaviour analysis based on 3D simulation and dynamic pattern classification. Our approach combines methods from games programming and robotics. The main advantage of this approach is that it allows the analysis of both individual and group dynamics in a single unified manner at different level of granularity depending on the needs. Although the framework is still under development, its core component - the simulator - is already completed and the experimental tests with simulated data in real time look very promising. The pattern classifier, which processes the event log generated during the simulation, for further analysis of the visual scene, is currently under development and will be reported separately in a forthcoming publication. It performs real time parsing according to the grammar of the event logging language and its input is the basis for the development of a suitable notification mechanisms. We plan to extend the language in order to represent more fine grained patterns of behaviour which go beyond the dynamics of pure body motion and include gestures as well.

The work reported here is conducted as part of the PhD research of the first author at London Metropolitan University and is sponsored by The Vinyl Factory of London, UK.

References

[1] S. Gong, T. Xiang, Visual analysis of behaviour from pixels to semantics. London: Springer, 2011.

[2] C. Hu, S. Wo, An efficient method of human behavior recognition in smart environments, in: Int. Conf. on Computer Application and System Modeling (ICCASM), Vol. 12, pp. 690–693, 2010.

[3] K. Yordanova, Modelling Human Behaviour Using Partial Order Planning Based on Atomic Action Templates, in: 7th Int. Conf. on Intelligent Environments (IE), pp. 338–341, 2011.

[4] C. Wang, F. Wang, A Knowledge-Based Strategy for Object Recognition and Reconstruction, in: Int. Conf. on Information Technology and Computer Science (ITCS), pp. 387–391, 2009.

[5] M. Attamimi, T. Nakamura, T. Nagai, Hierarchical multilevel object recognition using Markov model, in: 21st Int. Conf. on Pattern Recognition (ICPR), pp. 2963–2966, 2012.

[6] S. Wu, B. Moore, M. Shah, Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition CVPR2010, pp. 2054–2060, 2010.

[7] P. Saboia, S. Goldenstein, Crowd Simulation: Improving Pedestrians' Dynamics by the Application of Lattice-Gas Concepts to the Social Force Model, in: 24th SIBGRAPI Conf. on Graphics, Patterns and Images (Sibgrapi), pp. 41–47, 2011.

[8] R. Guo, H. Huang, A mobile lattice gas model for simulating pedestrian evacuation, in: Physica, Part A: Statistical Mechanics and its Applications, Vol. 387, pp. 580–586, 2007.

[9] L. Hluchy, M. Kvassay, S. Dlugolinský, B. Schneider et al., Handling internal complexity in highly realistic agent-based models of human behaviour, in: 6th IEEE Int. Symp. on Applied Computational Intelligence and Informatics (SACI), pp. 11–16, 2011.

[10] A. Varas, M. Cornejo, D. Mainemer, B., Toledo et al., Cellular automaton model for evacuation process with obstacles, in: Physica A: Statistical Mechanics and its Applications, Vol. 382, pp. 631–642, 2007.

[11] X. Ben, X. Huang, Z. Zhuang, R. Yan, S. Xu, Agent-based approach for crowded pedestrian evacuation simulation, IET Intelligent Transport Systems, Vol. 7, pp. 56–67, 2011.

[12] S. Sharma, S. Lohgaonkar, Simulation of agent behavior in a goal finding application, in: IEEE Southeast Conf. (SECON), pp. 424–427, 2010.

[13] E. Lengyel, Mathematics for 3D Game Programming and Computer Graphics, 2nd ed., Hingham, MA: Charles River Media, 2003.

[14] R. Eden, JMonkeyEngine 3.0 Cookbook, Birmingham: Packt Publ., 2014.

[15] G. Fisher, Blender 3D Basics, 2nd ed., Birmingham: Packt Publ., 2014.

[16] A. Bogdanovych, M. Bauer, S. Simoff, Recognizing Customers' Mood in 3D Shopping Malls Based on the Trajectories of Their Avatars, in: Filipe, J., Cordeiro, J. (Eds.), Enterprise Information Systems, LNBIP, Berlin: Springer, pp. 745–757, 2009.

[17] M. Wang, H. Lu, Research on Algorithms of Intelligent 3D Path Finding in Game Development, in: Int. Conf. on Industrial Control and Electronics Engineering (ICICEE), pp. 1738–1742, 2012.

[18] T. Terzimehic, S. Silajdzic, V. Vajnberger et al., Path finding simulator for mobile robot navigation, in: XXIII Int. Symp.on Information, Communication and Automation Technologies (ICAT), pp. 1–6, 2011.

[19] A. Croitoru, A., Deriving Low-Level Steering Behaviors from Trajectory Data, in: Proc. IEEE Int. Conf. on Data Mining Workshops (ICDMW), pp. 583–590, 2009.

[20] B. Salomon, N. Govindaraju, A. Sud, R. Gayle, M. Lin, D. Manocha, Accelerating Line of Sight Computations Using Graphics Processing Units, in: Proc. 24th Army Science Conference, 2004.

[21] J. Beaudoin, J. Hughes Clarke, J. Bartlett, Application of Surface Sound Speed Measurements in Post-Processing for Multi-Sector Multibeam Echosounders, International Hydrographic Review, Vol. 5, No. 3, pp. 26-31, 2004.

[22] D. Grune, C. J. H. Jacobs, Parsing Techniques: A Practical Guide, 2nd ed., NY: Springer, 2008.

[23] 3VRVideoIntelligence Platform [http://3vr.com/products/videoanalytics last visited: 31-05-2016]

[24] savVi Real-Time Event Detection [http://www.agentvi.com/61-Products-282-savVi_Real_Time_Event_Detection; last visited: 31-05-2016]

[25] PureTech Systems VideoAnalytics [http://www.puretechsystems.com/video-analytics.html; last visited: 31-05-2016]

[26] Indigo Vision Control Center [https://www.indigovision.com/en-us/products/management-software/control-center; last visited: 31-05-2016]

[27] IBM Intelligent Video Analytics [http://www.ibm-3.com/software/products /en/intelligent-video-analytics; last visited: 31-05--05-2016]

Forecasting of severe Thunderstorms using K- nearest neighbor technique

Himadri Chakrabarty (Bhattacharyya)[1], Sonia Bhattacharya[2]
[1]Department of Computer Science, Surendranath Collage, Calcutta University, Kolkata, India
[2] Department of Computer Science, West Bengal State University, Kolkata, West Bengal, India

AbstractT - *K-nn is one of the popular techniques in the field of pattern recognition. Here it has been applied for the prediction of severe thunderstorms. Three types of weather parameters i. e., moisture difference, adiabatic lapse rate and wind shear at different geopotential heights of the upper atmosphere have been taken into account for this job. Applying K-nn methodology we get more than 98% correct prediction for 'squall days' and 90% correct prediction for 'no squall days' with 12 hours lead time. Both surface as well as upper air data which are measured by radiosonde/ rawindsonde in the early morning are used in this case.*

Key Words: Severe thunderstorms, Pattern recognition, K-NN methodology

INTRODUCTION

Severe thunderstorms produce large hail, damaging wind, very heavy rainfall and tornadoes. Thunderstorms disrupt human life in more than one way. The felling of millions of trees, deaths due to lightning hazard and wind shear are just some of the dissipation manifestations.

The strong surface wind is known as squall, which is a sudden and sharp increase in the wind speed of 45 kilometers or more per hour during a short time interval of minimum 1 minute, [1]. The typical thunderstorm is 15 miles in diameter and lasts an average of 30 minutes [2]. Substantial research work was carried out in the last two decades about the understanding of the life cycle of thunderstorm. Several climatic parameters play roles to generate squall-storms [3]. There occur severe thunder-squalls and hailstorms in India, including Nor'wester of North-East India, Bangladesh and Assam in the pre-monsoon period, [4]. Accurate prediction of such severe weather feature is very difficult task due to the dynamic nature of atmosphere, [5]. In India, the coastal areas (like the west coast, Orissa, Andhra Pradesh and West Bengal) as well as north east have been affected many times by thunderstorm and heavy rainfall [6]. Chakrabarty1 et al predicted the 'occurrence' and 'no occurrence' of squall-storms in their previous work in 2015 using the weather data of moisture difference, dry adiabatic lapse rate and vertical wind shear. These weather data were sensed by radiosonde and measured at different geopotential heights of the upper atmosphere. There were thirteen input variables, out of first five belongs to moisture difference, next five belongs to adiabatic lapse rate and remaining three belongs to vertical wind shear. All these thirteen RSRW input data recorded at 6.00 a.m. local time (00.00UTC) in Kolkata (22.3^{o}N/88.3^{o}E), India. They got 91% correct prediction for squall days and 87.63% for no squall days using K-nearest neighbor technique. In this present paper research has been conducted using same data set and same technique. The main difference with the previous job is that in the previous paper K is chosen as 35 whereas in the current paper K is chosen as 1, 3, and 5 respectively. Here we get more than 98% correct prediction for 'squall days' and 80% correct prediction for 'no squall days'. The best choice of K depends upon the data; generally, larger values of K reduce the effect of noise on the classification, [7] but make boundaries between classes less distinct. If K = 1, then the object is simply assigned to the class of its nearest neighbor. Brath et al. [8] and Jayawardena et al. [9] applied K-NN method for flood forecasting. Jan et al. [14] used data mining technique for the seasonal to Inter-Annual Climate prediction. Here in this paper, K-NN technique has been applied on the weather data to forecast the 'occurrence'/ 'no occurrence' of squall-storm having around 12 hours lead time.

2 DATA

2.1 DATA COLLECTION

All the weather data were procured from India Meteorological Department, Govt. of India during the period of 18 years from 1990 to 2008 for the months of March-April-May. The data were recorded at 06.00 a.m. local time (00:00 UTC) by radiosonde and rawindsonde over Kolkata, North-East India. Here data have been considered both for 'squall' and 'no-squall' days. The numbers of 'squall-storm' days are 69 and 'no squall-storm' days are 315.

2.2 DATA DESCRIPTION

Here thirteen weather variables have been considered for the prediction. Out of which x1, x2, x3, x4 and x5 indicate vertical moisture difference profile at surface level, and at 1000hpa (approximately 75 meters), 850hpa (approximately 1500 meters), 700hpa (approximately 3100 meters), and 600hpa (approximately 4500 meters) respectively. x6, x7, x8, x9 and x10 indicate dry adiabatic lapse rates, i.e., the dry bulb temperature difference between surface and 850hpa (approximately surface to 1500 meters), between 850hpa and 700hpa (approximately 1500 to 3100 meters), between 700hpa and 600hpa (approximately 3100 to 4500 meters), between 600hpa and 400hpa (approximately 4500 to 7500 meters), and between 400hpa and 300hpa (approximately 7500 to 9600 meters) respectively. Remaining three weather inputs i.e. , x11, x12,x13indicate vertical wind shear at 900hpa to 700hpa (approximately 980 meters to 2500 meters), at 700hpa to 500hpa (approximately 2500 meters to 12340 meters), and at 500hpa to 200hpa (approximately 12340 meters to 35000 meters) respectively. Moisture difference is the difference between dry bulb temperature and the dew point temperature. This moisture, when carried out to the upper atmosphere by vertical wind shear forms the thundercloud [10]. Dry adiabatic lapse rate is the measure of the conditional instability of the atmosphere [11]. The more the atmosphere is unstable, more moisture would be carried out to the upper atmosphere from the surface level to form thunderclouds [11]. Vertical wind shear plays an important role to carry the moisture to the upper atmosphere. This is very significant for the formation of thundercloud, [3].

3. METHODOLOGY

In this paper K-nearest neighbor technique has been adopted for correct forecasting of 'squall days' and 'no squall days'. In the classification phase, K is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the K training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the **ove**rlap metric (or Hamming distance). The naive version of the algorithm is easy to implement by computing the distances from the test example to all stored examples, but it is computationally intensive for large training sets. Using an appropriate nearest neighbor search algorithm makes KNN computationally tractable even for large data sets. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. KNN is a special case of a variable-bandwidth, kernel density "balloon" estimator with a uniform kernel, [12][13]. In this paper the total data set is divided into two classes and these are training dataset and test dataset. The total number of squall data is 69 and total number of no squall data is 315. The 'squall days' and 'no squall days' have been arranged consecutively in four phases, each phase having seven data sets. Each data set contained 10 data points, keeping six sets for training purpose and one is for test purpose. Now from the remaining 'no squall days', again 69 data points has been taken and combined with 69 'squall class days' for training purpose. This process has been repeated in four phases with different sets of 69 numbers of 'no squall' data points and the same set of 69 number of 'squall' data points. Thus, all the possible combinations of data have been constructed and checked by applying K-NN methodologies (1-nn, 3-nn and 5-nn). The similarity measure has been taken between each data vector of test set with each data vector of training set. Similarity between two observation vectors say,
$p = (p_1, p_2, \ldots\ldots, p_\gamma)$, $q = (q_1, q_2, \ldots\ldots, q_\gamma)$ is defined as,

$$\frac{\sum p_{11k} X q_{1mk}}{\surd\,(p^{2}_{11k} q x^{2}_{1mk})}$$

The similarity measures between two vectors reflect the cosine of the angles between them. The similarity is more if the angle is smaller. The similarity measure indicates vicinity between the two data vector with each of the training data vector is determined. These cosine angles are arranged in the decreasing order. Here the value of K is chosen as 1, 3 and 5.

4 RESULTS

SET 1	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	9, 90%	2, 20%	11, 55%	8, 80%	7, 70%	15, 75%	6, 60%	8, 80%	14, 70%	4, 40%	6, 60%	10, 50%
3-NN	9, 90%	7, 70%	16, 80%	8, 80%	8, 80%	16, 80%	9,90%	9, 90%	18, 90%	10,100%	9, 90%	19, 95%
5-NN	9, 90%	3, 30%	12, 60%	9, 90%	8, 80%	17, 85%	9, 90%	8, 80%	17, 85%	10, 100%	7, 70%	17, 85%

SET 2	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	6, 60%	6, 60%	12, 60%	4, 40%	6, 60%	10, 50%	7, 70%	7, 70%	14, 70%	6, 60%	11, 55%	8, 40%
3-NN	10, 100%	8, 80%	18, 90%	8, 80%	8, 80%	16, 80%	10, 100%	9, 90%	19, 95%	5, 50%	15, 75%	10, 50%
5-NN	10, 100%	8, 80%	18, 90%	8, 80%	9, 90%	17, 85%	10, 100%	8, 80%	18, 90%	4, 40%	14, 70%	9, 45%

SET 3	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	3, 30%	5, 50%	8, 40%	4, 40%	3, 30%	7, 35%	5, 50%	8, 80%	13, 65%	3, 30%	9, 90%	12, 60%
3-NN	8,80%	8, 80%	16, 80%	7, 70%	7, 70%	14, 70%	8, 80%	9, 90%	17, 85%	8, 80%	10, 100%	18, 90%
5-NN	8, 80%	5, 50%	13, 65%	7, 70%	7, 70%	14, 70%	7, 70%	6, 60%	13, 65%	8, 80%	9, 90%	17, 85%

SET 4	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	7, 70%	8, 80%	15, 75%	7, 70%	8, 80%	15, 75%	5, 50%	4, 40%	9, 45%	5, 50%	3, 30%	8, 40%
3-NN	9, 90%	8, 80%	17, 85%	8, 80%	9, 90%	17, 85%	10, 100%	7, 70%	17, 85%	7, 70%	5, 50%	12, 60%
5-NN	10, 100%	7, 70%	17, 85%	8, 80%	5, 50%	13, 65%	10, 100%	6, 60%	16, 80%	7, 70%	3, 30%	10, 50%

SET 5	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	7, 70%	7, 70%	14, 70%	6, 60%	9, 90%	15, 75%	6, 60%	6, 60%	12, 60%	9, 90%	5, 50%	14, 70%
3-NN	9, 90%	8, 80%	17, 85%	8, 80%	10, 100%	18, 90%	8, 80%	10, 100%	18, 90%	9, 90%	5, 50%	14, 70%
5-NN	9, 90%	4, 40%	13, 65%	8, 80%	10, 100%	18, 90%	9, 90%	7, 70%	16, 80%	9, 90%	4, 40%	13, 65%

SET 6	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**
1-NN	5, 50%	6, 60%	11, 55%	5, 50%	8, 80%	13, 65%	4, 40%	6, 60%	10, 50%	5, 50%	3, 30%	8, 40%
3-NN	7, 70%	6, 60%	12, 60%	7, 70%	7, 70%	14, 70%	6, 60%	6, 60%	12, 60%	7, 70%	5, 50%	12, 60%
5-NN	9, 90%	6, 60%	15, 75%	6, 60%	7, 70%	13, 65%	6, 60%	6, 60%	12, 60%	7, 70%	3, 30%	10, 50%

SET 7	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
	Total no. of days = 10 (squall)	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 10 (squall)**	**Total no. of days = 10 (no squall)**	**Total no. of days = 20**	**Total no. of days = 9 (squall)**	**Total no. of days = 9 (no squall)**	**Total no. of days = 18**
1-NN	5, 55.5%	4, 44.4%	9, 50%	6, 66.6 %	5, 55.5%	11, 61.1 %	6, 75%	5, 62.5 %	11, 68.75 %	7, 77.7 %	4, 44.4 %	11, 61.1 %
3-NN	6, 66.6%	6, 66.6%	12, 66.6%	7, 77.7 %	6, 66.6%	13, 72.2 %	7, 87.5 %	5, 62.5 %	12, 75%	9, 100%	6, 66.6 %	15, 83.3 %
5-NN	7, 77.7%	6, 66.6%	13, 72.2%	5, 55.5 %	5, 55.5%	10, 55.5 %	6, 75%	4, 50%	12, 62.5 %	9, 100	5, 5.55 %	14, 77.7 %

Here we can see the 3-nn gives most promising result among the above. It gives 95% accurate prediction for the total data set.

5 DISCUSSIONS AND CONCLUSION

Correct forecasting of severe thunderstorms is always a difficult job due to dynamic character of weather. Here three kind of weather parameter i.e., moisture difference, adiabatic lapse rate and vertical wind shear has been considered. Regardless of the direction of shear, this process induces flow perpendicular and to the left of the shear vector, and thus by itself introduces a leftward deflection of the motion from the shear [14]. A clockwise turning of the wind shear vector with height favors the development of a cyclonic, right-moving storm; while conversely, a counterclockwise turning favors the anticyclonic, left-moving storm [15]. Chakrabarty et. al., in their previous paper got 91.42% correct prediction for 'squall days' and 87.63% for 'no squall days', whereas using the same data set we get more than 98% correct prediction for 'squall days' and 90% for 'no squall days' with 12 hour lead time.

6 REFERENCES

[1] Chakrabarty 1, Himadri, C. A. Murthy, Sonia Bhattacharya and Ashis Das Gupta, May, 2013. "Application of Artificial Neural Network to Predict Squall-Thunderstorms Using RAWIND Data", International Journal of Scientific & Engineering Research, Volume 4, Issue 5, pp. 1313-1318, ISSN 2229-5518.

[2] Federal Emergency Management Agency (FEMA) is an agency of the United States Department of Homeland Security.

[3] Himadri Chakrabarty, Sonia Bhattacharya , "Application of K-Nearest Neighbor Technique to Predict Severe Thunderstorms" International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 10, January 2015.

[4] Ramaswami, C., 1956. "On the sub-tropical jet stream and its role in the development of large-scale convection", Tellus, 8, 26-60.

[5] Y. Radhika and M. Shashi, "Atmospheric Temperature Prediction using Support Vector Machines", International Journal of Computer Theory and Engineering, Volume 1, Number 1, pp.55-58, 2009.

[6] Amit Kesarkar, "PREDICTION AND CLASSIFICATION OF THUNDERSTORMS

USING ARTIFICIAL NEURAL NETWORK" International Journal of Engineering Science and Technology ISSN**:** 0975-5462, Vol. 3 No. 5 May 2011.

[7] Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011) Miscellaneous Clustering Methods, in Cluster Analysis, 5th Edition, John Wiley & Sons, Ltd, Chichester, UK.

[8] Brath, A., Montanari A and Toth E, 2002, "Neural networks and non-parametric methods for improving real-time flood forecasting through conceptual hydrological models", Hydrology and Earth System Sciences, Vol. 6 (4), pp.-627-640.

[9] Jayawardena, A.W., Fernando D.A.K. and Zhou M.C., 1997, "Comparison of Multilayer Perceptron and Radial Basis Function networks as tools for flood forecasting", Proceedings of the Conference Water-Caused Natural Disasters, their Abatement and Control, held at Anaheim, California, Publ. no. 239.

[10] Moran, J.M., Moran M.D. and Pauley P.M., Meteorology: "The Atmosphere and the Science of Weather", 5th Edition, Chapter 6, Prentice Hall, 1997.

[11] Volland, Hans, 1995, "Handbook of Atmospheric Electrodynamics", Vol. 1, ISBN: 0-8943-8647-0(V. 1)

[12] D. G. Terrell; D. W. Scott (1992). "Variable kernel density estimation". Annals of Statistics **20** (3): 1236–1265. doi:10.1214/aos/1176348768.

[13] Mills, Peter. "Efficient statistical classification of satellite measurements". *International Journal of Remote Sensing*.

[14] Kristen L. Corbosiero and Molinari John "The Relationship between Storm Motion, Vertical Wind Shear, and Convective Asymmetries in Tropical Cyclones", Journal of Atmospheric Sciences, 2002 Volume 60 Page No. 366.

[15] Rotunno, Richard and Klemp Joseph B "The Influence of the Shear-Induced Pressure Gradient on Thunderstorm Motion", 1982, Monthly Weather Review, Vol. 110, pp. 136-151.

A Robust Feature Descriptor: Signed LBP

Chu-Sing Yang [1], **Yung-Hsian Yang** [*1]

[1]Department of Electrical Engineering, National Cheng Kung University, No.1, University Road, Tainan City 701, Taiwan. Email: csyang@mail.ee.ncku.edu.tw

[*] Corresponding Author: Tel.: + 886-989540617; E-mail: q38991043@mail.ncku.edu.tw

Abstract - *We improve a texture descriptor, Local Binary Feature (LBP), called Signed Local Binary Pattern (SLBP) which is more robust in rotation and scale. In this paper we will introduce how to obtain more information in LBP, which with signed bit, and make feature more stable with mean of local area instead of single pixel's intensity. Finally, to reach more robust in scale by difference smooth factor and implement by Integral Images to reduce computation cost. A pixel can perform different texture information in each scale, thus we select meaningful edge type in smallest scale. And signed bit is adding by current center area is greater or less then whole neighbor area. Then we implement cell and block concept from Histogram of Gradient to test the character recognition. In result part we prove SLBP have more robust than LBP in rotation, scale by texture image and natural scene image. The last part is testing the performance of recognition rate in IIIT5K database.*

Keywords: Local Binary Pattern, Texture Descriptor, Integral Image, Histogram.

1 Introduction

Recently, number of digital image increased very fast by street camera, portable camera device, cell-phone, and even google street view with some flaws such like low resolution, blurred with hand shake, or low luminance. These flaws result it's hard to extract interest image object from image database, but the information in images is usually useful in many field such like surveillance system, image search engine, auto license plate recognition (ALPR), text recognition, or human and face detection etc., which can help our life to be more convenience. And this information is too large to analysis by human resource. Thus the image object recognition will play an important position in intelligence system in feature.

By Bag-of-Words [1] model is more popular in object recognition at current computer vision. Its performance is based on descriptor's robustness to find the separate visual words and statistic these word into an image object. Therefore, stronger descriptor had a prominent part in this model. In general, descriptor have three parts to overcome: luminance change, rotation, and scale. In original Local Binary Pattern, LBP^{riu2} [2] [3] [4], has been proven capability in gray scale and rotation invariant in texture recognition. Based on LBP, we improve the LBP kernel with sign bit and remove redundancy information in original LBP bring to more texture information in pixel level. Then we search the scale to find the significant edge type at current pixel. Like HOG [5] we build two level histogram information with cell and block to statistic edge information into high dimension to recognition text object.

In the remaining part, related work will explain original LBP, Integral image, histogram and relative paper's technology we sited. Chapter 3 will expound our method in detail, and experimental results in chapter 4 with compared between original LBP and SLBP in rotation, scale, gray scale and text recognition with IIIT5K [6] database.

2 Related work

In visual word and bag of word model were most popular in image object recognition and matching, how to descript the visual word efficiently and high accuracy become more important to achieve robust and invariant at many kinds. Image descriptor can classify into three part roughly, color descriptor, texture descriptor, and intensity descriptor. Color descriptor were out of favor because unexpected luminance will make information lost in color domain, thus major methods trend to texture base and intensity base method.

2.1 Histogram of Gradient

Histogram of Gradient (HOG) is robust descriptor to static eadges. HOG count the edge's gradient and magnitude by an area called cell and histogram into angle bins, which number of bin can adjust by case. Then construct these cells into overlapped block as feature. Each block were normalized to overcome the unbalanced intensity. HOG feature size will be decide by number of angle's bin, cell size, block size, block overlap, and image size. In our experiment accuracy of HOG in proportion to feature number. And HOG have high recognition rate in text recognition. Although HOG can extract more detail of edge features, but it still have some drawbacks to be overcome, such like how to choose edge operator, smooth operator size [7], descriptor blocks (C-HOG), or use image pyramid to achieve scale invariant, which make high computation cost.

2.2 Haar-like Feature and Integral Image

In viola and Jones's [8] work propose a Haar-like feature to detect the different intensity in between rectangles. For example in face detection, nose and cheek have significant intensity difference. And then implement by Integral Image (II), which is a fast way to calculate sum of a rectangle by four operator after II was construct. Integral image was propose by Frank Crow at 1984 and widely used in invariant feature extraction [9]. II can define as equation 1.

$$II = \sum_{\substack{x'<x \\ y'<y}} i(x', y') \qquad (1)$$

The value of $II(x, y)$ means the sum of image from left-top to point(x, y), thus to calculate a sum of Specific area sum by $area\ sum = A - B - C + D$.

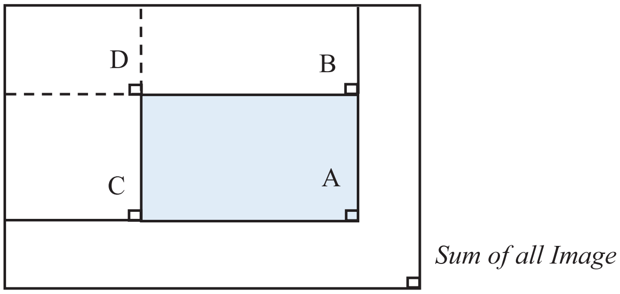


Figure 1. Schematic to show Integral Images

Instead of the scale problem the method have a large feature table in 24×24 learning image. And then they select the most significant features to filter out the none-face sub-images by using cascade Adaboost [10] classifier, which combined by many weak classifier into a strong classifier to achieve detection speed. The error rate will decrease when strong classifier is increase.

2.3 Region base methods

The bottle neck of Region base match is how to extract complete regions after view angle and luminance change. MSER [11] is an efficiency method to extract the region object, in Per-Eric's work performs the performance will effect by blur.

In Chen's [12] work propose a edge enhancement method to divide MSER Regions by edge detection result. In

2.4 Rotation invariant feature and descriptor

BRISK [13] proposed an idea similar to DAISY [14], which is pre-calculate smooth image with pyramid kernel to reduce redundant computing cost, and also use the binary feature to fit the interest points. Then the trend start to find the pairs from coarse-to-fine from the bigger Gaussian kernel at periphery and smaller kernel when closer to center to extract more detail. With this trend, inspired us to develop multi-scale in Local binary pattern. ORB (Oriented FAST and Rotated BRIEF) [15] is a fast robust local feature detector and It is based on the FAST [16] keypoint detector and the visual descriptor BRIEF. Its aim is to provide a fast and efficient alternative to SIFT.

2.5 Local Binary Pattern (LBP)

LBP feature is compare each pixel with neighbor pixels to obtain a circle binary feature shown as figure 2, and equation shown as equation 2. Where g_c is gray value at current pixel and g_p is gray value of its neighbors and binary feature express by compare g_c with every g_p with equation 3. Notice that P is pair number and R is the radius from central pixel. Features present the relationship between central pixel and neighbor and express by binary feature with 36 kinds of combination. In rotation invariant version of LBP was achieved by circular bit-wise right shift (ROR). Then in Ojala's [4] work, mark U as transition between '0' and '1'. By Ojala's [4] experiment, LBP feature have significant meaning when $\mathrm{U} \leq 2$ so they only count the first P+1 combination and group remind feature into same category as P+2. Mark that the features with $\mathrm{U} \geq 4$ is rare and can't express local texture well or we can called it is disorganized. Thus the number of combination of LBP feature is reduced to P+2, where is 10 bins in $LBP_{8,1}^{riu2}$. And author also present stronger descriptor by extent P and R

$$LBP_{P,R} = \sum_{P=0}^{P-1} s(g_p - g_c)2^p \qquad (2)$$

$$\mathrm{s(p)} = \begin{cases} 1, p \geq 1 \\ 0, other\ wise \end{cases} \qquad (3)$$

3 Main Method

In this section we will introduce Signed Local Binary Pattern (SLBP), which have better performance keep the same feature bin. We will point out the difference between SLBP and original LBP step by step.

3.1 LBP Base on Integral Image

In our proposed method, feature is computing base on Integral Image to speed up. So at first, we defined the II as Integral Image and $II_s(\mathrm{x,y})$ is mean value of a sum of area as equation 4.

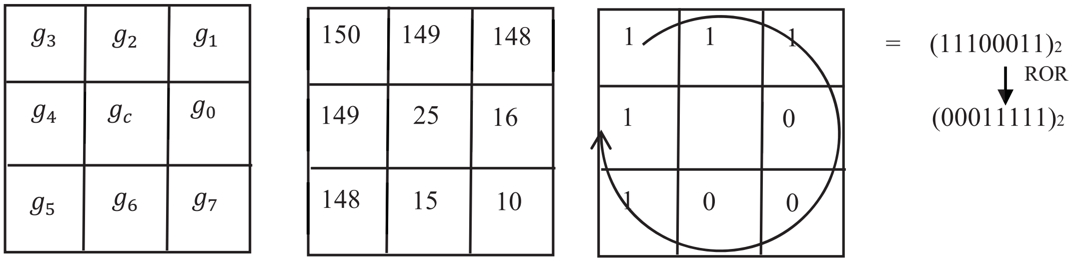


Figure 2. Schematic diagram of $\mathrm{LPB}_{8,1}^{riu2}$

$$
\begin{gathered}
II_s(x, y) = \\
(II(x + 3^s, y + 3^s) + \\
II(x - 3^s - 1, y - 3^s - 1) - \\
II(x + 1, y - 3^s - 1) - \\
II(x - 3^s - 1, y + 1))/3^{2s}
\end{gathered}
\quad (4)
$$

Parameter s is scale for smooth factor as 3^s with s = [1, 2… N] to present each scale. The final scale can be adjust by current situation.We mark the II_0 as original image. Thus first LBP feature can present as equation 5. Briefly, $II_1(x, y)$ means average value in 3 × 3 block instead of g_c to build LBP in first scale and II_0 means original image.

$$\mathrm{LBP}_{8,1,s}(\mathrm{x}, \mathrm{y}) = \sum_{p=0}^{8-1} s(II_s(x, y) - II_{s-1}(x, y, g_p)) \times 2^p \quad (5)$$

This idea was presented by MLBP [17], and we also agree with this approach is more robust than original LBP. Thus we also use Integral Image to obtain first scale LBP by compare with mean value of 3 × 3 block. Then we also use the ROR operator to achieve rotation invariant.

$$\mathrm{LBP}_{8,1,s} = \min\{\mathrm{ROR}(LBP_{8,1,s}i), i = 1,2,3, \ldots, 8 - 1\} \quad (6)$$

3.2 Scale invariant

As mention before the feature is meaningless when U ≥ 4, these pixels maybe were electronic noise at current scale. For Example, shows in Figure 3, we extract the LBP feature at yellow box with $(00101011)_2$ after ROR and this information is meaningless when we histogram this into a bin in P+2. But in next scale, blue box, the pixel can extract the useful feature as $(00011111)_2$ after ROR process. So in our algorithm, we extract P+1 feature bins from $(00000000)_2$ to $(11111111)_2$ at current scale and keep zero when U ≥ 4. By integral image, the computation cost of each scale is same. Schematic diagram is shown as Figure 4.

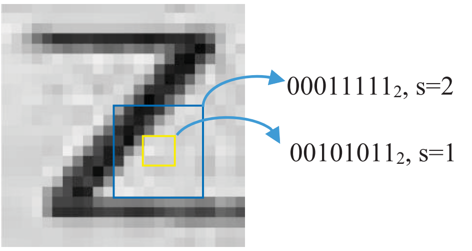


Figure 3. Image shows the difference scale can extract different feature

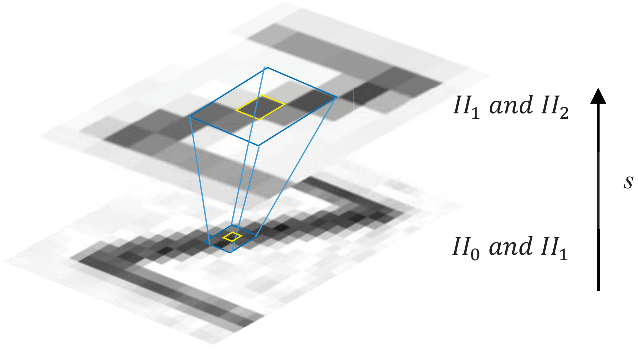


Figure 4. First and third images show current pixel is in edge. Second and fourth images show current pixel is out of edge.

Then the LBP feature can extract from each scale with difference meaning. Without loss of generality, we select the first none-zero value in scale space as LPB feature to avoid too much blur to interfere correct edge information. So the LPB with scale invariant ($LPB_{1,8}^{si}$) feature can express as equation 7.

$$\mathrm{LPB}_{8,1}^{\mathrm{si}}(\mathrm{x}, \mathrm{y}) = \arg \underset{\mathrm{LPB}_{8,1,s}}{\min} \mathrm{s}, \text{if } \mathrm{LPB}_{8,1,s} \neq 0. \quad (7)$$

And in our experiment, scale s were set to 3 can achieved every pixel have significant edge type.

3.3 Signed Bit

Last step is to decide the sign bit, which can descript the center information from LBP. Since we apply the propose method from MLBP, which is use the mean value to instead of center intensity to achieve more stable. It's makes the intensity of center area can be an additional information for current pixels. Thus we modified the LBP equation into equation 8.

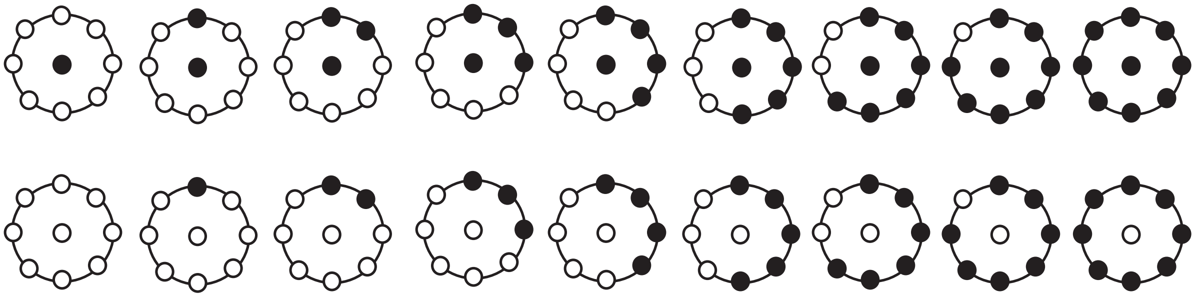


Figure 5. All edge type in SLPB, which from black spot to white spot.

$$SLBP_{8,1}(x,y) = \begin{cases} LBP_{8,1}^{si}(x,y)+1, \;\; if \; g_{c,s-1}(x,y) \geq II_s(x,y) \\ LPB_{8,1}(x,y), \;\; otherwise \end{cases} \tag{8}$$

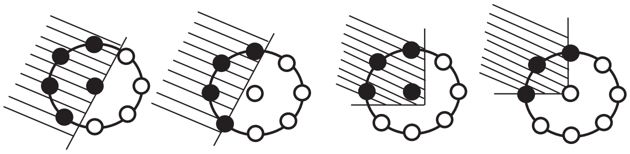


Figure 6. Schematic diagram for integral image apply to each scale.

After add sign bit, the SLBP feature bin will resume into 10 bins, but this sign bit can express the edge type more specifically. The main physical meaning is showing as Figure 5. It mark the current pixels is in edge or out of edge. In our experiments use sign bit to instead P+2(U ≥ 4) type can improve significant recognition rate in text recognition. And total combination of edge type shows as Figure 6.

The goal of the SLBP, we proposed, is to extract the useful edge information form text. So after features extracted in pixel level, we use the concept in Histogram of gradient to divide pixel into cell; and overlapped cells into block to build high dimension feature vector. After this step we expect similar recognition rate in text, but reverse is true, the performance of SLBP or LBP for text recognition is disappointed; we will show the result in next section.

4 Result and experiment

In generally, we will test the rotation, scale and luminance first in this section. And the second part is to apply in character recognition by IIIT5K with simple method: k-nearest neighbor (KNN).

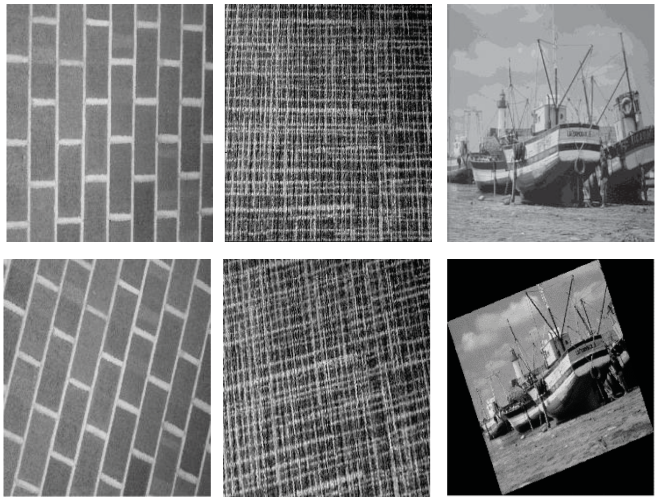


Figure 7. First row shows test images, and second row shows rotated test

We use three images, which is 'Brick Wall', 'Carpet', and 'Boat' to test invariant and compare between SLBP$_{8,1}$ and original $LBP_{8,1}$. And we believe the result will be much better when implement higher P and R in SLBP. But we remind high order SLPB in our feature work because complete SLBP need more detailed reflections and we will implement on CUDA architecture to achieve real time feature descriptor extraction.

4.1 Rotation Invariant Test

At first, we test rotation invariant by rotated image with 360 degree and 10 bins in both LBP and SLBP. First row in figure 7 shows original images and second row shows rotated images respectively. Results shows as Figure 8, 9, and 10 in squared error with none rotated images. And we can observe obviously four peaks, which is caused by discrete digital image and without Gaussian weight. This phenomenon can eliminate by high order P or weighted by Gaussian. The result shows the squared error. Form the seriese of results, SLBP has proved have lower error than the original LBP. We used exist original LBP function build in Matlab and set the parameters same as SLBP to obtain fair comparison result. In rotation test, SLBP have lower square error in each image. It is because we use the mean value instead of single pixel and search the scale space to find meaningful edge at each pixel.

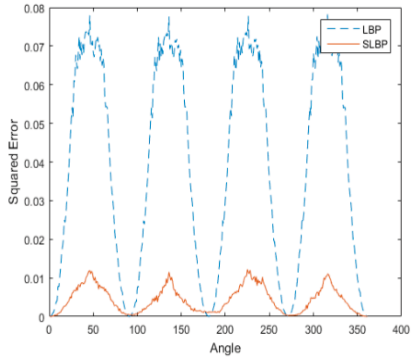


Figure 8. Square error in 360 degree rotation with Brick Wall.

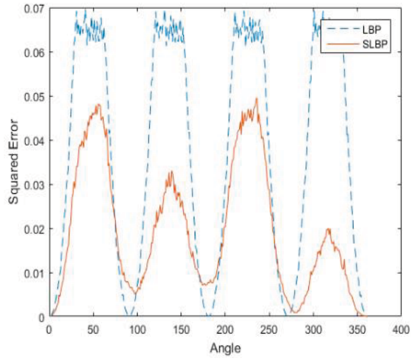


Figure 9. Square error in 360 degree rotation with Carpet.

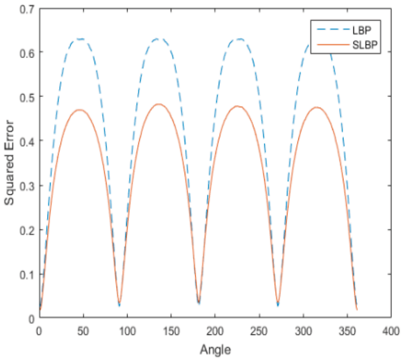


Figure 10. Square error in 360 degree rotation with Carpet

4.2 Scale Invariant Test

In scale test, we scaled image from scale factor 0.5 to 1.5 and also calculate squared error as Figure 11, 12 and 13. Result shows SLBP is more robust in scale than original LBP and squared error is less than 0.15 when images size was scaled by 0.5. Searching scale space to find correct edge type make the SLBP have more scale invariant.

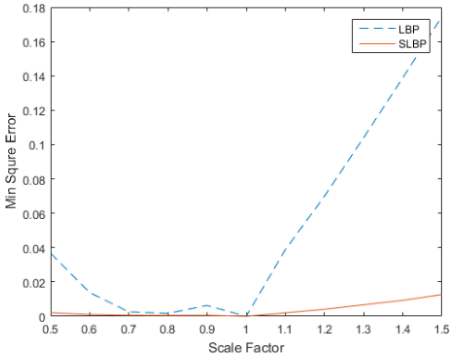


Figure 11. Square error in scale test with Brick Wall.

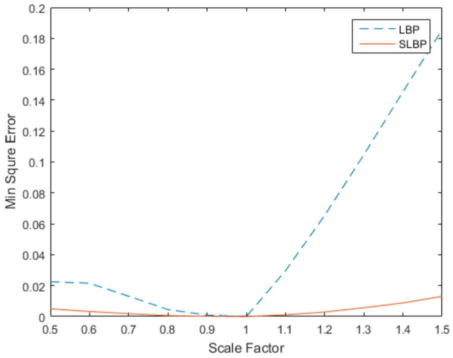


Figure 12. Square error in scale test with Carpet.

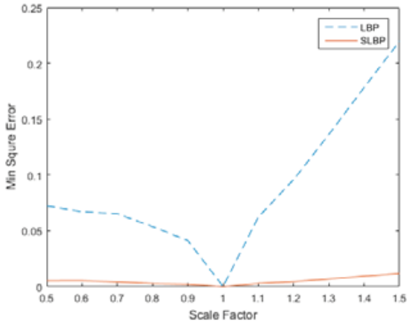


Figure 13. Square error in scale test with Boat.

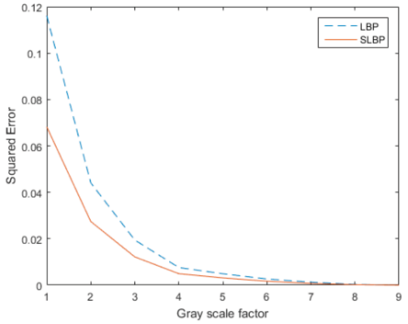


Figure 14. Square error in gray scale test with Brick Wall.

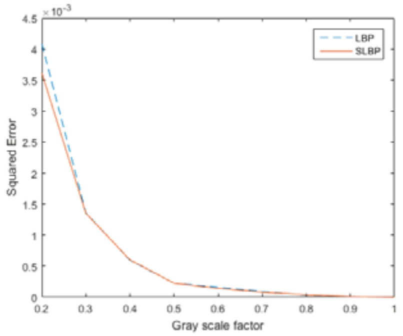


Figure 15. Square error in gray scale test with Carpet.

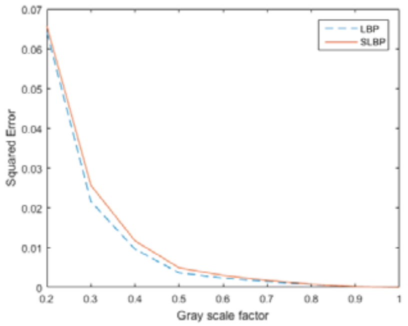


Figure 16. Square error in gray scale test with Boat.

4.3 Luminance Invariant Test

And in luminance test, we didn't actually take pictures in each luminance. So we used alternative solution, which scale the gray value of original images from 0.2 to 1 to simulate different luminance. The results show in figure 14, 15, and 16. As our expected, LBP and SLBP have similar performance in gray scale. But LBP's gray scale invariant has been proved in [4]'s work, thus compression does not make much sense.

4.4 Character Recognition Test

We test the IIIT5K character database Last, Each character in IIIT5K training database was extracted and normalized these images size into 27×27 pixel^2 to build KNN model. Which KNN classifier is set K = 63 for 0-9, a-z, up-case, and none-character. Parameters in LBP and SLBP is setup by $P = 8, R = 1$, and 2 cell size, which is 3 and 9, to descript text texture. And in test part, we also crop the image from test database then normalize the image size as test dataset. Although KNN is a simple classifier, but it can test descriptor's performance quickly and easy to implement. The result shows as Table 1. In this paper we only compare original LBP and SLBP by simple method, so recognition rate is low as our expected without any particular algorithm. On the other hand, we thought LBP feature is not enough capability in text recognition. The same method in HOG performed much better than LBP and SLBP. But we will keep improve LBP feature to fit text recognition. In same situation, HOG with 4 cell and 2 overlap block's recognition rate is 84.4%. This result allows us to doubt LBP is a good repetitive texture descriptor, but not a good solution in character recognition.

Table 1. Compare the Character Recognition Performance with LBP, SLBP, and blocks SLBP

Descriptor	*Number of bins*	*Recognition rate*
$LBP_{8,1}$ *with* 3×3 *cell*	90	30.2%
$LBP_{8,1}$ *with* $9 \times 9\ cell$	810	40.8%
$SLBP_{8,1}$ *with* 3×3 *cell*	90	**51.3%**
$SLBP_{8,1}$ *with* 9×9 *cell*	810	**64.1%**
$SLBP_{8,1}$ *with* 3×3 *cell, 4 blcoks*	540	53.2%
$SLBP_{8,1}$ *with* 9×9 *cell, 4 blcoks*	1440	68.1%

5 Conclusions

In this paper, we improve SLBP which is more robust than original LBP in rotation, scale and similar performance in gray scale. SLBP remove redundancy edge type and fetch more useful information by scale up in scale space and add sign bit to make pixels have correct edge type and more information. Although we only compared with P = 8 and R = 1 but result can be foreseeable with significant improvement when SLBP implement in higher P and R. In computation time, SLBP cost N times than original LBP when N scale space, in this paper N = 3. But still keep small time complex by LBP originally is a fast and simple algorithm. This paper is additional reward when we research for more robust feature and descriptor in text extraction and recognition in real scene. Thus, to complete full SLBP in different P and R and apply it into text extraction is our feature work.

6 References

[1] J. S. Zisserman, «Efficient visual search of videos cast as text retrieval,» *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* p. 591–605, February 2009.

[2] D. He et L. Wang, «Texture Unit, Texture Spectrum, And Texture Analysis,» *Geoscience and Remote Sensing, IEEE Transactions on,* pp. 509 - 512, 1990.

[3] L. Wang et D. He, «Texture Classification Using Texture Spectrum,» *Pattern Recognition,* pp. 905 - 910, 1990.

[4] T. Ojala, l. Pietikainen et T. Maenpaa, «Multiresolution Gray Scale and Rotation Invariant Texture Classification With Local Binary Patterns,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 971-987, July 2002.

[5] N. Dalal et B. Triggs, «Histograms of Oriented Gradients for Human Detection,» *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* p. 886–893, June 2005.

[6] A. Mishra, K. Alahari et C. Jawahar, «Scene text recognitionition using higher order language priors,» chez *Proceedings of the British Machine Vision Conference*, 127.1-127.11.

[7] N. He, J. Cao et L. Song, «Scale Space Histogram of Oriented Gradients for Human Detection,» chez *International Symposium on Information Science and Engieering*, 2008.

[8] P. Viola et M. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features,» chez *COMPUTER VISION AND PATTERN RECOGNITION 2001*, 2001.

[9] H. Bay, A. Ess, T. Tuytelaars et L. V. Gool, «Speeded Up Robust Features,» *Computer Vision and Image Understanding,* p. 346–359, June 2008.

[10] Y. Freund et R. E. Schapire., «A decision-theoretic generalization of on-line learning and an application to boosting,» chez *Computational Learning Theory: Eurocolt*, Springer-Verlag, 1995.

[11] P.-E. Forssen et D. Lowe, «Shape Descriptors for Maximally Stable Extremal Regions,» chez *ICCV*, 2007.

[12] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R.

Grzeszczuk et B. Girod, «Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions,» chez *18th IEEE International Conference on Image Processing*, Brussels, 2011.

[13] S. Leutenegger, M. Chli et R. Y. Siegwart, «BRISK: Binary Robust Invariant Scalable Keypoints,» chez *ICCV*, 2011.

[14] V. L. a. P. F. E. Tola, «Daisy: An efficient dense descriptor applied to wide-baseline stereo,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 815-830, 3 2010.

[15] E. Rublee, V. Rabaud, K. Konolige et G. Bradski, «ORB: an efficient alternative to SIFT of SURF,» *IEEE International Conference on Computer Vision,* pp. 2564-2571, 2011.

[16] E. Rosten et T. Drummond, «Machine learning for highspeed speed corner detection,» chez *In European Conference on Computer Vision*, 2006.

[17] B. O'Connor et K. Roy, «Facial Recognition using Modified Local Binary Pattern and Random Forest,» *International Journal of Artificial Intelligence & Applications (IJAIA),* pp. 25-33, November 2013.

SESSION

7TH WORKSHOP ON SOFT COMPUTING IN IMAGE PROCESSING AND COMPUTER VISION, SCIPCV

Chair(s)

Dr. Gerald Schaefer
Dr. Iakov Korovin

Differential Evolution Algorithm-based Range Image Registration with a Novel Point Descriptor

Taifeng Li, Liang Gao*, Quanke Pan, Peigen Li
The State Key Laboratory of Digital Manufacturing Equipment and Technology
Huazhong University of Science & Technology, Wuhan, China

Abstract - *Range image registration plays increasingly important role in a variety of research fields. In general, the registration results are sensitive to many practical factors, such as the sizes, geometries, and initial positions of the range images. This paper is such an attempt to register two range images with different sizes under complex initial situations. First, in coarse registration, a novel point descriptor is designed based on the electrostatic field theory. In term of the identified properties space, a good source of prior information is achieved. Then, combine with the coarse registration, an enhanced differential evolution algorithm is proposed for fine registration. Experiment results reveal that the proposed methods are able to provide competitive results to solve the challenging registration problems when compared with the other methods.*

Keywords: Range image registration, Point descriptor, Differential evolution algorithm

1 Introduction

Range image registration is a critical technique in computer vision and pattern recognition areas. In the past years, it has been widely used in quality inspection [1], virtual museum [2], reverse engineering [3], and other fields. The goal of the registration is to find the optimal transformation matrix to align two range images as close as possible. The transformation matrix includes rotation and translation parameters, in some papers the scale parameters are taken into account also [4]. According to the matching accuracy, registration can be divided into coarse registration and fine registration [5]. In general, the coarse registration is an approximate matching to provide good initial position for fine registration.

In coarse registration, the property of each point is extracted to search the correspondence between two point clouds. To improve the efficiency, generally, only the points which can effectively contribute to finding the good corresponding point pairs are used. In literature, the normal and curvature are the popular information for point description [6, 7]. However, they are difficult to describe the distinctive points for the highly symmetric or planar models, and the process might result in selecting many points that essentially contain the same information. Rusu et al. [8] proposed a persistent feature histograms (PFH) to match point clouds from different views, and each point was estimated by a 16D features based on the normal. Guo et al. [9] presented a rotational projection statistics (RoPS) for local feature description of point set. Yang et al. [10] proposed a local feature statistics histogram (LFSH) for registration, in LFSH the local depth, point density, and normal were encoded to describe the local shape geometries. In fine registration, iterative closest point (ICP) [11] is the best known method. However, the ICP is sensitive to the initial position of two models, and it is easy trapped in local minima when two models are far away from each other. In practical application, the complex position and the geometric shape of the models make the registration problem more difficult. More recently, in the literature, range image registration is considered as an optimization problem, and some heuristic algorithms are employed to solve this problem. Such as the simulated annealing (SA) algorithm [12], genetic algorithm (GA) [13], scatter search (SS) algorithm [14], and artificial bee colony (ABC) algorithm [15].

In this paper, we propose a coarse-to-fine method for range image registration. In coarse registration, the point cloud is considered as a conductor with electrostatic equilibrium, and the points are seen as the free charges of the conductor. Then, a novel point descriptor is designed based on the electrostatic field theory (EFT). With EFT, the electric force, electric filed, and electric potential energy are encoded for point description. In fine registration, based on the initial solutions achieved in the coarse registration, we employed the improved differential evolution algorithm (DE) for searching the global optimal solution. The contributions of this paper are as follows.

- A novel point descriptor is designed based on the EFT for coarse registration.

- In terms of the coarse registration, an improved DE algorithm is proposed for fine registration.

- The proposed methods can achieve successful results for range image registration with complex initial situations.

The rest of this paper is structured as follows. Section 2 presents the EFT-based point descriptor. The improved DE algorithm is proposed in Section 3. Experiments and analyses are conducted in Section 4. Finally, Section 5 gives the conclusion and future work.

2 Electrostatic field theory-based point descriptor

In electrostatic field, the conductor contains a lot of free charges which move easily. Due to the interaction of each charge, finally a steady state (called electrostatic equilibrium) of the conductor is achieved. There are many similarities between point cloud and conductor. First, the location of the points and the free charges are fixed in the point cloud and conductor, respectively. Second, the property of the point and free charge can be approximated by its neighborhood. Furthermore, the curvature is an important information for point cloud, and the value of the curvature may be positive and negative for different points. Similarly, the free charges are the signed magnitudes, and the quantity of each charge is different. In this paper, the points are considered as the free charges with electrostatic equilibrium, and the property of each point is described by the electric force, electric filed, and electric potential energy.

2.1 Electric force

Given two static points c_1 and c_2, assuming that the electric quantity of c_1 and c_2 are $+q_1$ and $+q_2$, respectively. According to the Coulomb's law, the vector form of the electric force between particles c_1 and c_2 is as follows:

$$\vec{F}_{21} = k_e \cdot \frac{q_1 q_2}{\left|r_{12}\right|^2} \cdot \hat{r}_{21} \qquad (1)$$

where k_e is Coulomb's constant k_e=8.99×109 Nm^2C^{-2}, and the $\hat{r}_{21}$ denotes a unit vector pointing from c_2 to c_1. In this paper, we assign the quantity of electricity for each point based on the curvature value.

$$q_i = \lambda \cdot C_{ci} \qquad (2)$$

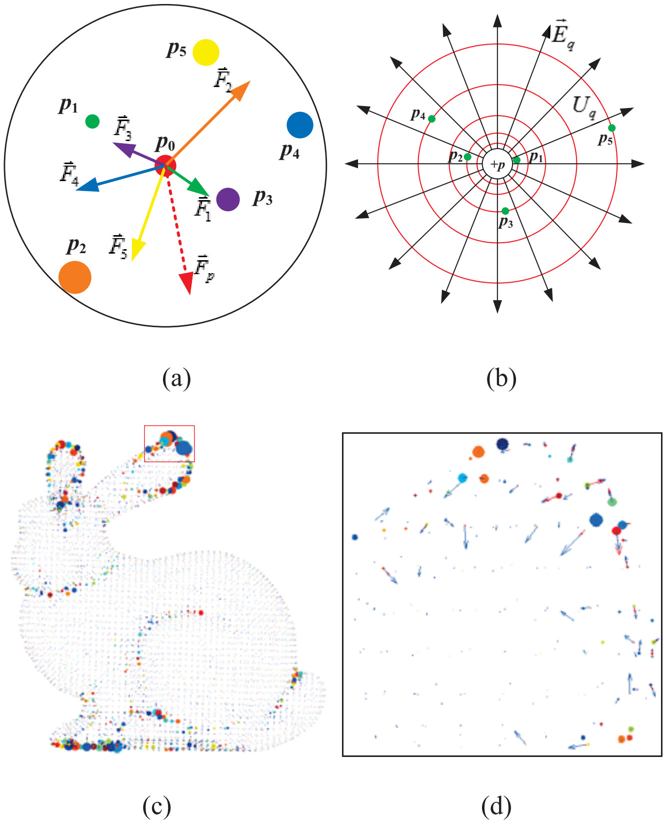
where C_{ci} is the curvature value of point c_i, and λ is the amplification coefficient, in this paper, λ is set as λ=200. As shown in Fig 1 (a), the points p_1, p_2, …, p_5 represent the neighborhood of point p_0 with different colors, and the size of each point represents the quantity of electric charge. Because the electric force satisfies the superposition principle, the electric force at p_0 is $\boldsymbol{F_{p0}= F_{p1}+ F_{p2}+ F_{p3}+ F_{p4}+ F_{p5}}$.

2.2 Electric field

According to the Coulomb's law, the distribution of charges c_1 and c_2 can create electric fields E_1 and E_2, respectively. The electric field is a vector field, and it satisfies the superposition principle also. This principle is useful to calculate the field created by multiple point charges. If charges c_1, c_2, …, c_n are stationary in space at r_1, r_2, …, r_n, the resulting field is the sum of fields generated by each particle as described by:

$$\vec{E}(r) = \sum_{i=1}^{N} \vec{E}_i(r) = \frac{1}{k_e} \cdot \sum_{i=1}^{N} q_i \frac{r - r_i}{\left|r - r_i\right|^3} \qquad (3)$$

where k_e is Coulomb's constant, and N is the number of neighbors of the point c.



(a) (b)

(c) (d)

Fig 1. (a) Electric force computation of p_0. (b) Description of electric filed and the equipotential surfaces. (c) Point description of Bunny, and (d) is the zoom-in of the local region at the ear of Bunny.

2.3 Electric potential energy

In this paper, we use the electric potential energy U_E to illustrate the relationship between the center point and its neighborhood points, respectively. The electric potential energy of one neighborhood point charge c (with $+q$ quantity) in the presence of a point charge C (with $+Q$ quantity) is:

$$U_E(d) = k_e \cdot \frac{qQ}{d} \qquad (4)$$

where d is the distance between the point charges c and C, and q and Q are the signed values of the charges. As for point description, in Fig 1 (b) we compute the electric potential energy for each neighbor of point C with $+p$ quantity, the red circles are the equipotential surfaces, and the black arrows denote the electric field lines.

In this section, we present an EFT-based point descriptor which aims at encoding the shape around a point in terms of a set of numerical values. Take Bunny for example, first, the curvature of each point is computed, and the quantity is assigned based on the curvature. Then, k nearest points are selected as the neighborhood of the given point c_i to calculate the electric force, electric filed, and electric potential energy, respectively. In this paper, we use the scalar form of the electric force and electric filed as the first two bins for point description, and the other k bins are electric potential energies. Therefore, the point can be presented by a (2+k) dimensional bins. As shown in Fig 1 (c) and (d), the red arrows are the electric force and the blue arrows denote the electric filed. As for registration, the initial corresponding point pairs are achieved by searching the points with similar properties in two point clouds.

3 Differential evolution algorithm for registration

3.1 Rigid registration problem

The purpose of rigid registration is to find a transformation matrix to match the point clouds onto a common coordinate system. Given two point clouds P and Q, where $P=\{p_i \mid i=1,2,\ldots, N_p\}$ and $Q=\{q_j \mid j=1,2,\ldots, N_q\}$, the N_p and N_q are the number of points of P and Q, respectively. Assuming that Q is fixed and the P is moving. Based on the transformation matrix $T=[T_r, T_t]$, P is updated as follows:

$$p_{i\_new} = T_r \cdot p_i + T_t \tag{5}$$

where T_r is the rotation matrix and the T_t is the translation vector. Then, a corresponding table T_c is generated in terms of the temporary location of two models. According to the Euclidean distance [11], the current error is computed as follows:

$$Er_t = \sqrt{\frac{1}{N_p} \cdot \sum_{i=1}^{N_p} \left[p_{i\_new} - T_c(q_i) \right]^2} \tag{6}$$

where $T_c(\ )$ means searching the correspondence between two models. The goal of the registration is to find a transformation matrix, which include three rotation parameters (r_x, r_y, and r_z) and three translation parameters (t_x, t_y, and t_z), the parameters with subscripts x, y, and z mean they are updated along x, y, and z axes, respectively. In this paper, we cast the registration as a six dimensional non-linear optimization problem, and the improved DE algorithm is employed to search the global optimal solution.

3.2 DE algorithm for fine registration

DE algorithm is proposed by Storn et al. [16], in the past several decades, various enhanced DE algorithms have been proposed for different applications. Because range image registration is a low-dimension optimization problem and the coarse registration is performed, in this paper, a simple and effective optimization structure of DE algorithm is preferred.

As mentioned previously, when use DE algorithm to solve the registration problem the transformation matrix [t_x, t_y, t_z, r_x, r_y, r_z] is set as the individual of population, and the real-code is employed. The objective function is to minimize the root mean square error (RMSE) with Euclidean distance between two point clouds. In practice, the mutation factor F plays a crucial role for the efficiency of the DE algorithm, and the setting of F should be problem dependent. In this paper, we introduce an adaptive mutation operator [17] into the DE algorithm as follows:

$$F = F_0 \cdot 2^{e^{\left(1-\frac{G}{G+1-G_{\max}}\right)}} \tag{7}$$

where F_0 is the mutation constant, G is the current evolutional generation, and G_{max} is the maximum evolutional generation.

As for registration, because the units of the translation and the rotation are millimeter and degree, respectively, we assign different mutation constants for translation parameters and rotation parameters. Besides, to speed up the convergence of DE algorithm the centroids of two point clouds are translated to the origin of coordinates. Furthermore, we use the approximate solutions achieved by the coarse registration for population initialization of DE, and the mutation strategy “DE/best/2” is employed.

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}) + F \cdot (x_{r3,G} - x_{r4,G}) \tag{8}$$

where the $r1$, $r2$, $r3$, and $r4$ are distinct integers randomly selected from the range [1, NP] and any of them that are not equal to i. NP is the population size, and $x_{best,G}$ represents the best individual among the current population based on the fitness value.

When use the DE for registration, the main computation time is searching the correspondence between two point clouds. In general, the corresponding point pairs are determined by the Euclidean distance of the 3D points in two models. Assuming that the number of points of P and Q are N_p and N_q, respectively, the computation complexity is $O(N_pN_q)$. In this paper, k-D tree method is employed to accelerate the computation, and the computation complexity is reduced to $O(N_p\lg(N_q))$. In the procedure of evolution, the population size of DE is set as m and the number of iterations is k, and the whole computational complexity is $O(mkN_p\lg(N_q))$.

4 Experiments and results

4.1 Experiment case

In this section, we compared the proposed registration algorithm, named FADE, with three deterministic methods (ICP [11], FICP [18] and ADF [19]) and two improved heuristic algorithms (IFFO [20] and IDE [21]).The test models are shown in Fig 2, from left to right they are Bunny, Dragon, Air Intake, and Fandisk models.

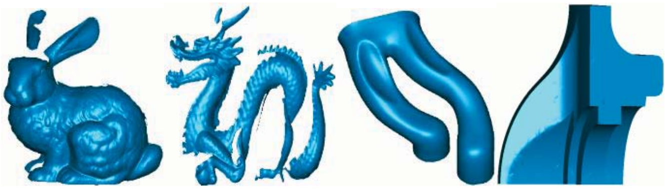


Fig 2. Test models.

The number of points of each model are 4026, 8261, 9001, and 11165, respectively. Because in registration the rotation matrix has higher effect than translation vector, we pay more attention to the rotation matrix. Following the work of Li et al. [19], we set the original point set as Q, and the moving point cloud P is achieved by the presetting transformation matrices conducted on the Q. The presetting matrices are shown in Table 1, where L is the largest size (width, height, and depth) of the model.

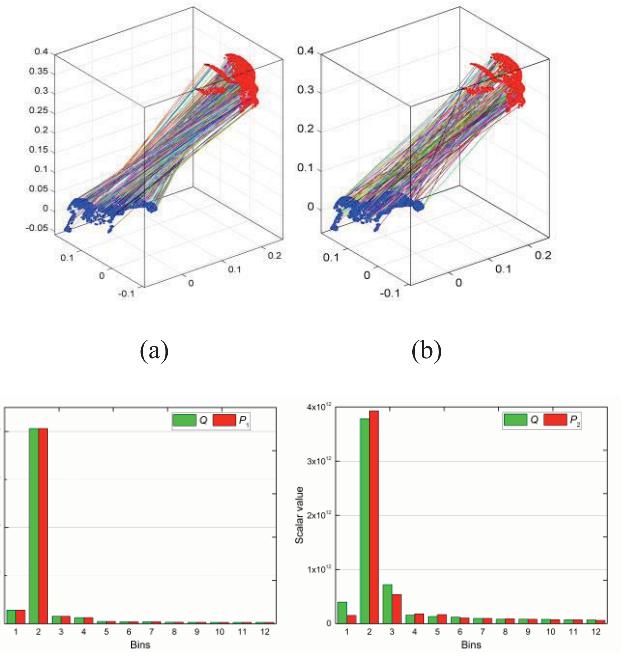
Table 1. Presetting transformation matrices.

No. of Trans	T_x (*mm*)	T_y (*mm*)	T_z (*mm*)	R_x (º)	R_y (º)	R_z (º)
T_1	0.1**L*	0.1**L*	0.1**L*	90	90	90
T_2	0.5**L*	0.5**L*	0.5**L*	120	120	120
T_3	1.0**L*	1.0**L*	1.0**L*	150	150	150

4.2 Experiments

In the coarse registration experiments, we aim to find the corresponding points between Q and $P_1=Q_T$, and Q and $P_2=Q_S$, where Q_T means the Q is transformed by the matrix T, and Q_S denotes the down-sampling of Q. In this section, we use the Bunny as the Q to register $P_1=Q_{T2}$ and $P_2=Q_{ST2}$, respectively. The number of the neighbors is set as k =10. Note that to keep the geometric structure of the model, we simplified the original point cloud based on the triangular meshes. Thus, although the number of points of P_2 is smaller than Q, the point cloud P_2 is not a real subset of Q. With 60% simplified triangular meshes of Q, the P_2 is achieved with 2446 points. The corresponding point pairs of Q-P_1 and Q-P_2 are shown in Fig 3 (a) and (b), respectively. For property bins comparison, we randomly selected one point pair from Q-P_1 and Q-P_2, respectively, and the results are shown in Fig 3 (c) and (d).

In fine registration, we use the simplified point sets to match the original point clouds. The simplified Bunny, Dragon, Air Intake, and Fandisk point sets have 2446, 4235, 4501, and 6699 points, respectively. ICP, FICP, and ADF are deterministic methods, the maximum iteration is set as $Iter_{max}$=30 for these methods. As for the IFFO, IDE, and FADE, the population size is set as NP=30, and the terminal condition in this experiment is the maximal number of function evaluations (FEs) FEs=3000. In IFFO, the maximum search radius of translation parameter is λ_t = 1.0*10-7, and the maximum search radius of rotation parameter is λ_r = 1.0*10-10. The parameters setting of IDE are follow with [21]. In FADE the initial mutation constants of translation and rotation are F_{t0}=0.2 and F_{r0}=0.4. In some papers, the initial transformation matrix is determined by only a few point pairs from the coarse registration, and if the bad point pairs are selected that will lead a false result in fine registration. Considering the effect of noise and occlusion, in this paper, we randomly select 3 different point pairs from two point clouds 2000 times to compute the transformation matrices, and the best 30 matrices are saved as the initial population of FADE. To enable a fair comparison, each heuristic algorithm is run 25 times independently and the median value is saved. The registration errors of each algorithm are shown in Table 2, where the bold values mean the best results in the rows.



(a) (b)

(c) (d)

Fig 3. (a) Corresponding point pairs of Q and P_1. (b) Corresponding point pairs of Q and P_2. (c) Property bins of Q and P_1. (d) Property bins of Q and P_2.

Table 2. Registration error comparison of each algorithm.

Models	***T***	**ICP**	**FICP**	**ADF**	**IFFO**	**IDE**	**FADE**
Bunny	T_1	1.32E-02	**7.98E-04**	1.00E-02	1.59E-02	1.17E-02	8.85E-04
	T_2	1.24E-02	1.24E-03	9.26E-03	1.49E-02	4.95E-04	**3.92E-04**
	T_3	1.60E-03	**2.92E-04**	6.17E-04	1.47E-03	9.78E-04	3.92E-04
Dragon	T_1	1.11E-02	9.59E-03	**1.19E-03**	9.14E-03	8.20E-03	8.38E-03
	T_2	1.20E-02	1.09E-03	9.75E-03	1.31E-02	8.60E-03	**3.67E-04**
	T_3	2.22E-03	9.91E-04	6.36E-03	2.00E-03	1.10E-03	**1.01E-04**
Air Intake	T_1	3.45E-01	1.33E-01	2.77E-01	3.43E-01	3.28E-01	**5.58E-02**
	T_2	3.41E-01	1.10E-02	5.97E-02	5.48E-02	4.24E-01	**8.84E-03**
	T_3	2.77E-01	3.31E-01	5.97E-02	5.33E-02	3.78E-02	**9.45E-03**
Fandisk	T_1	3.34E-01	1.09E-01	3.99E-01	3.31E-01	3.51E-02	**9.35E-03**
	T_2	6.43E-01	3.64E-01	3.99E-01	4.27E-03	1.33E-04	**2.69E-05**
	T_3	5.21E-02	2.30E-03	2.34E-02	2.09E-03	**1.47E-05**	3.06E-05

4.3 Discussion

In this section, we illustrated the performance of EFT-based point descriptor for point clouds coarse registration. From Fig 3 (a) we can see that all the points in Q can find a right corresponding point in P_1, and the matched property bins of the corresponding points were shown in Fig 3 (c). This illustrated that the EFT-based point descriptor was invariant to 3D rigid transformations. Fig 3 (b) and (d) presented a snapshot of corresponding point pairs of two point clouds with different sizes, and some false correspondence were achieved on the low curvature regions of the model. That’s because the bins of the free charge with low electric quantity were sensitive to its neighbors.

In fine registration, we conducted experiments with complex initial positions. As shown in Table 2, the deterministic methods were sensitive to the initial position and the geometry of the models. ICP and ADF methods achieved successful results for Bunny and Dragon models with simple case T_1, however, they were stick in local minima with the Air Intake and Fandisk models. Relatively, the FICP performed better than ICP and ADF. As for heuristic algorithms, the results gained by IDE were more accurate than IFFO, however, the structure of IDE was more complicated than IFFO. It was found from our experiments that FADE performed best for the test models under current initial positions, benefit from the coarse registration it was more robust to the initial position and the geometry of the models.

5 Conclusion

In this paper, we aim to obtain the most possible accurate alignment of two models with complex initial position, and a coarse-to-fine method is proposed for range image registration. In coarse registration, a novel point descriptor is designed based on electrostatic field theory, and the initial correspondence is achieved to reduce the search space. In addition, combine with the coarse registration, an improved differential evolution algorithm is presented for fine registration. The extensive comparison studies reveal that the proposed point descriptor is invariant to 3D rigid transformations for coarse registration, and the FADE performs superior to many different existing algorithms used to solve the fine registration problem.

The limitations of this paper are as follows. First, although the proposed fine registration method is accurate and more robust, the heuristic algorithms take more computation time than the deterministic methods. Furthermore, in special situation, if the false corresponding point pairs take a greater proportion, they are different to be removed in terms of random selection.

In the future, other properties of the electrostatic field can be considered for point description, such as the electric flux and the charge density. Besides, in coarse registration, a novel point detection strategy can be developed to remove the false point pairs. Furthermore, to improve the efficiency of registration, deterministic methods can be employed to combine with the improved EFT-based point descriptor.

Acknowledgment

This project is supported by National Natural Science Foundation of China (Grant No. 51435009, 51421062, and 51575212). The authors would also like to thank Dr. Lixin Tang for the IDE code providing.

6 References

[1] P. Biradar and S.S. Pande. “Efficient algorithms for automated inspection of freeform surfaces”; Procedia Manufacturing, vol. 1, pp. 35-46, 2015.

[2] L. Gomes, L. Silva, and O.R.P. Bellon. "3D reconstruction methods for digital preservation of cultural heritage: A survey"; Pattern Recognition Letters, vol. 50, pp. 3-14, 2014.

[3] J. Chen, X. Wu, M.Y. Wang, and X. Li. "3D shape modeling using a self-developed hand-held 3D laser scanner and an efficient HT-ICP point cloud registration algorithm"; Optics & Laser Technology, vol. 45, no. 1, pp. 414-423, 2013.

[4] S. Du, N. Zheng, L. Xiong, S. Ying, and J. Xue. "Scaling iterative closest point algorithm for registration of m-D point sets"; Journal of Visual Communication & Image Representation, vol. 21, no.5, pp. 442-452, 2010.

[5] Y. Diez, F. Roure, X. LLADO, and J. Salvi. "A qualitative review on 3D coarse registration methods"; ACM Computing Surveys, vol. 47, no. 3, pp, 1-36, 2015.

[6] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino. "3D geometric scale variability in range Images: features and descriptors"; International Journal of Computer Vision, vol. 99, no. 2, pp. 232-255, 2012.

[7] B. He, Z. Lin, and Y.F. Li. "An automatic registration algorithm for the scattered point clouds based on the curvature feature"; Optics & Laser Technology, vol. 46, no. 1, pp. 53-60, 2013.

[8] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz. "Aligning point cloud views using persistent feature histograms"; IEEE International Conference on Intelligent Robots and Systems, pp. 3384-3391, 2008, Nice, France.

[9] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. "Rotational projection statistics for 3D local surface description and object recognition"; International Journal of Computer Vision, vol. 105, no. 1, pp. 63-86, 2013.

[10] J. Yang, Z. Cao, and Q. Zhang. "A fast and robust local descriptor for 3D point cloud registration"; Information Sciences, vol. 346, pp. 163-179, 2016.

[11] P.J. Besl and N.D. Mckay. "A method for registration of 3-D shapes"; IEEE transactions on pattern analysis and machine intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[12] C.C. Queirolo, S. Luciano, O.R.P. Bellon, and P.S. Mauricio. "3D face recognition using simulated annealing and the surface interpenetration measure"; IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 2, pp. 206-219, 2010.

[13] K.C. Chi, H.T. Tsui, and L. Tong. "Surface registration using a dynamic genetic algorithm"; Pattern Recognition, vol. 37, no. 1, pp. 105-117, 2004.

[14] A. Valsecchi, S. Damas, J. Santamaría, and L. Marrakchi-Kacem. "Intensity-based image registration using scatter search"; Artificial Intelligence in Medicine, vol. 60, no. 3, pp. 151-163, 2014.

[15] J. Santamaría, O. Cordón, and S. Damas. "A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling"; Computer Vision & Image Understanding, vol. 115, no. 9, pp. 1340-1354, 2011.

[16] R. Storn and K. Price. "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces"; Journal of Global Optimization, vol. 11, no. 4, pp. 341-359, 1997.

[17] X. F. Yan, J. Yu, and F. Qian. "Adaptive mutation differential evolution algorithm and its application to estimate soft sensor parameters"; Control Theory & Applications, vol. 23, no. 5, pp. 744–748, 2008.

[18] D.J. Kroon. "Segmentation of the mandibular canal in cone-beam CT data"; University of Twente, PhD thesis, 2011.

[19] W.L. Li, Z.P. Yin, Y.A. Huang, and Y.L. Xiong. "Three-dimensional point-based shape registration algorithm based on adaptive distance function"; IET Computer Vision, vol. 5, no. 1, pp. 68-76, 2011.

[20] Q.K. Pan, H.Y. Sang, J.H. Duan, and L. Gao. "An improved fruit fly optimization algorithm for continuous function optimization problems"; Knowledge-Based Systems, vol. 62, no. 5, pp. 69-83, 2014.

[21] L. Tang, Y. Dong, and J. Liu. "Differential evolution with an individual-dependent mechanism"; IEEE Transactions on Evolutionary Computation, vol. 19, no. 4, pp. 560-574, 2015.

Efficient Noise-free Image Acquisition of Moving Objects in a Complex Background using CUDA

Iakov Korovin[1], Maxim Khisamutdinov[1], and Gerald Schaefer[2]
[1]Southern Federal University, Rostov-on-Don, Russia
[2]Department of Computer Science, Loughborough University, U.K.

Abstract— *In this paper, we propose an algorithm to process a sequence of images of a moving object in a complex background to obtain a single noise-free image of this object. We present an efficient software implementation of the algorithm that is based on multi-thread processing on a CUDA architecture. Experimental results on astronomical image sequences confirm high quality resulting images and a significant speed-up in terms of processing time.*

Keywords: Image noise, noise removal, astronomical imaging, CUDA.

1. Introduction

Formation of a noise-free image of a moving object in a complex background is an important problem that arises in various areas such as astronomy, aviation, medicine, and others. The problem implies processing of a single image or a sequence of images for generation of a single noise-free image.

Common noise reduction algorithms [1], [2], typically intended for processing of a single image, usually lead to an improved peak signal-to-noise-ratio (PSNR) and thus improved image quality. To estimate the quality of such algorithms, noise, commonly white Gaussian noise, is added to noise-free images. Then, after applying a selected algorithm, the obtained result image is compared with the initial one, with PSNR often the metric of choice [3], [4].

When developing a method that is based on analysis of an image series captured from one angle of a photodetector, it is reasonable to consider a combination of several images. However, it is necessary to take into account how exactly the images are being obained as the objects in separate images can be displaced and do not coincide leading to the problem of video stabilisation [5]. We thus need to take into account the movement of the specified object against the background.

To address this problem, in this paper, we propose an algorithm for formation of a single noise-free image of a moving object in a complex background that comprises of two stages:

- alignment of the images of the series;
- combination of several images to obtain a single noise-free image of the moving object.

In particular, our implementation of the algorithm is realised on a CUDA [6] architecture to allow for efficient processing.

2. Proposed Algorithm

A summary of our proposed algorithm is given in Fig. 1.

In the first stage of the algorithm, the images of the series are aligned. For this, the user can select the area of the required object in a reference frame $frame[ref]$. The remainder of the processing is then completely automatic. Based on analysis of the correlation matrix, a motion matrix T is obtained, which contains motion data between the reference image $frame[ref]$ and the other images in the series $frame[t]$. Since for astronomical imaging an exact definition field of rotation is difficult to determine due to the small scale and blurring effects, a rotation matrix R is obtained by analysing the correlation of key points of $frame[ref]$ and $frame[t]$. All images of the series $frame[t]$ are matched with $frame[ref]$ using the displacement T of the area of the selected object and rotation R.

The second stage of our proposed algorithm is the combination of the set of images matched during the first stage. The simplest solution would be to perform averaging or median processing. However, the final image would then include background signals which are not desired as they would result in artefacts. In this paper, we therefore employ a modified median averaging where the modification uses of part of the series images for each pixel of the final image $frame\_res[i][j]$. An array of image brightness M of the pixels of all images is generated for each pixel. Then, the array items are sorted, and the first $k = n - 2$ values, where n is the total number of images in the sequence, are averaged. Hence, the approach uses $(n - 2)/n$ data points to remove the background noise. Clearly, k can be selected experimentally; the value $k = n - 2$ was selected for the astronomical problem of recognition of a comet against the background of stars.

In our software implementation, which is capable of processing a sequence of images of the moving object in FITS (Flexible Image Transport System) format [7], we use multi-thread loading on a CUDA architecture (see Fig. 2) for improved efficiency.

Since the newer versions of the CUDA platform (v.4.0 and above) support the selection of one CUDA context

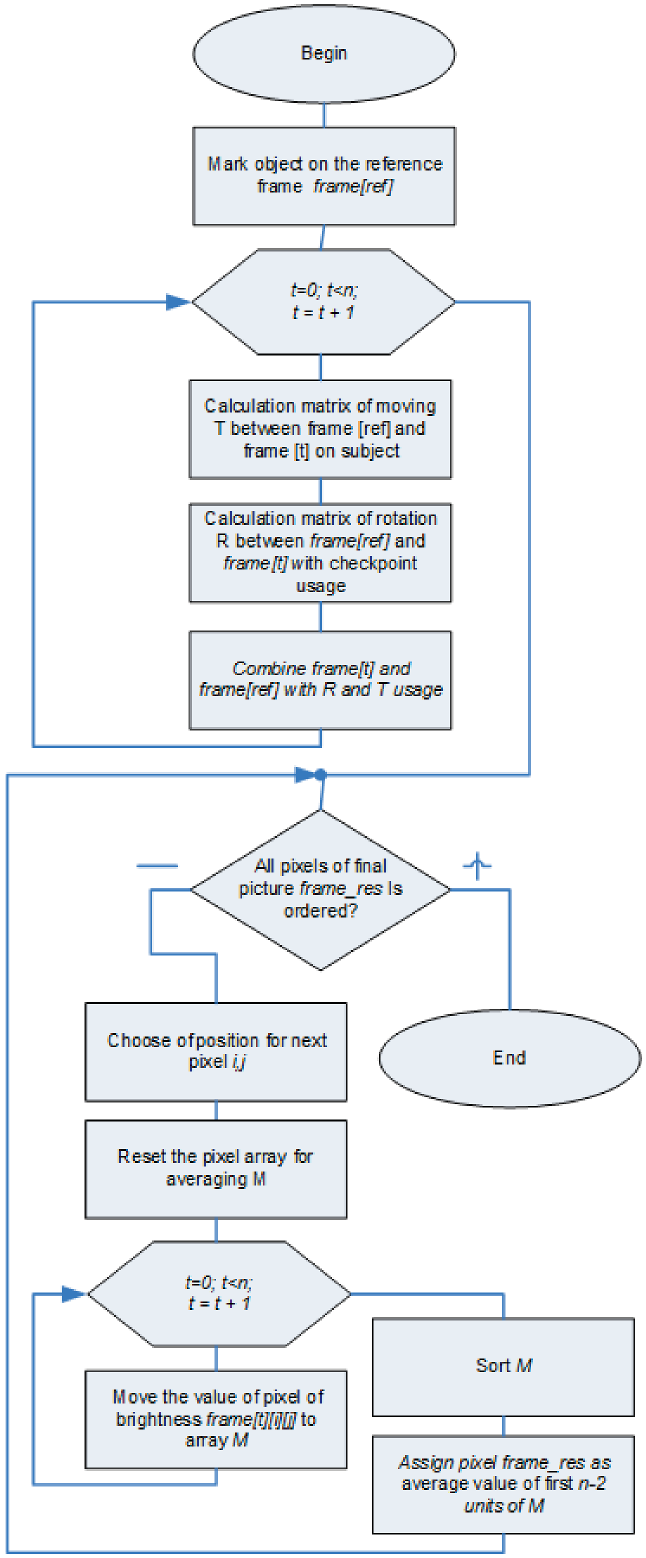

Begin
Mark object on the reference frame frame[ref]
t=0; t<n; t = t + 1
Calculation matrix of moving T between frame [ref] and frame [t] on subject
Calculation matrix of rotation R between frame[ref] and frame[t] with checkpoint usage
Combine frame[t] and frame[ref] with R and T usage
—
All pixels of final picture frame_res Is ordered?
+
Choose of position for next pixel i,j
End
Reset the pixel array for averaging M
t=0; t<n; t = t + 1
Move the value of pixel of brightness frame[t][i][j] to array M
Sort M
Assign pixel frame_res as average value of first n-2 units of M

Fig. 1: Our proposed algorithm for obtaining of a noise-free image of a moving object in a complex background.

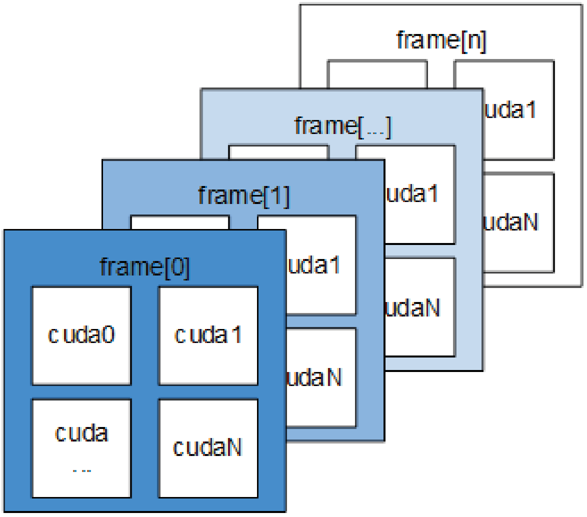


Fig. 2: Multi-thread loading on CUDA architecture.

for one process in the system, there are difficulties when implementing a program for multi-threaded load on CUDA. To overcome this limitation, one can use the CUDE API and the creation of in-memory virtual CUDA contexts corresponding to one thread. This approach allows to remove the restriction on CUDA context selection for the process, and each stream will have its own CUDA context. A limitation here is the amount of available memory of the computing device. For our problem, with 8GB of RAM we can create 8 CUDA contexts, taking into consideration the memory load according to the system needs.

The basic idea of our approach is to load a group of images into video memory and process the fragments in separate threads so that the image area is divided equally between the threads. The number of threads ($cuda1 \ldots cudaN$) is selected experimentally in accordance with the video card and the amount of available memory; the recommended load is less than 80% of GPU load. Here, the main indicator should be the GPU load, rather than a limit on core temperature. In the BIOS of video cards, the core temperature limit is used to activate the protection mechanism (throttling) and is typically in the range of 105-140°C. Monitoring the temperature during the execution of 3D applications shows that core temperatures near 90°C on Nvidia GPUs do not cause problems.

It should be noted that the temperature measuring systems used in the graphics cards are typically inaccurate. First, only in the NV43 kernel and later versions is the temperature monitoring systems fully integrated into the chip, while in earlier GPUs there was a dependence on external components that could lead to errors. Second, bit ADC, used in most temperature monitoring systems, is inadequate for measuring with high precision and the actual temperature values are obtained using interpolation or tables with correction factors. Third, the tables of recalculation themselves often

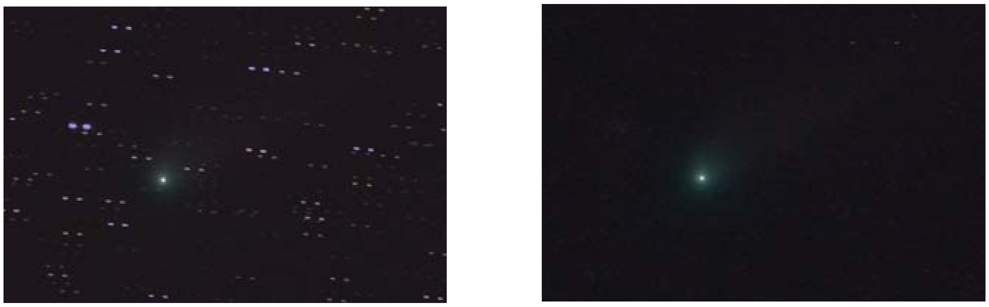


Fig. 3: Processing results of astronomical image sequence of a comet: median (left) and our proposed algorithm (right).

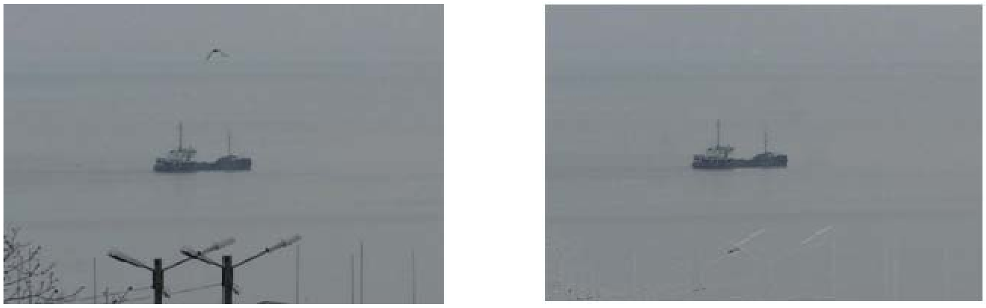


Fig. 4: Processing results of image sequence of a moving ship: median (left) and our proposed algorithm (right).

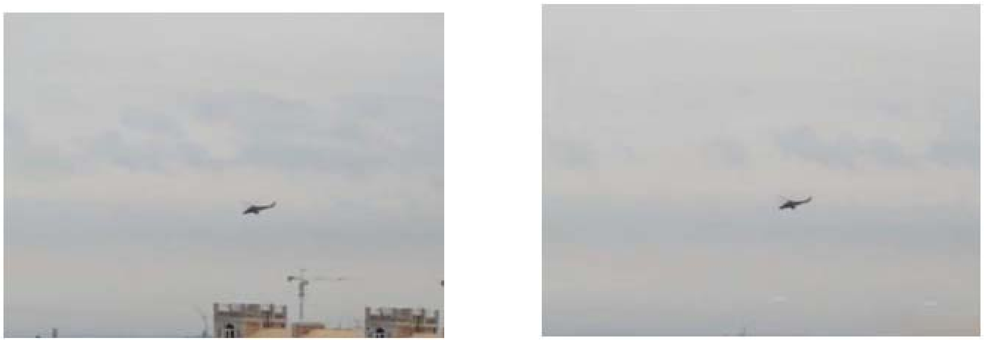


Fig. 5: Processing results of image sequence of a moving helicopter: median (left) and our proposed algorithm (right).

contain errors. Thus, Nvidia 7x.xx drivers give temperature values that are inflated by 20% for GeForce 6x00 cards, although this error has been fixed for versions 71.89 to 76.45. Errors were also present in many versions of the ASUS SmartDoctor application, which led to unrealistically high values reported for core temperature on the card series V9999. In addition, the BIOS of the first cards, based on the G92 GPU, did not use conversion tables reading thermal diode at all, leading to errors reported in the ASUS SmartDoctor utility by about ten degrees.

Consequently, high temperatures reported by drivers and utilities are not necessarily a sign of overheating. In contrast, image distortion and stops in video streaming can indicate problems with the graphics card. For GPU load estimation, it is recommended to use MSI Afterburner[1] or similar utilities.

3. Experimental Results

In our experiments, we used an Nvidia GeForce 780GT card, set the number of threads from 1 to 8, while the input data are 2 to 64 images of the comet C/2013 US10 (Catalina), with each image having a resolution of 4288×2848 pixels.

For visual comparison of a simple median approach and our suggested algorithm, combination of two frames of the sequence was performed and the results are shown in Fig. 3. We can clearly see, that our proposed approach is capable of completely removing the background objects and gives a clear image of the moving object, i.e. the comet.

While we have primarily developed our algorithm for processing sequences of astonomical image sequences, its use is not restricted to astronomical imaging. In Fig. 4 and Fig. 5, we show results of processing an image sequence

[1] https://gaming.msi.com/features/afterburner

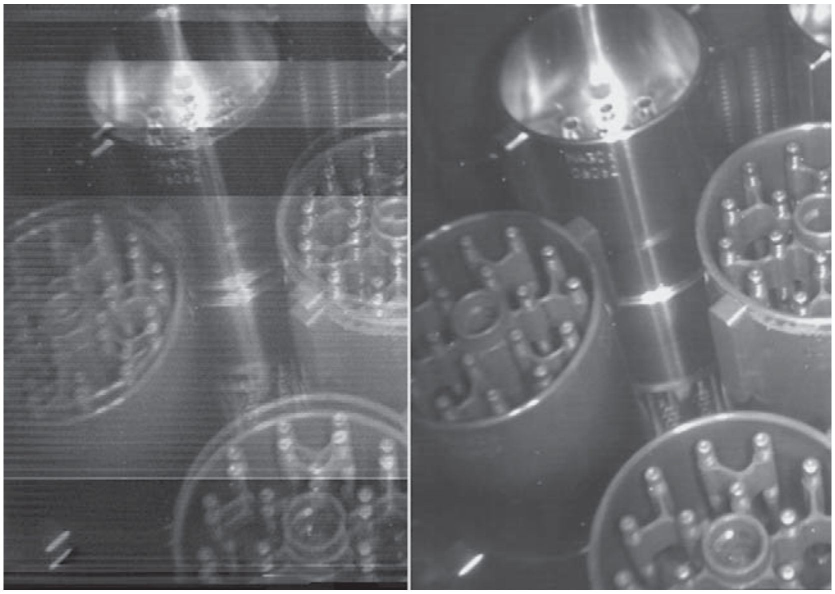


Fig. 6: Processing results of image sequence of nuclear fuel assemblies in atomic power plant: median (left) and our proposed algorithm (right).

Table 1: CUDA processing time [s] results.

frames \ threads	**1**	**2**	**4**	**8**
2	1.20	0.80	0.45	0.22
4	3.40	2.30	1.20	0.55
8	5.90	3.30	1.59	0.79
16	12.80	7.10	3.80	1.70
32	24.50	13.50	7.60	3.40
64	97.30	51.80	26.10	13.50

of a moving ship and a moving helicopter respectively. It is evident that our approach is capable of clearly extracting the objects of interest while blending out background objects. Fig. 6 shows an example of nuclear fuel assemblies in an atomic power plant. Again, we can observe vastly improved image quality compared to standard median processing.

To measure the processing time of our software implementation based on CUDA multi-thread programming, we performed the following test. We vary the number of threads from 1 to 8, and the number of input images from 2 to 64. If we use 1 thread and 64 images, then the loading of the graphic processor is less than 10%, while if we use 8 threads and 64 images the loading is 80%.

Table 1 shows the obtained results in terms of measured processing times. As we can see, employing a multi-threaded CUDA implementation leads to a clear reduction of the time required for processing the image sequences.

4. Conclusions

In this paper, we have presented an approach to processing an image sequence to deliver a single noise free image. To do so, we first align the object of interest across several frames, followed by selective combination of pixel values to remove noise. By applying CUDA technology, we optimise the algorithm's processing time through multi-thread processing on GPUs. Our experimental evaluation confirmed significantly improved efficiency, coupled with improved image quality compared to a standard median filtering algorithm. In future work, we will look at developing and modifying our approach for application to similar DSP tasks.

References

[1] W. K. Pratt, *Digital image processing*, Wiley, p. 807, 2007.

[2] R. S. Gonzalez, *Digital Image Processing*, Prentice Hall, p. 954, 2002.

[3] O. I. Shelukhin, *Estimation of quality of stream video transmission in telecommunication networks with software and hardware tools application*, O. I. Shelukhin, Yu. A. Ivanov, *Electric technical and informational complexes and systems*, #4, pp. 48-56, 2009.

[4] Ia. S. Korovin, M. V. Ksisamutdinov, *The method of obtaining a noisy image based on the video sequence processing*, Computer Optics, v. 38, #1, pp. 112-117, 2014.

[5] B. D. Lucas, T. Kanade, *An iterative image registration technique with an application to stereo vision.* Imaging Understanding Workshop, pp. 121–130, 1981.

[6] T. Valich, *nVidia Launches CUDA Toolkit 3.0, expands OpenCL.*

[7] http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

[8] D. I. Kleopatrov, A. A. Frenkel, *Forecast of economic indexes using the method of simple exponential smoothing. – Statistical analysis of economical time series and forecasting*, Moscow, Nauka, p. 298, 1973.

[9] I. L. Legostayeva, A. N. Shiryaev, *Minimum weights in the problem of determination of stochastic process trend.* Probability theory and its application, Vol. XVI, #2, 1971.

[10] R. G. Brown, *Smoothing, Forecasting and Prediction*. Prentice-Hall, 1963.

MD-LBPV Texture Analysis for Nailfold Capillaroscopy Image Classification

Niraj P. Doshi[1], **Gerald Schaefer**[2] **and Iakov Korovin**[3]
[1]dMacVis Research Lab, India
[2]Department of Computer Science, Loughborough University, U.K.
[3]Southern Federal University, Russia

Abstract—*Nailfold capillaroscopy (NC) is a non-invasive imaging technique employed to assess the condition of blood capillaries in the nailfold and is particularly useful for diagnosis of scleroderma spectrum disorders and Raynaud's phenomenon. Diagnosis is based on the identification of particular scleroderma patterns in the images which are typically grouped into early, active and late patterns. In this paper, we present a computer vision approach to recognising scleroderma patterns in NC images. Following a preprocessing step to enhance image quality, we extract texture information in a holistic way rather than trying to extract and measure individual capillaries. As texture features we employ multi-dimensional LPB variance descriptors which capture multi-resolution texture and local contrast information. Our experimental results confirm our approach to work well and to outperform an earlier approach.*

Keywords: Medical imaging, nailfold capillaroscopy, texture, LBP, MD-LBPV.

1. Introduction

Nailfold capillaroscopy (NC) is a non-invasive imaging technique employed to assess the condition of blood capillaries in the nailfold. It is particularly useful for early detection of scleroderma spectrum disorders [1] and evaluation of Raynaud's phenomenon [2]. Diagnosis using NC images involves the classification into Early, Active and Late groups, also known as NC patterns or scleroderma (SD) patterns [3], [4] (see Fig. 1 for example images) based on the identification of enlarged or giant capillaries, haemorrhages, loss of capillaries, disorganisation of the vascular array, and ramified/bushy capillaries in the images [5].

While diagnosis based on NC is typically performed by manual inspection, computerised nailfold capillaroscopy can help to reduce the inherent ambiguity in human judgement while greatly reducing the time for diagnosis [6]. However, unfortunately the literature on computer aided approaches to NC image analysis is relatively sparse. Existing approaches [7], [8], [9], [10], [11] typically aim to segment capillaries and analyse the extracted structures.

In contrast, in [12] we have proposed a novel holistic approach for analysing NC images using texture analysis. In particular, we have shown that local binary pattern variance (LBPV) [13] texture features can be successfully employed to distinguish between different NC patterns. In this paper, we build upon this approach and show that by extracting these features in a multi-scale fashion while at the same time maintaining information between the scales, improved recognition performance can be achieved.

2. Scleroderma Patterns

In healthy subjects, the capillaries observed at the nailfold are fairly homogeneous in terms of size and shape and are regularly arranged. However, in patients with scleroderma as well as some other connective tissue diseases, abnormalities manifest themselves which can be identified using nailfold capillaroscopy.

The degree of these abnormalities indicates the severity and progression of the disease. Three NC patterns can be defined and characterised by [4]:

- **Early (E):** few giant capillaries, few capillary haemorrhages, relatively well preserved capillary distribution, no evident loss of capillaries.
- **Active (A):** frequent giant capillaries, frequent capillary haemorrhages, moderate loss of capillaries with some avascular areas, mild disorganisation of the capillary architecture, absent or some ramified capillaries.
- **Late (L):** irregular enlargement of the capillaries, few or absent giant capillaries, absence of haemorrhages, severe loss of capillaries with large avascular areas, severe disorganisation of the normal capillary array, frequent ramified/bushy capillaries.

3. Multi-dimensional LBPV Texture Features

3.1 Local binary patterns

Local binary patterns (LBP) are simple yet effective texture descriptors. The original LBP variant [14] operates on a per-pixel basis, and describes the 8-neighbourhood pattern of a pixel in binary form. If $\{g_1, g_2, \ldots, g_8\}$ is the set of 8-neighbourhood pixels of a centre pixel g_c, then the neighbouring pixels are set to 0 and 1 respectively by thresholding them with the centre pixel value. An LBP

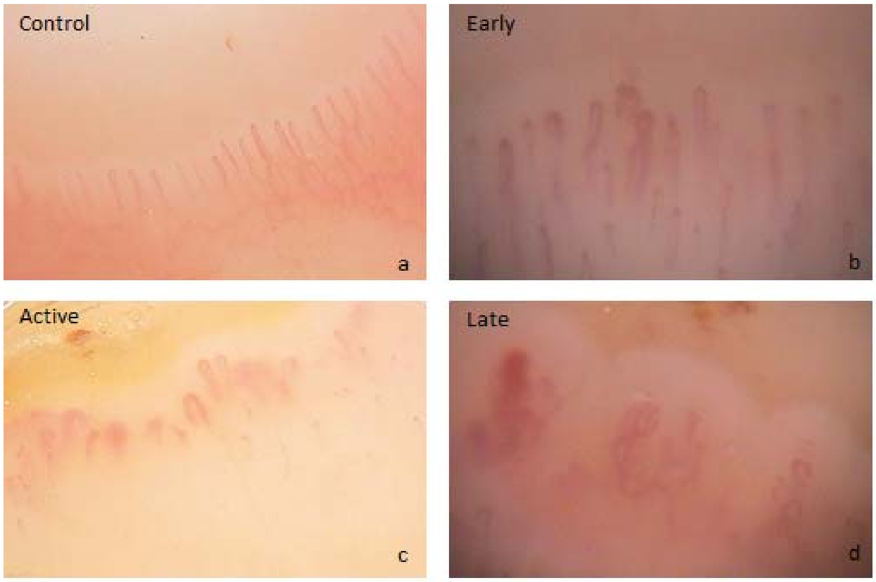


Fig. 1: NC image samples: (a) healthy subject; (b) early SD pattern; (c) active SD pattern; (d) late SD pattern.

pattern is thus obtained by

$$\text{LBP} = \sum_{p=1}^{8} s(g_p - g_c)2^{p-1}, \tag{1}$$

where

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \tag{2}$$

The 256 possible resulting patterns are then typically used to build a histogram, which serves as a texture feature for an image or an image region.

Alternatively, a circular neighbourhood can be employed [15] by R and P, where R defines the distance of the neighbours to the centre, and P is the number of samples at that distance that are employed as neighbours. Locations that do not fall exactly at the centre of a pixel are obtained through interpolation.

Rotation invariance can be easily addressed in LBP. If a texture is rotated, essentially the patterns (that is the 0s and 1s around the centre pixel) rotate with respect to the centre. Rotation invariant LBP codes, $\text{LBP}^{ri}_{P,R}$, can be thus be obtained by grouping together corresponding rotated LBP patterns [15].

LBP patterns can also be grouped based on the number of spatial transitions from 0s to 1s and vice versa in the bit pattern. Certain binary patterns are fundamental properties of texture and sometimes their frequency exceeds 90%. These patterns are called uniform [15], leading to $\text{LBP}^{u}_{P,R}$, and are defined by a uniformity measure which corresponds to the number of spatial transitions in the LBP code.

Clearly, rotation invariant and uniform patterns can be combined, leading to $\text{LBP}^{riu2}_{P,R}$. For eight neighbours, there are nine rotation invariant uniform LBP codes, two without any 0-1 changes (i.e., one with all 0s and one with all 1s) and the remaining seven with $1, \ldots, 7$ ones in sequence. It has been shown [15] that focussing on these uniform patterns while aggregating all other (i.e., non-uniform) patterns into one group leads to improved texture descriptors. While LBP generates 256 patterns for an 8-neighbourhood, and LBP^{ri} generates 36 patterns, LBP^{riu2} results in 10 pattern classes for the same neighbourhood.

3.2 LBP Variance

The contrast in an image

$$\text{VAR}_{P,R} = \frac{1}{P}\sum_{p=0}^{p-1}(g_p - \mu)^2 \tag{3}$$

with $\mu = \frac{1}{P}\sum_{p=0}^{P-1} g_p$ can be incorporated with $\text{LBP}_{P,R}$ to generate a joint distribution of $\text{LBP}_{P,R}$/$\text{VAR}_{P,R}$ which gives a powerful texture descriptor as it contains both local pattern and local contrast information. An alternative is the use of a hybrid scheme, LBP variance (LBPV) [13], which also captures joint LBP and contrast information but where the variance $\text{VAR}_{P,R}$ is used as an adaptive weight to adjust the contribution of the LBP code in histogram calculation.

LBPV histograms are calculated as

$$\text{LBPV}_{P,R}(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} \omega(\text{LBP}_{P,R}(i,j), k), \tag{4}$$

with

$$\omega(\text{LBP}_{P,R}(i,j), k) = \begin{cases} \text{VAR}_{P,R}(i,j) & \text{if } \text{LBP}_{P,R}(i,j) = k \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

and $k \in [0, K]$ defining the various LBP codes.

3.3 Multi-dimensional LBPV

By using several radii around a pixel, multiple concentric neighbourhood LBP codes can be extracted [15]. Such multi-resolution LBP features provide powerful texture descriptors. Clearly, this principle can also be applied to LBPV. In fact, in [16], multi-resolution LBPV was shown to give the best classification results on texture datasets captured under varying rotation and varying illumination and to outperform more than 30 other LBP-based texture features.

When recording multi-resolution texture information using LBPV, a histogram is generated for each scale/radius, while the histograms are concatenated to form a one-dimensional feature vector. In [17], it was shown that storing multi-scale LBP features in such a fashion leads to a loss of information between the different scales and added ambiguity. The joint distribution of LBP codes at different scales can be preserved by building a multi-dimensional LBP (MD-LBP) histogram [17]. To do so, LBP codes are calculated at different scales while the combination of the codes identifies the histogram bin that is incremented.

A similar approach can be devised to obtain multi-dimensional LBPV (MD-LBPV) texture descriptors, which incorporate image variance information as adaptive weights to build multi-dimensional LBP histograms [18].

MD-LBPV histograms are calculated as

$$\text{MD-LBPV}_{P,R=\{r_1,r_2,\ldots,r_n\}}(k_1,k_2,\ldots,k_R) = \sum_{i=1}^{N}\sum_{j=1}^{M}\omega(\text{LBP}_{P,R=\{r_1,r_2,\ldots,r_R\}}(i,j),k_1,k_2,\ldots,k_R), \quad (6)$$

with

$$\omega(\text{LBP}_{P,R=\{r_1,r_2,\ldots,r_R\}}(i,j),k_1,k_2,\ldots,k_R) = \begin{cases} f(\mathcal{V}) & \text{if } \text{LBP}_{P,R=r_s}(i,j)=k_s \quad \forall s\in\{1,2,\ldots,R\} \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

and

$$\mathcal{V} = \{\text{VAR}_{P,r_1}(i,j),\text{VAR}_{P,r_2}(i,j),\ldots,\text{VAR}_{P,r_R}(i,j)\}. \quad (8)$$

MD-LBPV based on 2 radii will hence yield a 2-dimensional histogram, MD-LBPV based on 3 radii a 3-dimensional one and so on.

While in MD-LBP the histogram is always incremented in unit values and local contrast information is not utilised, in MD-LBPV local contrast is integrated into the way multi-dimensional texture histograms are generated. Of the various ways of how variance information can be incorporated into MD-LBPV histograms [18], we chose the maximum variance method which uses the maximum value of variance over all scales

$$f(\mathcal{V}) = \max\{\text{VAR}_{P,r_1}(i,j),\text{VAR}_{P,r_2}(i,j),\ldots,\text{VAR}_{P,r_R}(i,j)\}. \quad (9)$$

4. MD-LBPV-based NC Image Analysis

Since NC images are rather challenging due to image noise, dust on lenses, micro-motion of fingers and air bubbles in the immersion oil, we first pre-process the images using a bilateral enhancer filter [19], which we have previously shown to be suitable for NC image enhancement [20].

After the image is enhanced, MD-LBPV features are extracted from the image. For decision making, first each finger is classified, followed by aggregating the outcomes for a patient.

For finger classification, we employ a standard support vector machine (SVM) classifier [21]. Since, we have more than two classes (control, early, active, and late), we employ a one-against-one multi-class SVM [22], where for each SVM, we use a linear kernel, and optimise the cost parameter $C \in [-1.1; 3.1]$ using a cross validation approach [23].

The final result is then obtained by a voting mechanism where we select the majority class of the finger classifications. Should none of the classes have the majority (i.e. there is a tie between classes) then we reject the diagnosis rather than randomly assigning it to one of the classes. A rejected case should hence be manually inspected.

5. Experimental Results

Experiments were carried out on a dataset of 12 subjects with NC images captured for three to four fingers for each patient and three patients for each class (i.e. control, early, active, and late). The images were obtained at the Dermatology Unit, Clinical Hospital of Chieti, following their standard protocol. A ground truth for all patients was also obtained by manual inspection approved by a consultant. For evaluation, a standard leave-one-out cross validation on a patient basis is performed. That is, the classifier is trained on all but one subject for which the test is run, and the procedure is repeated for all patients (i.e. 12 times in total).

The results are given in Table 1, both in terms of finger and patient classification. From there we can see that our proposed approach does indeed afford very good performance. The finger classification accuracy is 73.17% while for 10 out of 12 patients the correct SD pattern has been identified giving a patient classification accuracy of 83.33%.

For comparison, we give, in Table 2 the results using multi-scale LBPV, that is essentially the method from [12]. It is apparent that here more misclassifications occur, both for finger and patient classification. Overall, only for 8 patients a correct classification is obtained.

6. Conclusions

In this paper, we have presented a holistic method for automatic identification of scleroderma patterns in nailfold capillaroscopy images. Rather than identifying individual capillaries, we perform texture analysis coupled with a

	finger 1	finger 2	finger 3	finger 4	patient
Control 1	C	C	C	-	C
Control 2	C	C	C	-	C
Control 3	C	C	C	C	C
Early 1	E	E	E	-	E
Early 2	E	E	L	E	E
Early 3	E	C	C	-	C
Active 1	L	L	A	-	L
Active 2	L	A	A	-	A
Active 3	C	A	A	E	A
Late 1	L	L	A	-	L
Late 2	C	E	L	L	L
Late 3	L	L	L	L	L

Table 1: Classification results for nailfold capillary analysis using MD-LBPV$^{riu2}_{R=1,3}$

	finger 1	finger 2	finger 3	finger 4	patient
Control 1	C	C	C	-	C
Control 2	C	A	A	-	A
Control 3	C	C	C	C	C
Early 1	E	E	E	-	E
Early 2	E	E	L	E	E
Early 3	E	C	C	-	C
Active 1	A	A	A	-	A
Active 2	L	L	L	-	L
Active 3	C	A	C	A	reject
Late 1	L	L	L	-	L
Late 2	C	E	L	L	L
Late 3	L	L	L	L	L

Table 2: Classification results for nailfold capillary analysis using LBPV$^{riu2}_{R=1,3}$

classification approach. In particular, we extract, following an image enhancement step, multi-dimensional LBP variance (MD-LBPV) features from capillary regions to capture multi-resolution texture information. Experimental results confirm the efficacy of our method and improved classification accuracy compared to an earlier approach.

References

[1] W. Grassi, P. D. Medico, F. Izzo, and C. Cervini, “Microvascular involvement in systemic sclerosis: Capillaroscopic findings,” *Seminars in Arthritis and Rheumatism*, vol. 30, no. 6, pp. 397–402, 2001.

[2] M. Cutolo, W. Grassi, and M. Matucci Cerinic, “Raynaud’s phenomenon and the role of capillaroscopy,” *Arthritis & Rheumatism*, vol. 48, no. 11, pp. 3023–3030, 2003.

[3] H. Maricq and C. LeRoy, “Patterns of finger capillary abnormalities in connective tissue disease by wide-fieldĂİ microscopy,” *Arthritis & Rheumatism*, vol. 16, no. 5, pp. 619–628, 1973.

[4] M. Cutolo, A. Sulli, C. Pizzorni, and S. Accardo, “Nailfold videocapillaroscopy assessment of microvascular damage in systemic sclerosis,” *The Journal of Rheumatology*, vol. 27, pp. 155–60, 2000.

[5] M. Cutolo, C. Pizzorni, and A. Sulli, “Capillaroscopy,” *Best Practice and Research Clinical Rheumatology*, vol. 19, no. 3, pp. 437–452, 2005.

[6] N. P. Doshi, G. Schaefer, and K. Howell, “A review of computerised nailfold capillaroscopy,” in *17th Annual Conference in Medical Image Understanding and Analysis*, 2013.

[7] M. Paradowski, U. Markowska-Kaczmar, H. Kwasnicka, and K. Borysewicz, “Capillary abnormalities detection using vessel thickness and curvature analysis,” in *13th Int. Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2009, pp. 151–158.

[8] C.-H. Wen, W.-D. Liao, T.-Y. Hsieh, D.-Y. Chen, J.-L. Lan, and K.-C. Li, “Computer-aided image analysis aids early diagnosis of connective-tissue diseases,” in *SPIE Newsroom, Biomedical Optics & Medical Imaging*, 2009.

[9] C.-H. Wen, T.-Y. Hsieh, W.-D. Liao, J.-L. Lan, D.-Y. Chen, K.-C. Li, and Y.-T. Tsai, “A novel method for classification of high-resolution nailfold capillary microscopy images,” in *1st IEEE International Conference on Ubi-Media Computing*, 2008, pp. 513–518.

[10] H. Kwasnicka, M. Paradowski, and K. Borysewicz, “Capillaroscopy image analysis as an automatic image annotation problem,” in *6th Interenational Conference on Computer Information Systems and Industrial Management Applications*, 2007.

[11] B. F. Jones, M. Oral, C. W. Morris, and E. F. J. Ring, “A proposed taxonomy for nailfold capillaries based on their morphology,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 4, pp. 333–341, 2001.

[12] N. P. Doshi, G. Schaefer, and A. Merla, “Nailfold capillaroscopy pattern recognition using texture analysis,” in *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 2012.

[13] Z. Guo, L. Zhang, and D. Zhang, “Rotation invariant texture classification using LBP variance (LBPV) with global matching,” *Pattern Recognition*, vol. 43, no. 3, pp. 706–719, 2010.

[14] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study for texture measures with classification based on feature distributions,” *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

[15] T. Ojala, M. Pietikäinen, and T. Maenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[16] N. P. Doshi and G. Schaefer, “A comparative analysis of local binary pattern texture classification,” in *Visual Communications and Image Processing*, 2012.

[17] G. Schaefer and N. P. Doshi, “Multi-dimensional local binary pattern descriptors for improved texture analysis,” in *21st International Conference on Pattern Recognition*, 2012, pp. 2500–2503.

[18] N. P. Doshi and G. Schaefer, “Texture classification using multi-dimensional LBP variance,” in *2nd IAPR Asian Conference on Pattern Recognition*, 2013.

[19] C. Gatta and P. Radeva, “Bilateral enhancers,” in *16th IEEE International Conference on Image Processing*, 2009, pp. 3161 –3164.

[20] N. P. Doshi, G. Schaefer, and A. Merla, “Enhancement of nailfold capillaroscopy images,” in *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 2012.

[21] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.

[22] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[23] C.-C. Chang and C.-J. Lin, “libSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

SESSION

POSTER PAPERS

Chair(s)

TBA

Image fusion using two symmetric exposed images in the JPEG stream

Geun-Young Lee[1]**, Sung-Hak Lee**[1]**, Hyuk-Ju Kwon**[1]**, and Kyu-Ik Sohng**[1]
[1]School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea

Abstract - *The objective of fusing differently exposed images is to well represent the luminance information of the scene in a single image. In order to fuse multi exposure images, it is necessary to determine overexposed or underexposed areas in each of the captured images using an activity measure which extracts image features. In this paper, we presents image fusion using two symmetric exposed images in the JPEG stream. We find out that the AC coefficients after the quantization in JPEG baseline can describe local image features in the image block. Based on a maximum selection rule, a proposed image fusion is conducted by the comparison of the AC coefficients after the quantization instead of activity measures which need additional computations. This simple method allows multi exposure images to quickly combine a single image in JPEG baseline, therefore, it can be embedded in resource limited imaging platform such as a surveillance system.*

Keywords: Image fusion, multi exposure images, JPEG, resource limited imaging platform

This paper is Extended Abstract/Poster Paper.

1 Introduction

In general, luminance range in real scene is wider than the range of digital camera. In addition, a commercial image format is capable of storing only 8 bits per channel, so that the range of storable luminance in one image is limited. Therefore, in order to cover whole luminance information in real scene, the information must be divided and allocated to several images with different exposures. However, this division needs not only more storage memory but also inconvenience which is caused by scanning several images to recognize luminance information.

In order to solve such problems, the methods fusing differently exposed images into a single image (called exposure fusion [1]) have been proposed. In the process of the multi exposure image fusion, an activity measure is essential in order to select what part of each image is included in an fusion image. Song et al. [2] measured quality using the visible contrast and gradients. Mertens et al. [1] used the quality information constructed by contrast, saturation, and well-exposedness. In addition, entropy measure was used in [3] and Pu [4] used directive contrast using subband images in the discrete wavelet transform domain.

In this paper, we propose image fusion using two exposed images which have symmetric exposure values (EVs), +EV and –EV. In particular, our approach is based on a maximum selection rule which utilizes the zig-zag order of the highest frequency AC coefficient remaining after quantization in the JPEG standard. Based on the assumption that the zig-zag order of the highest frequency AC coefficient indicates the degree of details in the 8×8 block, we select AC coefficients from the JPEG stream having the higher zig-zag order. The proposed method is effective to acquire a fusion image with fair quality, simply and quickly.

2 Image fusion in the JPEG stream

2.1 JPEG base line

JPEG [5] is a widely used image compression standard. Due to the simplicity of the processing and good compression performance for fair quality images, all kinds of digital cameras store images using the JPEG standard. In the JPEG baseline, a RGB color image is first transformed to YCbCr color space. JPEG compression divides an image into non-overlapped 8×8 blocks and then each pixel in the block is transformed into frequency domain using discrete cosine transform (DCT). The transformed 8×8 block consists of one DC coefficient and other 63 AC coefficients. And, the quantization process is that DCT coefficients are divided by the quantization matrix and rounded off to the nearest integer. For coding efficiency, the quantized coefficients are listed in zig-zag order. Because a number of high frequency AC coefficients are rounded off to zeros, the zig-zag ordering forms long sequence of zeros following low frequency AC coefficients.

2.2 JPEG data stream

JPEG data stream is the result of entropy encoding for DCT coefficients in the zig-zag order. And, the length of the sequence without the zero sequence corresponding to high frequency AC coefficients rounded off to zeros can be directly estimated from run-length coded data stream. For example, JPEG data stream for the block in Fig. 1(c) is (0, 8)(176), (1, 3)(-6), (1, 3)(-4), (0, 0). This data stream has the sequence of which length is 5 (16, 0, -6, 0, and -4). Note that bit stream encoding using Huffman code is skipped for clear understanding.

2.3 Image Fusion using JPEG data stream

In general, the detail in the bright region well appears in +EV image, whereas the detail in the dark does in –EV image. In other words, in order to reproduce the detail of the scene in a fusion image, it is necessary to decide which of two images well represents the detail in each region. Fortunately, the quantization process in the JPEG baseline makes the insufficient level of the detail converge to zero.

Let $I = \{i_{x,y}\}(x = 0, \ldots, N{-}1$ and $y = 0, \ldots, M{-}1)$ be an image and it is divided into $N{\times}M$ blocks, i, of size 8×8. Let $C_n = \{d_{n,u,v}\}(n = 0, \ldots, N{\times}M{-}1$, $u = 0, \ldots, 7$, and $v = 0, \ldots, 7)$ be corresponding DCT coefficients of each nth 8×8 block. The fusion rule is that the block with the maximum length of the JPEG data stream belongs to a fusion image. Therefore, in proposed image fusion, the nth block of the fusion image, $C^F{}_n$, is obtained as follows;

$$C^F{}_n = C^K{}_n, \quad \text{where} \quad K = \arg\max_k \{Z^k{}_n\}, \quad k = +EV, -EV. \qquad (1)$$

$Z^k{}_n$ is the length of the JPEG data stream of the nth block of the k image. DC coefficients of a fused image are the average values of those of two images.

3 Simulations

In Fig. 2, we show the simulation images (top row). In order to peer into the images, cropped and enlarged regions (sub1 and sub2) in each images are shown in middle and bottom rows. Compared to input images (+*EV* image and –*EV* image), a fusion image well captures the details in the dark region in the scene which appears in +*EV* image and bright region which appears in –*EV* image.

4 Conclusions

In this paper, we propose image fusion using two symmetric exposed images in the JPEG data stream. And, we confirm that it is efficient for fusing two images without additional activity measures in JPEG baseline. The use of the quantized DCT coefficients as activity measure makes the fusion process simple and fast. Therefore, it can be available for resource limited imaging platform such as a surveillance system.

5 Acknowledgement

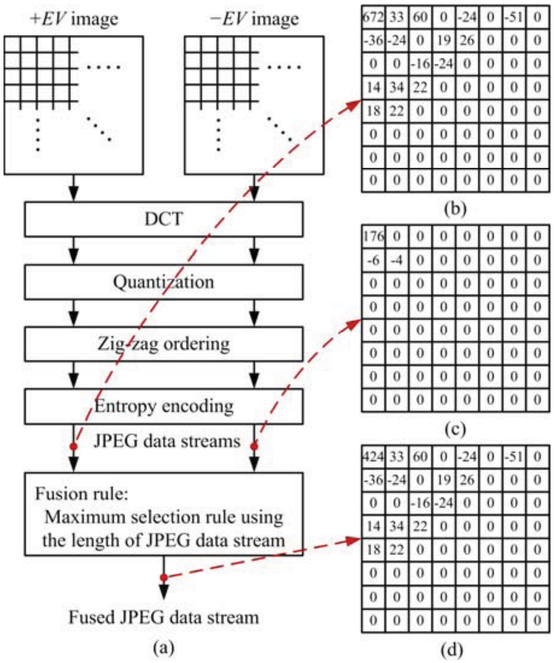
This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A01059929).

6 References

[1] T. Mertens, J. Kautz, and F. Van Reeth, “Exposure fusion: A simple and practical alternative to high dynamic range photography,” *Computer Graphics Forum*, vol. 28, no. 1, pp. 161-171, 2009.

[2] M. Song, D. Tao, C. Chen, J. Bu, J. Luo, and C. Zhang, “Probabilistic exposure fusion,” *IEEE Trans. Image processing*, vol. 21, no. 1, pp. 341-357, 2012.

[3] A. A. Goshtasby, “Fusion of multi-exposure images,” *Image and Vision Computing*, vol. 23, no. 6, pp. 611-618, 2005.



672	33	60	0	-24	0	-51	0
-36	-24	0	19	26	0	0	0
0	0	-16	-24	0	0	0	0
14	34	22	0	0	0	0	0
18	22	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

176	0	0	0	0	0	0	0
-6	-4	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

424	33	60	0	-24	0	-51	0
-36	-24	0	19	26	0	0	0
0	0	-16	-24	0	0	0	0
14	34	22	0	0	0	0	0
18	22	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 1. Proposed flowchart and example of 8×8 blocks; (a) proposed flowchart, (b) +EV image block which has the coefficient length of 38, (c) –EV image block which has the coefficient length of 5, and (d) fused image block. Because the sequence of +EV image block is longer than that of –EV image block, fused image block has the AC coefficients of +EV image block. DC coefficient of fused image block is the average of those of two image blocks.

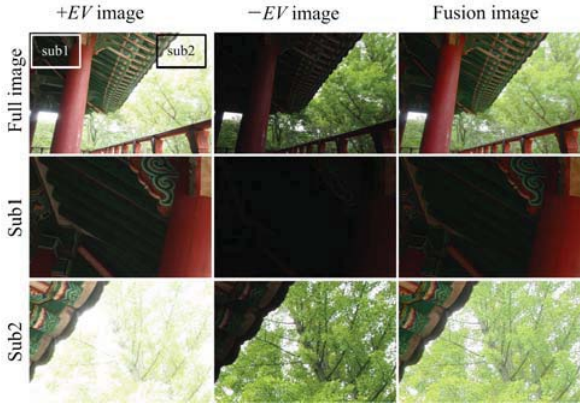


Fig. 2. Simulation images (top) and crop images (middle and bottom).

[4] T. Pu and G. Ni, “Contrast-based image fusion using the discrete wavelet transform,” *Optical Engineering*, vol. 39, no. 8, pp. 2075-2082, 2000.

[5] G. K. Wallace, “The JPEG still picture compression standard,” *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

Robust Image Matching using Statistical Modeling and Geometric Similarity

In-su Won[1]**, Sang-min Lee**[1]**, and Jang-woo Kwon**[2]
[1]Dept. of Electronic Engineering, Inha University, Incheon, Korea
[2]Dept. Computer and Information Engineering, Inha University, Incheon, Korea

Abstract - *We propose a robust image matching method using statistical modeling and clustering of geometric similarity between matching-pairs. Local feature matching is an uncertain process which may provide incorrect matches due to some causes that include among other factors, the uncertainly in feature location. Since the statistical modeling of the Log Distance Ratio (LDR) for outliers are significantly different from those of inliers. Although fast and efficiently, LDR has some weakness, especially related to the inability to take into consideration the uncertainly in the feature location and performance degrades when strong perspective transform. We add a method that clustering the similarity of geometric relationship. The proposed method robustly matches images, even with various kinds of transformation.*

Keywords: Image Matching, Statistical Modeling, Geometric Similarity

1 Introduction

Image matching is a fundamental step for computer vision, such as object detection, recognition, tracking and augmented reality. This is mostly achieved relying on local feature computed in the neighborhood of detected features. Early research used neighbor-pixel information around the features. However, these methods may result in many incorrect matches, known as outlier. This is overcome by RANSAC [1], which is a good outlier-removal method. RANSAC has been adopted in many application, proving its effectiveness: nevertheless, it presents some well-known drawbacks, due to its iterative process, which results in significant computational complexity in existence of many outliers.

Recently the Log Distance Ratio (LDR) [2] is presented to detect outliers to a low-complexity through statistical modeling. It found that inliers have a specific ratio when LDR is calculated using feature coordinates. The effectiveness of the presented approach is adopted as the standard in Moving Picture Expert Group (MPEG) - Compact Descriptors for Visual Search (CDVS) [3]. Although fast and efficiently, LDR still presents some weakness, especially related to the inability to take into consideration the uncertainly in the feature location. Furthermore, performance degrades when strong perspective transform. To overcome the weakness in LDR, we present a method that clustering the similarity of geometric relationship.

In the following section, we briefly the LDR for outlier detection. Section 3 introduces the proposed robust image matching method.

2 The Log distance ratio statistic

Suppose that for a given two images the features have been found and matched (x_1,y_1), (x_2,y_2),…,(x_N,y_N), where x_n is the feature coordinates in the first images and y_n is the feature coordinates in the second image that is matched to x_n. The LDR sets (Z) of all matching-pairs is given by the following:

$$Z_{i,j} = ln\left(\frac{\|x_i - x_j\|}{\|y_i - y_j\|}\right), Z = \{z_{ij} | i \neq j\} \qquad (1)$$

where the symbol ||~|| denotes the Euclidean distance. It may be observed that the LDR for inliers is distributed in a manner that is distinctively different to how the LDR for outliers and mixed pairs are distributed. This behavior is studied by first forming a histogram *h(k)* for these values, by counting the occurrences over each bin,

$$h(k) = \#(Z \cap \zeta_k). \qquad (2)$$

The bins $\zeta_1, \dots \zeta_K$ are adjacent intervals. The inlier behavior can be expressed by the double inequality

$$a\|x_i - y_j\| \leq \|y_i - y_j\| \leq b\|x_i - x_j\|. \qquad (3)$$

where a and b define the boundaries of the LDR for inliers. The inliers would contribute to bins contained in $[-\ln b, -\ln a]$ which for most cases is a limited portion of the histogram. The outlier behavior is modelled and is express through a discrete probability density function, called outliers model function, as *f(k)*. Exploiting this, the match between two images can be determined. Pearson's good-of-fit test is used to compare *h(k)* and *f(k)*. Equation 4 is used to compute this similarity. A greater value for *c* implies that the difference between *h(k)* and *f(k)* is large, and that the matching pair has numerous inliers.

$$c = \sum_{k=1}^{K} \frac{\left(h^k - nf_k\right)^2}{nf_k} \geq \chi^2_{1-a,K-1}. \qquad (4)$$

n is the total number of matching pairs, and $\chi^2_{1-a,K-1}$ is the threshold of χ^2 having *K-1* degrees of freedom. It can see that the LDR of matching pairs is narrow with numerous inliers, and that a match can be rapidly identified by calculating this difference. The exact number of inliers can be estimated by solving the eigenvalue problem described in [2].

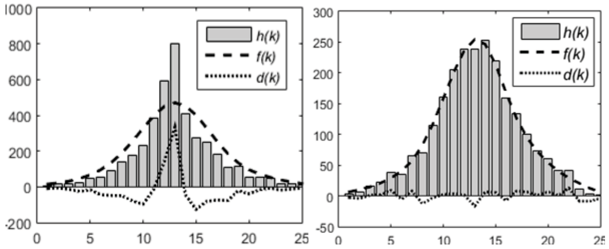


Figure 1. An example of LDR histogram h(k), model function f(k), and difference d(k). (left) Correct matching-pair, (right) Incorrect matching-pair

3 Clustering of inliers for similarity of geometric relationship

The LDR used just distance ratio between matching-pairs, so it includes the location uncertainly. we used a method of clustering inliers with similar geometric models. Letting two matching pairs be M_i, and M_j, transformations T_i and T_j and translations t can be calculated from each matching-pair, using the enhanced WGC method, as shown in Equation 5:

$$\begin{bmatrix} x'_q \\ y'_q \end{bmatrix} = s' \begin{bmatrix} cos\theta' & -sin\theta' \\ sin\theta' & cos\theta' \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \quad t = \left| q'(x_q, y_q) - q(x_q, y_q) \right|. \tag{5}$$

s is scale, θ is the dominant orientation. Using Equation 5, the matching-pairs can be expressed as $M_i = ((x_i, y_i), (x'_i, y'_i), T_i)$ and $M_j = ((x_j, y_j), (x'_j, y'_j), T_j)$ and the geometric similarity of the two matching-pairs is calculated using Equation 6:

$$d(M_i, M_j) = \frac{1}{2}\left(\|X'_j - T_i X_j\| + \|X'_i - T_j X_i\|\right). \tag{6}$$

where $X_k = [x_k, y_k]^t, X'_k = [x'_k, y'_k]^t, (k = i, j)$. If transformation T_i and T_j are similar, $d(M_i, M_j)$ will be close to zero. Using this relationship, it can be assumed that the matching-pairs with a small $d(M_i, M_j)$ have a similar geometric relationship. Hierarchical clustering groups the clusters through linkages by calculating the similarity within each cluster. Figure 2 shows the matching results of hierarchical clustering based on geometric similarity between matching pairs. Hierarchical clustering finally forms a single cluster. Therefore, clustering must be stopped at some point. During the linkage of clusters, clustering is stopped when the geometric similarity of the matching-pair exceeds a set threshold. However, if such a thresholding method is the only one used, the number of clusters becomes excessive, with most of the clusters likely being false. Therefore, clustering validation is used to remove false clusters. If the number of matching-pairs that form the clusters is too small, it is less likely that the resulting clusters become objects. Hence, two methods are used for clustering validation. First, if the number of matching-pairs that form the cluster is greater than τ_m. Secondly, if the area of the matching-pairs that form the cluster is larger than a certain portion of the entire area. The area of the matching pairs that form the cluster is calculated using a convex hull.

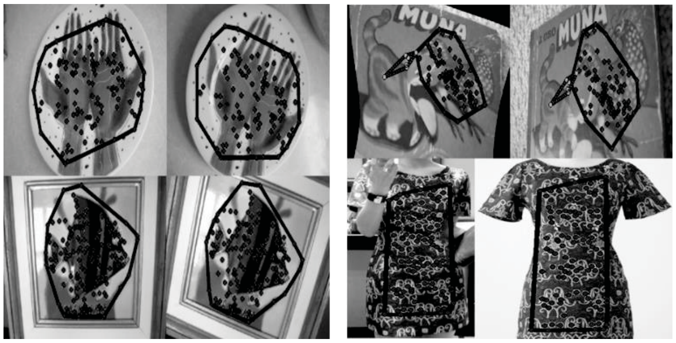


Figure 2. Matching results for clustering of geometric similarity. (left) Perpective transformation, (right) deformable transformation

4 Experiments

In order to evaluate the performance of the proposed matching method, the Stanford Mobile Visual Search (SMVS) dataset [3]. (4,200 query images, total 12,800 images)

Table 1. Performance results of image matching methods

	TPR	FPR	Accuracy
LDR [2]	83.51%	6.41%	88.55%
Proposed	**89.78%**	**7.12%**	**91.33%**

5 Conclusions

We have proposed a robust image matching using statistical modeling and clustering of geometric similarity. It may apply to various computer vision applications such as object detection, recognition and so on.

6 Acknowledge

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2010-0020163)

7 Reference

[1] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol.24, no.6, pp.381-395, 1981.

[2] S. Lepsøy, G. Francini, G. Cordara, and P.P.B. de Gusmao, "Statistical modelling of outliers for fast visual search," IEEE International Conference on Multimedia and Expo (ICME), pp.1-6, July, 2011.

[3] ISO/IEC JTC1/SC29/WG11/N13925, "Text of ISO/IEC CD 15938-13 Compact Descriptors for Visual Search", November, 2013.

Object Tracking Method based on Color Particle Filter

Mai Thanh Nhat Truong*, Sanghoon Kim*
*Department of Electrical, Electronic, and Control Engineering, Hankyong National University

Corresponding author: kimsh@hknu.ac.kr(Sanghoon Kim)

Abstrat

Object tracking in real time video is a challenging task and has many necessary applications. Particle filtering has been proven very successful for non-Gaussian and non-linear estimation application. In this research, we tried to develop a color-based particle filter. And the color distributions of video frames are integrated into particle filtering. Color distributions are applied because of their robustness and computational efficiency. The model of the particle filter is defined by the color information of the tracked object. The model is compared with the current hypotheses of the particle filter using the Bhattacharyya coefficient. The proposed tracking method directly incorporates the scale and motion changes of the objects. Experimental results have been presented to show the effectiveness of our proposed system

Keywords: object tracking; particle filter; color based model;

1. Introduction

Object tracking is required in many vision-based applications such as human-computer interfaces, video communication, road traffic control, security and surveillance systems. The main goal of object tracking is to obtain a record of the trajectory of the moving single or multiple targets over time and space, by processing information from distributed sensors. Object tracking in video sequences requires on-line processing of a large amount of data. Additionally, most of the problems encountered in visual tracking are nonlinear, non-Gaussian, multi-modal or any combination of these. Therefore, tracking objects can be extremely complex and time-consuming, especially when it is done in outdoor environments.

Recently, the particle filter method, a numerical method that allows finding an approximate solution to the sequential estimation, has been shown to be very successful for nonlinear and non-Gaussian estimation problems. This approximates a posterior probability density of the state, such as the object's position, by using samples which are called particles. The particle filter based tracking algorithms usually use contours, color features, or appearance models [1, 2, 3, 4, 5]. The color histogram is robust against noise and partial occlusion, but suffers from illumination changes, or the presence of the confusing colors in the background.

Although particle filters have been widely used in recent years, they have important drawbacks. One is sampling impoverishment, i.e., samples are spread around several modes pointing out the different hypotheses in the state space, but most of these may be spurious. In addition, the objects with a higher likelihood may monopolize the sample set, and objects whose samples exhibit a lower likelihood have a higher probability of being lost. On the other hand, the computation is expensive if the tracked region and the number of samples are large. The contour-based methods are invariant against the illumination variation but computationally expensive which restricts the number of samples (particles). Unfortunately when the dimensionality of the state space increases, the number of samples required for the sampling increases exponentially.

2. Particle filter

2.1 Classical particle filter

Particle filtering [1] was developed to track objects in clutter, in which the posterior density $p(X_t \mid Z_t)$ and the observation density $p(Z_t \mid X_t)$ are often non-Gaussian. The quantities of a tracked object are described in the state vector X_t while the vector Z_t denotes all the observations $\{z_1, \ldots, z_t\}$ up to time t.

The key idea of particle filtering is to approximate the probability distribution by a weighted sample set $S = \{(s_n, \pi_n) \mid n = 1 \ldots N\}$. Each sample consists of an element s which represents the hypothetical state of an object and a corresponding discrete sampling probability π. The evolution of the sample set is described by propagating each sample according to a system model. Then, each element of the set is weighted in terms of the observations and N samples are drawn with replacement, by choosing a particular sample with probability $\pi_n = \mathrm{p}(z_t \mid X_t = s_t(n))$. The mean state of an object is estimated at each time step by

$$E[S] = \sum_{n=1}^{N} \pi_n s_n \qquad (1)$$

Particle filtering provides a robust tracking framework, as it models uncertainty. It can keep its options open and consider multiple state hypotheses simultaneously. Since less likely object states have a chance to temporarily remain in the tracking process, particle filters can deal with short lived occlusions.

2.2 Color Model

We want to apply such a particle filter in a color-based context. To achieve robustness against non-rigidity, rotation and partial occlusion we focus on color distributions as target models. These are represented by histograms which are produced with the function $h(x_i)$, that assigns one of the m-bins to a given color at location x_i. The histograms are

ISBN: 1-60132-442-1, CSREA Press ©

typically calculated in the RGB space using 8x8x8 bins. To make the algorithm less sensitive to lighting conditions, the HSV color space could be used instead with less sensitivity to V (e.g. 8x8x4 bins).

Not all pixels in the region are equally important to describe the objects. For example, pixels that are further away from the region center can be assigned smaller weights by employing a weighting function

$$k(r) = \begin{cases} 1 - r^2 & : \quad r < 1 \\ 0 & : \quad \text{otherwise} \end{cases} \tag{2}$$

where r is the distance from the region center. Thus, we increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded. It is also possible to use a different weighting function for example the Epanechnikov kernel.

The color distribution $p(y) = \{p_u(y)\}_{u=1\ldots m}$ at location y is calculated as

$$p_u(y) = f \sum_{i=1}^{I} k\left(\frac{\|y - x_i\|}{a}\right) \delta[h(x_i) - u] \tag{3}$$

where δ is the Kronecker delta function and a is used to adapt the size of the region. The normalization factor is

$$f = \frac{1}{\sum_{i=1}^{I} k\left(\frac{\|y - x_i\|}{a}\right)} \tag{4}$$

In a tracking approach the estimated state is updated at each time step by incorporating the new observations. Therefore, we need a similarity measure which is based on color distributions. A popular measure between two distributions $p(u)$ and $q(u)$ is the Bhattacharyya coefficient [6],

$$\rho[p, q] = \int \sqrt{p(u)q(u)}du \tag{5}$$

Considering discrete densities such as our color histograms $p = \{p_u\}_{u=1\ldots m}$ and $q = \{q_u\}_{u=1\ldots m}$ the coefficient is defined as

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{6}$$

The larger ρ is, the more similar the distributions are. For two identical histograms we obtain ρ = 1, indicating a perfect match. As distance between two distributions we define the measure

$$d = \sqrt{1 - \rho[p, q]} \tag{7}$$

which is called the Bhattacharyya distance.

3. Experiments

In this section, we present the performance of object tracking method. The algorithm is implemented in C++, using OpenCV library. The tracking process is shown in Figure 1. It shows the result of the color-based particle filter using N = 2000 samples, in this case the histograms are calculated in the RGB color space using 8x8x8 bins. The figure shows that the color-based particle filter yields good performance when applied in object tracking.

4. Conclusions and future work

In this study we developed a color-based particle filter for object tracking. The experimental results show that our method capable of tracking object presented in color video, yielding high accuracy and reliable results. For further development, we will develop a multiple object tracking for visual inspection system, which uses color-based particle filter. The aim of future tracking method is making it robust in dealing with different problems in visual object tracking. This multiple object tracking will also have low computational complexity in order to be deployed on embedded systems.

5. Acknowledgments

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2015R1D1A1A01057518)'

References

[1] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. Int'l Journal of Computer Vision, 29(1):5–28, Aug. 1998.

[2] M. Isard and A. Blake. Condensation: Unifying low-level and high-level tracking in a stochastic framework. In Proc. European Conf. Computer Vision, volume 1, pages 893–908, 1998.

[3] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In Proc. Int'l Conf. Computer Vision, pages 572–578, 1999.

[4] Y. Wu. Robust visual tracking by integrating multiple cues based on co-inference learning. Int'l Journal of Computer Vision, 58(1):55–71, June 2004.

[5] Shun-Sheng Ko, Chien-Sheng Liu, Yang-Cheng Lin, Optical Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. In Proc. IEEE Conf. Comp. Vision Pattern Recognition, volume 2, pages 980–986, Washington DC, 2004.

[6] F. Aherne, N. Thacker, and P. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. Kybernetika, 32(4):1–7, 1997.

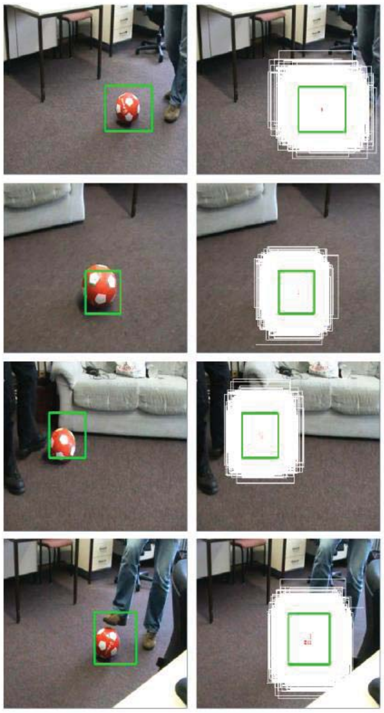


Figure 1. The object tracking method is used to track a ball in video. Each row shows tracked object (left) and the locations of corresponding particles (right).

Laser Based Vision System for Inspection of Small Adhesive Bead

Hyuk-Ju Kwon[1], Young-Choon Kim[2], and Sang-Ho Ahn[3]
[1]School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea
[2] Dept. of Information Communication & Security, Youngdong University, Asan, Republic of Korea
[3] Dept. of Electronic Engineering, Inje University, Gimhae, Republic of Korea

Abstract - *Line laser based vision system for an inspection of a small adhesive bead is introduced. To measure the height and width of a small adhesive bead, a robust vision software is proposed. To reduce a wide laser beam influence, a noise is re reduced and a center line is extracted. For finding the best-fitting line and curve from a set of center points, RANSAC (RANdom SAmple Consensus) algorithm is used. To calculate the width and height of the adhesive bead, feature points are extracted.*

Keywords: line laser, vision system, adhesive bead, optical triangulation, non-invasive inspection

This paper is Extended Abstract/Poster Paper.

1 Introduction

In many industries, manufacturers of metal components are turning to structural acrylic adhesives to replace or augment rivets, bolts, welding and other traditional fastening methods in their assembly processes. There are a number of reasons for doing so, including improved product performance, improved aesthetics, reduced overall assembly time and lower production costs.[1]

A camera based vision system integrated with inspection equipment is widely used to inspect objects. In case of an adhesive bead inspection, a height inspection as well as a width inspection is also required to recently enhance reliability of product. Although a stereo vision system can be applied for inspecting the height of adhesive, it is difficult to acquire the disparity of maximum height, since the width of adhesive is narrow and its surface is flat.[2]

On the other hand, a line laser based vision system using an optical triangulation is more usable to inspect the height, due to its inherent relative simplicity, robustness and reliability. The optical triangulation is a typical technique which achieves the high-range accuracy with simple calculation.[3,4]

The width and height of automotive adhesive bead are usually small. In this case, the laser based vision system is depends on the laser beam size and the camera resolution. The laser based vision system for inspection of small adhesive beads requires very thin laser beam size and high resolution camera. Because the thinness of laser beam has limit, the laser beam width illuminated on the small adhesive bead is relatively wide. Therefore, to measure the height and width of the small adhesive bead, a robust vision software is required.

In this paper, a line laser based vision system is introduced for the inspection of the small adhesive bead. The line laser vision system uses the optical triangulation geometry. To measure the height and width of the small adhesive bead, a robust vision software is proposed. To reduce a wide laser beam influence, a noise is reduced and a center line is extracted. For finding the best-fitting line and curve from a set of center points, RANSAC (RANdom SAmple Consensus) algorithm is used. To calculate the width and height of the adhesive bead, feature points are extracted. The validity of proposed line laser based vision system is verified through experiments for adhesive beads,

2 Line laser based vision system

A concept of optical triangulation geometry is shown in Fig. 1. The line laser is vertically installed to reference plane and the camera is leaned with gaze angle θ. The image of laser beam mirrored to the reference plane falls at zero central coordinates. For the distance B between camera and laser, vertical distance D_0 between the camera and the reference plane is given by

$$D_o = B/\tan\theta \qquad (1)$$

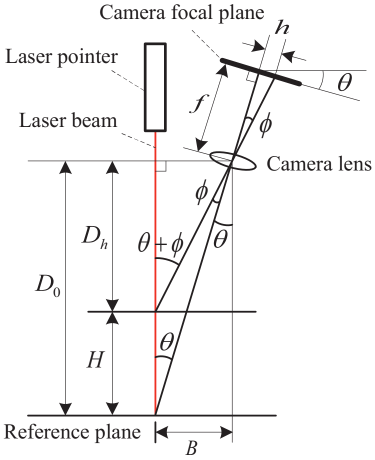


Fig. 1. Concept of optical triangulation geometry.

When laser beam is illuminated on an object located at the height H from the reference plane, the image is mirrored on the image plane falls on a location away with h from the central coordinates. In other words, as the height of object H get higher, the position of laser beam mirrored on the image plane is dislocated with h, the related formula is as follows;

$$\phi = \tan^{-1}\left(h/f\right) \tag{2}$$

$$D_h = \frac{B}{\tan(\theta+\phi)} \tag{3}$$

$$H = D_o - D_h \tag{4}$$

where f is the focal length of the lens.[1]

The captured image from camera is processed through an image processing flow such as Fig. 2 and a width and a height of structural adhesive bead are calculated. The captured image is converted to binary image by a threshold processing and noise is eliminated by filtering. Center points of the binary object image are extracted using a local labeling. For finding the best-fitting line and curve to given set of center points, RANSAC algorithm is used. RANSAC algorithm is a learning technique to estimate parameters of a model by a random sampling of observed data. Feature points are extracted and the width and height of the object are calculated.

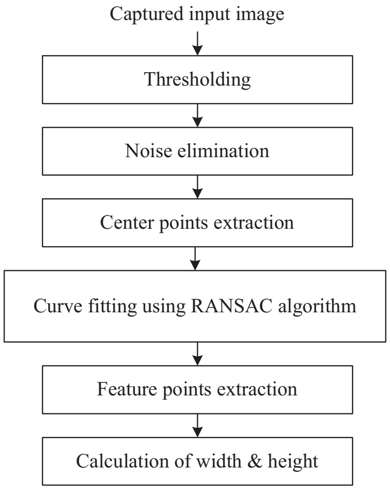
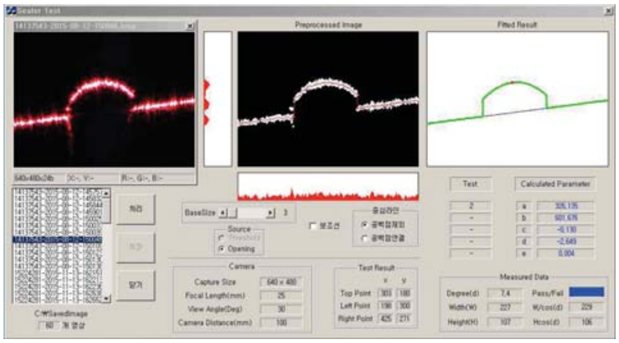

Captured input image
Thresholding
Noise elimination
Center points extraction
Curve fitting using RANSAC algorithm
Feature points extraction
Calculation of width & height

Fig. 2. Flowchart of image processing for adhesive bead inspection.

3 Implementation of vision system

We implemented a vision system based on the line-laser for the adhesive bead inspection. An example of processed result using the developed software is shown in Fig. 3. The left image is an original capture image of the adhesive bead which laser beam illuminated. We can see that the laser beam size is wider than the size of the adhesive bead. The middle image illustrates a binary image with the extracted center line. And the right image illustrates a best-fitting line and curve using RANSAC algorithm. The red point on the green curve is a feature point. It is a point of contact between the parallel line of blue line and the green curve. Also two contact points between line and curve are extracted. The three feature points are used for calculation of the width and height of the adhesive bead. The results of the width and height of the adhesive bead are illustrated in the bottom right of Fig. 3.



BaseSize 3

Source: ○ Threshold ● Opening

Camera
Capture Size 640 x 480
Focal Length(mm) 25
View Angle(Deg) 30
Camera Distance(mm) 100

Test Result (x, y)
Top Point 303 180
Left Point 196 300
Right Point 425 271

Calculated Parameter
a 326.135
b 601.676
c -0.130
d -2.649
e 0.004

Measured Data
Degree(d) 7.4
Width(W) 227
Height(H) 107
W/cos(d) 229
Hcos(d) 106

Fig. 3. Software for adhesive bead inspection.

4 Conclusions

In this research, a line laser based vision system is introduced for the inspection of a small adhesive bead. The line laser vision system uses the optical triangulation geometry. To measure the height and width of a small adhesive bead, a robust vision software is proposed. The validity was confirmed through experiments.

5 Acknowledgement

This work was supported by a research project of SMBA(small and medium business administration) in Korea.

6 References

[1] T. Satoh, Y. Miyazaki, Y. Suzukawa, an K. Nakazato. "On the development of structural adhesive technology for the automotive body in Japan"; *JSAE Review*, vol. 17, no. 2, pp. 165-178, 1996.

[2] N. Lazaros, G.C Sirakoulis, and A. Gasteratos. "Review of stereo vision algorithm: from software to hardware"; *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[3] T.A. Clarke, K.T.V. Grattan, and N.E. Lindsey. "Laser-based triangulation techniques in optical inspection of industrial structures"; *Proc. SPIE 1332, Optical Testing and Metrology III: Recent Advances in Industrial Optical Inspection*, vol. 474, 1991.

[4] Y. Kim, J. Son, T. Bae, and S. Ahn. "Inspection of structural adhesive using line-laser based vision system"; *Optical Engineering*, IJCA, vol. 8, no. 2, pp. 2075-2082, 2015.

Enhancing Stereo Image Quality based on Adaptive Hole-Filling of Depth Image for Kinect Camera

Ji-Min Cho[1], Young-Ji Yun[1], Seung-Woo Nam[2], and Sung-Il Chien[1]

[1]School of Electronics Engineering, Kyungpook National University, Daegu, Korea

[2]SW Content Research Lab, Electronics and Telecommunications Research Institute, Daejeon, Korea

Abstract - *Kinect camera produces depth images as well as color images. Yet undefined depth regions called holes are often observed in the depth image and are quite harmful in generating a second view image to construct a stereo pair for 3D viewing. An adaptive hole-filling method that can fill in even very large holes more naturally is proposed so that depth values for holes of various sizes are suitably estimated depending on the location of holes in the background or foreground. Removal of dot noise inside a hole is also introduced to further enhance the quality of the newly generated stereo image.*

Keywords: Kinect, Hole Filling, Multiview Generation

1 Introduction

From Kinect camera, we can obtain a color image and depth image. Kinects are quite popular as a motion sensing input device for game industry. However, the stereo images are not available from Kinect. The remedy for this is to generate a second view used to make a stereo image set by using color and depth information, on which the research has been performed until recently. Yet Kinect introduces undefined region called holes whose depth information is missing. If hole sizes are quite large, the procedure of second view generation is disrupted and resulting stereo images are not satisfactory for 3D applications.

Hole-filling with suitable depth values has also been studied until recently. The Telea method [1] is relatively fast but is sensitive to noise especially for big holes. By reducing the search area for exemplar based image inpainting, Patel method [2] becomes fast and preserves edges well but its speed is still relatively slow. Choi method [3] uses frame differences to focus on the moving objects for hole-filling in video application. For hole-filling, Chen method [4] uses edges information but this method seems to be inexact for the case of lots of edges in the background. Our hole-filling method is an adaptive approach. Holes are divided into large and small holes. Small holes are filled by Telea method. Large holes are further divided into two groups: holes inside an object and holes in the background or located between background and foreground. Each hole-filling method is tuned to be suitable for each case.

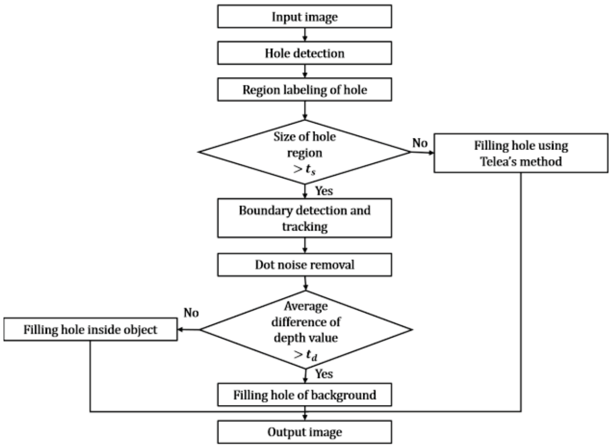

Input image
Hole detection
Region labeling of hole
Size of hole region $> t_s$
No
Filling hole using Telea's method
Yes
Boundary detection and tracking
Dot noise removal
Average difference of depth value $> t_d$
No
Filling hole inside object
Yes
Filling hole of background
Output image

Fig. 1. Overall diagram of proposed method

2 Hole-filling for Kinect depth images

Holes representing undefined depth regions due to various reasons appear in depth images of Kinect but sometimes need be filled with estimated depth values, since the depth determines the amount of pixel movement when the stereo view is generated using the depth and reference images. Our method of estimating depth values for the hole areas is summarized in Fig.1. Once the holes are divided into two groups of large and small holes. The depth values of the small holes are determined by Telea method [1], which is relatively fast and quite suitable for our case. First, for the large holes detected, we have performed a procedure of dot noise removal. It was found that, inside a relatively large hole, small dotted regions sometimes appear and become quite irritating to a viewer, when the resulting stereo pair is observed using a 3D monitor or Oculus. The depth values of quite small regions inside a big hole are replaced by an average depth value around a hole, while the depth values of the region above the threshold remains untouched. The next step is to divide the large holes into two types of holes by using the average of the depth values around the hole boundary: holes inside on object and holes of background or holes between background and foreground. The depth values inside the object are replaced by the average depth value of the neighboring pixels to the tracked hole boundary, which is found to provide viewers with smooth feeling about the object surface when watched on a 3D monitor. Another case deals with a situation that big holes are located on the background. The hole pixels are replaced by an average value of the first and second maximum depth values around the hole position of 3×3 window. The same procedure applies to the situation that holes are sandwiched between the background and foreground. The holes tend to be replaced by the depth values of background, since the depth values of background are much larger than those of foreground for the Kinect camera.

3 Experiment result

We have tested and evaluated our method for various images from indoor office environments. Each missing stereo image is generated from the original Kinect color image and the depth image of which the holes are filled by the proposed method. Fig. 3(a) shows such a typical image and Fig. 2(a) represents part of a corresponding depth image, in which the holes from Kinect are shown in black. Corrected depth images by Telea and our proposed method for the rectangle in Fig. 2(a) are shown in Fig. 2(b) and (c). Here the boundary of the object of the proposed method becomes clearer. When we observe the newly generated stereo view, the improvement is much clearer, in which the shape of the black monitor is well preserved by the proposed method.

4 Conclusions

Holes are first divided into two groups according to their sizes. Small holes are filled with a depth value calculated by Telea method. The large holes, for which hole-filing is not easy, is further divided into two groups according to the variation of depth values near the hole boundary: holes inside an object and holes in background. Suitable algorithms are adopted for two cases. It is found that final stereo images show better quality and irritating or unnatural feelings often reported on 3D viewing are much minimized.

5 References

[1] Alexandru Telea. “An Image Inpainting Technique Based on The Fast Marching Method”; Journal of Graphics Tools, Vol.9, No.1, 25-36, 2004.

[2] Jayesh Patel and Tanuja K. Sarode. “Exemplar based Image Inpainting with Reduced Search Region”; International Journal of Computer Applications, Vol. 92, No. 12, 27-33, Apr 2014.

[3] Sunghwan Choi and Bumsub Ham. “Space-Time Hole Filling With Random Walks in View Extrapolation for 3D Video”; IEEE Transaction on Image Processing, Vol. 22, No. 6, 2429-2441, Jun 2013.

[4] Li Chen, Hui Lin, and Shutao. “Depth Image Enhancement for Kinect Using Region Growing and Bilateral Filter”; International Conference on Pattern Recognition, 3070-3073, 2012.

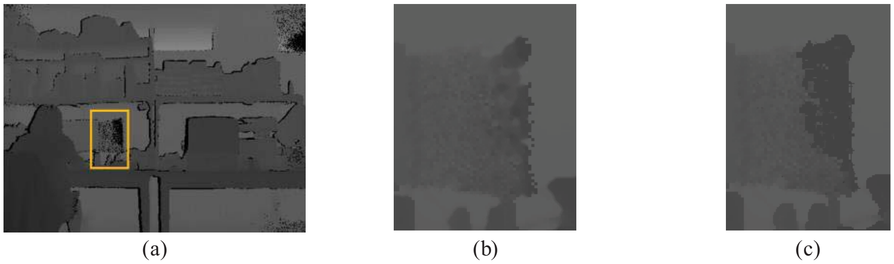


Fig. 2. Depth images: (a) depth image, and corrected depth images using (b) Telea method and (c) proposed method

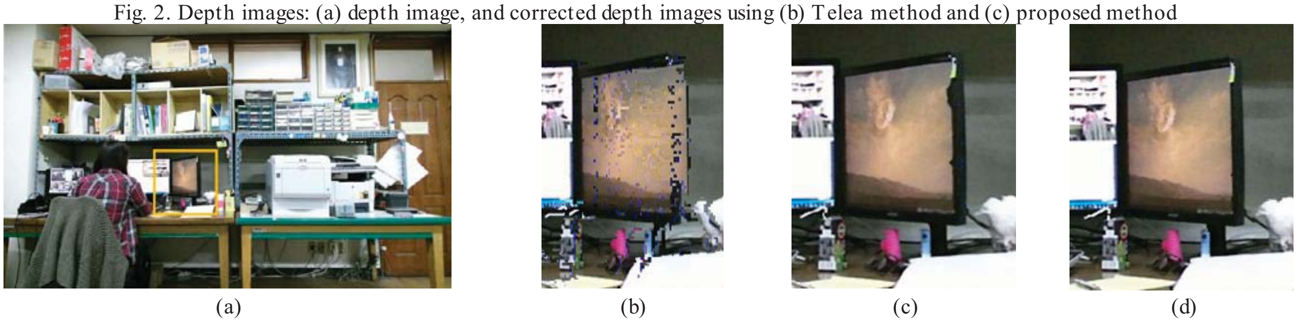


Fig. 3. Creating right-view images: (a) left-view image, and magnified right-view images using (b) original depth with holes, (c) depth using Telea method, and (d) depth using proposed method

INNATE: Intelligent Non-invasive Nocturnal epilepsy Assistive TEchnology

Hossein Malekmohamadi, Jethro Shell, and Simon Coupland
School of Computer Science and Informatics, De Montfort University, Leicester, UK
{hossein.malekmohamadi, jethros, simonc}@dmu.ac.uk

Abstract— *Epilepsy is a neurological disease that affects the brain and is characterised by repeated seizures. Generalised, focal and unknown are three major types of seizures. Each type has several subgroups. For this reason, seizure detection and classification are expensive and erroneous. Other factors can also affect the detection. For example, patients can have a combination of different seizures or start with one type and finish with another. Nocturnal epilepsy can be prominent in many sufferers of this disease. This displays seizures that occur during the sleep cycle. The nature of such seizures makes the gathering of data and the subsequent detection and classification complex and costly. The current standard for seizure detection is the invasive use of electroencephalogram (EEG) monitoring. Both medical and research communities have expressed a large interest in the detection and classification of seizures automatically and non-invasively. This project proposes the use of 3D computer vision and pattern recognition techniques to detect seizures non-invasively.*

Keywords: Epilepsy, Seizure, Kinect 2, Surface normals, PCL

1. Introduction

There are many applications of computer vision and pattern recognition in biomedical engineering. Some related examples are fall detection [1], Parkinson's analysis [2] and the smart bedroom [3], [4]. In these cases, the video system is based on the use of 2D/3D cameras. Within seizure detection, there are also a number of research cases that use RGB-D data. For seizure detection, there are some good examples such as [5], [6]. In some of the aforementioned examples, RGB-D data is obtained using a Microsoft Kinect 1. In [5], the focus is on the rhythmic movements of the patients limbs in a particular seizure phase. The training algorithms in [5], [6] are based on support vector machines and neural nets. The introduced system in [5] utilises coloured pyjamas for patients. The segmentation and tracking of the limbs are much easier but far from a real world scenario. As most individuals sleep covered with a duvet or sheet, nocturnal seizures will not occur in this manner outside of a laboratory setup. In [6], authors used maximum-likelihood detection for low cost and wireless seizure detection. This system is based on the assumption that clonic seizures have periodic movements. In [7], accelerometers are attached to the patients wrist and ankle and video data are used to detect two types of seizure in children. Movement patterns are classified using spatiotemporal features across point of interests. A very good review on vision-based motion detection, analysis and recognition of seizures can be found in [8]. Using infra-red and depth data has improved the results of night activity recognition as shown in [9]. A Microsoft Kinect 1 sensor detects children movements that fall into the defined seizure patterns which leads to an alarm to parents/carers.

Using 3D cameras to capture seizure information has some difficulties in areas such as setting up the correct viewpoint, occlusion or where there is insufficient data during nocturnal seizure due to a duvet or a sheet. For this purpose, we will adopt a sensor fusion approach by incorporating audio signals. In [10], an open source based seizure detector has been introduced with the help of a Microsoft Kinect 1, a smartwatch and audio signals.

There are some important factors in the hardware set-up for detecting seizures. The main issue is that the acquisition devices are very sensitive to ambient noise. Other issues can include the field of view or room geometry. A good example of tackling the field of view with the help of two sensors can be found in [4]. Another important factor in dealing with patients video data is privacy. There are some off the shelf candidates for capturing 3D data such as the Microsoft Kinect 1 and 2. In this paper, we employ the Microsft Kinect 2. In the next section, we describe our methodology and experimental set-up. In Section 3, we conclude this paper and discuss future plans.

2. Method and Experimental Set-Up

In this paper, we use the Microsoft Kinect 2 to collect infra-red and point cloud data. The Microsoft Kinect 2 has a higher resolution for both colour and depth. It also has a bigger horizontal and vertical field of view. The Microsoft Kinect 2 location is above the patient's bed. The data gathered is in a very low light environment to imitate night conditions. The current frame rate to capture data is approximately 7 frames per second (fps). We use the robot operating system (ROS) [11] and Libfreenect2 [12] libraries in an open source environment to record the data. We record ambient sound with the 4 microphones via the Microsoft

Kinect 2. For this purpose, we use the HARK libraries [13] in an open source platform. Point cloud data are fed into the processing stage as shown in Figure 1. A primary step is to pre-process the data before it is moved to the classification of any non-normative behaviour of the individual. The first step of this process is to segment the 3D data into different body parts. This data is then fed into a simple pass-through filter to remove the remaining background. This stage is repeated for each frame. Features are then extracted from keypoints of the point cloud data and from surface normal properties. Finally, we extract dynamic features from motion characteristics of consecutive frames in a time interval. These features are suitable to extract key frames of the whole scene. This part is accomplished using PCL libraries [14]. Additional features include extracting periodic movements in the users limbs. Following feature extraction, we will employ a classification algorithm to the gathered data. A good approach for this stage is a Bag-Of-Visual-Words to classify the seizures
To generate a test set of data, we have used a set of volunteers within a laboratory environment. The project will be expanded to incorporate volunteers with nocturnal epilepsy within real world environments. We recorded data in two conditions: 1) uncovered patient body movements and 2) covered patient body movements which are more realistic as nocturnal seizures happen at night while patients are asleep and covered with a duvet or a sheet. The second problem is one of the most interesting topics among computer vision researchers. Technically, it refers to a rigid body (human body) movements under a deformable dynamic surface (duvet, sheet).

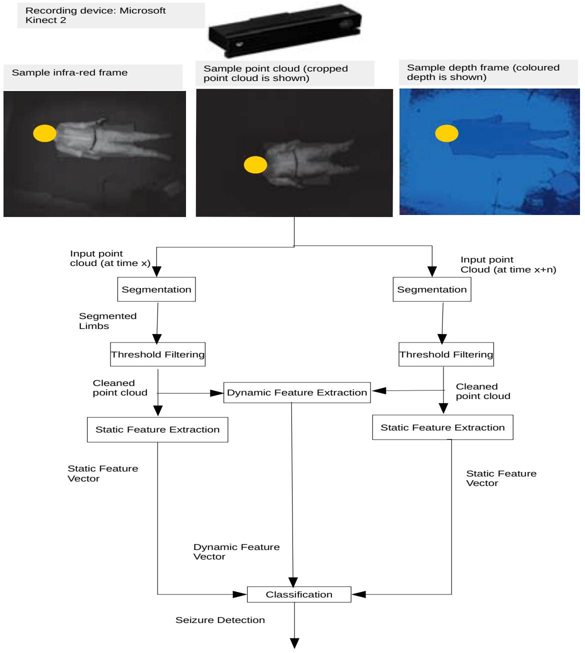

Recording device: Microsoft Kinect 2
Sample infra-red frame
Sample point cloud (cropped point cloud is shown)
Sample depth frame (coloured depth is shown)
Input point cloud (at time x)
Input point Cloud (at time x+n)
Segmentation
Segmentation
Segmented Limbs
Threshold Filtering
Threshold Filtering
Cleaned point cloud
Dynamic Feature Extraction
Cleaned point cloud
Static Feature Extraction
Static Feature Extraction
Static Feature Vector
Static Feature Vector
Dynamic Feature Vector
Classification
Seizure Detection

Fig. 1: *The proposed system for detecting and classifying epileptic seizures.*

3. Discussion and Future Plans

In this paper, we give an introduction to the project INNATE. The goal of this project is to detect and classify epileptic seizures with the help of 3D computer vision and pattern recognition methods. We described experimental set-up to record user data. This initial data is fed into a feature extraction algorithm. In this stage, first, we discard the background and go on to label the limbs using clustering. For each part, we compute features based on surface normals and keypoint features. Moreover, time varying properties are considered as another set of features to be fed into a learning algorithm. We are currently working on a learning algorithm and expanding the dataset. Future plans include gathering data across two scenarios, 1) uncovered body and 2) covered body. We are also producing ground truth data for each scenario through the addition of accelerometers.

References

[1] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 1, pp. 290–301, 2015.

[2] A. P. Rocha, H. Choupina, J. M. Fernandes, *et al.*, "Kinect v2 based system for parkinson's disease assessment," in *Engineering in Medicine and Biology Society, IEEE International Conference on*, pp. 1279–1282, IEEE, 2015.

[3] Y. Booranrom, B. Watanapa, and P. Mongkolnam, "Smart bedroom for elderly using kinect," in *Computer Science and Engineering Conference, 2014 International*, pp. 427–432, IEEE, 2014.

[4] P. Kittipanya-Ngam, O. S. Guat, and E. H. Lung, "Computer vision applications for patients monitoring system," in *Information Fusion, 2012 International Conference on*, pp. 2201–2208, IEEE, 2012.

[5] H. Lu, Y. Pan, B. Mandal, *et al.*, "Quantifying limb movements in epileptic seizures through color-based video analysis," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 2, pp. 461–469, 2013.

[6] L. Cattani, G. Ntonfo, F. Lofino, *et al.*, "Maximum-likelihood detection of neonatal clonic seizures by video image processing," in *Medical Information and Communication Technology, 2014 International Symposium on*, pp. 1–5, IEEE, 2014.

[7] K. Cuppens, *Detection of epileptic seizures based on video and accelerometer recordings*. PhD thesis, PhD thesis, KU Leuven, 2012.

[8] M. Pediaditis, M. Tsiknakis, and N. Leitgeb, "Vision-based motion detection, analysis and recognition of epileptic seizures: a systematic review," *Computer methods and programs in biomedicine*, vol. 108, no. 3, pp. 1133–1148, 2012.

[9] https://blogs.msdn.microsoft.com/kinectforwindows/2014/06/17/keeping-watch-in-the night/.

[10] http://www.openseizuredetector.org.uk/.

[11] M. Quigley, K. Conley, B. P. Gerkey, *et al.*, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[12] L. Xiang, F. Echtler, C. Kerl, *et al.*, "libfreenect2: Release 0.2," Apr. 2016.

[13] K. Nakadai, T. Takahashi, H. G. Okuno, *et al.*, "Design and implementation of robot audition system'hark'open source software for listening to three simultaneous speakers," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2010.

[14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library," in *IEEE International Conference on Robotics and Automation*, (Shanghai, China), May 9-13 2011.

Deep Context Injection for Super-resolution

Int'l. Conf. Image Processing, Computer Vision and Pattern Recognition: ABSTRACT/POSTER

Alex Rinaldi
COMputer Perception LAB (COMPLAB)
California State University, Bakersfield
Bakersfield, CA, USA 93311
Email: alexwgrinaldi@gmail.com

Albert C. Cruz
COMputer Perception LAB (COMPLAB)
California State University, Bakersfield
Bakersfield, CA, USA 93311
Email: acruz37@csub.edu

***Abstract*—As fossil fuel industries decline and the need for renewable energy sources increases, reliable photovoltaic (PV) panels are becoming more vital in society. Maintaining PV panels can be expensive, especially if imperfections, or hot spots, are not revealed early in the production process. Currently, maintenance is carried out manually to detect hot spots. Computers can be used to help automate this process; unfortunately, thermal images with enough quality for meaningful analysis are too expensive to capture with currently available commercial equipment. To address this, we propose a novel super-resolution method called the context injection system, which guides the unsupervised learning process of stacked autoencoders. Results show promise for the enhancement of low resolution images captured with the FLIR Lepton.**

***Index Terms*—Infrared thermography, image super-resolution, deep autoencoders**

I. Introduction

Inspection of PV panels is required to reveal defects that sharply decrease the efficiency of energy production. Infrared (IR) thermography is an amenable medium because thermal properties are an indicator for failure. Super-resolution is proposed as a cost-effective means of enhancing thermal images to reveal these properties. The overview of the system is given in Figure 1.

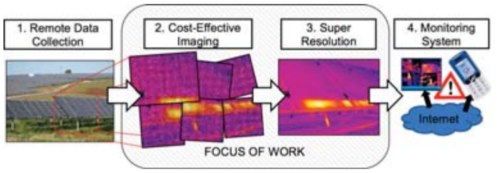


Figure 1. System overview.

Analysis of PV cells must be automatic to both improve objectivity of results and reduce the dangers of human analysis of live power system components. However, in order for heat distribution analysis (or thermography) to be analyzed by computer algorithms, thermal images must be high-quality and detailed. While the FLIR T460 infrared camera offers images with 76,800 pixels - a sufficiently high resolution for automatic analysis - it costs along the order of tens of thousands of dollars, making it cost-ineffective. The recently released FLIR Lepton, a microbolometer for infrared thermography, costs along the order of hundreds of dollars, but it only captures images of 4,800 pixels (80x60). We are presented with the two following technical challenges: (1) Accurately analyzing an image with such low resolution is unfeasible. (2) There are insufficient training examples to learn a mapping from low-resolution to high-resolution for this specific problem.

We propose a system for automatic detection of faults in PV cells. The system will use a novel method for image super-resolution that increases resolution of an image without the need for training examples. The system will accomplish this through unsupervised learning within a deep auto-encoder network. The system is distinctly different from others because we frame the deep network with context injection to guide the learning process.

II. Technical Approach for Research Plan

Several recent studies in super-resolution have proposed learning for super-resolution [1-4]. Learning-based super-resolution can either be supervised or unsupervised. The supervised learning approach begins with high-resolution images, downscales the images and degrades their quality, inputs the resulting low-resolution images into a machine learning system, and tests the accuracy of the reconstruction against the original high-resolution image [4]. This approach is unfeasible for our problem, however, as producing high-resolution thermal images would be far too expensive. Additionally, there is not a sufficient amount of publicly available data on infrared thermography to train a supervised deep learner. Instead, our approach will focus on unsupervised learning, which attempts to enhance an image using only the image data itself. One approach has been explored using the example-based approach [1], which attempts to reconstruct a high resolution image by selecting similar texture patches of high-resolution images and combining them into a full-size image. This approach is unfeasible because there is no cost-effective method for producing high-resolution example images. A more viable approach to unsupervised learning uses a denoising auto-encoder. Denoising auto-encoders attempt to discover high-level features of image data by training a single algorithm to produce output that matches its input (reproduction) over a sufficiently wide range of input images. Using the auto-encoder's ability to reveal high level features of image data in a super-resolution system is consistent with the self-similarity approach to super-resolution [5], which suggests that high-level features reoccur in the same image (such as repeating objects and shapes). This correlation was explored in [2] with encouraging results.

A. Proposed Method: Deep Context Injection for Super-resolution

We will build upon this approach with a deep learning auto-encoder solution that utilizes a context injection system. The context injection system is structured as follows: multiple auto-encoders are stacked to form a large network. We will guide the learning process by providing the auto-encoder with high-level image data organized in levels of abstraction similar to the layers of the human visual cortex; since each successive layer in a stacked auto-encoder is meant to provide a higher-level abstraction of the data, higher-level image data–such as shape and edges (image moments from segmentation) will be injected later in the encoding cascade than lower-level data–such as the texture features and intensity order statistics. We predict this will guide the self-discovery process at each layer. An overview is given in Figure 2.

Similar to other super-resolution attempts [1-3], a single low-resolution image will be up-scaled to the desired resolution using interpolation and then split into local patches using an extraction matrix. Each patch will be inputted into the stacked auto-encoder. Higher-level image data will be input at successive layers. The final output of the stacked auto-encoder will be the reproduced patch. After training, the patches will be recombined to form the final output.

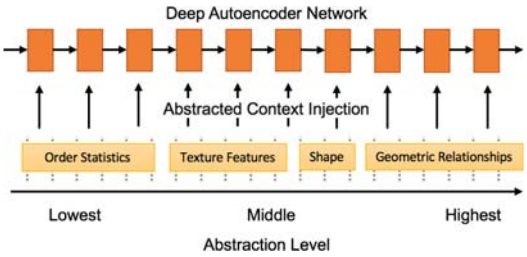


Figure 2. Overview of context-injection system.

The structure of the context-injection system will follow a convention for stacked denoising auto-encoders provided in [6]. Training the auto-encoder will be split into pre-training and fine-tuning phases. During pre-training, each layer of the stacked auto-encoder is trained separately to reproduce the input of the previous layer, with the initial layer's input being the pixel data itself. During fine tuning, all units in the network will be trained to minimize the difference between the input patch and the reconstructed output patch from the final layer. In both phases, the input to each layer and appropriately abstract high-level image data are encoded together, before being decoded into a reconstruction of the original input.

III. Preliminary Results and Conclusion

An early version of the unsupervised super-resolution system has shown promising results (see Figure 3). The preliminary version excludes the context-injection system, providing a conventional implementation of the stacked denoising auto-encoder. The preliminary system was tested with a thermal image of a solar panel captured with a FLIR Lepton. The image was up-scaled 2x with cubic interpolation, and then split into 5 24x24 pixel patches. The set of patches was used as an input set for the stacked-auto encoder. At the end of the training period, the auto-encoder network was able to produce output patches that visibly resemble the input when reconstructed. The inputs and outputs to the system are visible below at 50% size.

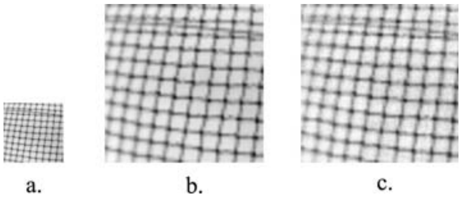


Figure 3. Inputs and outputs to the preliminary system downscaled to 50% size. a. The original image; b. The image up-scaled with cubic interpolation; c. The reconstructed output from the system. This figure should be viewed as soft copy. The proposed method (c) demosaics the result of interpolation.

The quality of the previous output contrasts that of an initial experiment, which used a larger set of example imagesas input for the auto-encoder network. This diverse set of data more closely resembles data used in the training process for a supervised system, as in the example-based approach [1] and the supervised system proposed in [4]. Without any additional constraints, the auto-encoder is unable to find common high-level features among the set of example images; the loss function does not converge using gradient descent, and the output image resembles an average of all the images in the example set (visible below). This reinforces the necessity for a super-resolution system that utilizes unsupervised learning as opposed to supervised. Future work in the research plan involves implementation of the context injection system.

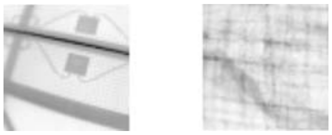


Figure 4. (Left) Input to and (Right) output from the system trained with a diverse image set.

IV. References

[1] W. T. Freeman et al., Example-based super-resolution, IEEE Comp. Graph. Appl., 2002.
[2] Z. Cui et al., Deep Network Cascade for Image, ECCV, 2014.
[3] C. Dong et al., Image Super-Resolution Using Deep Convolutional Networks, IEEE PAMI, 2015.
[4] Y. Liang et al., Learning visual co-occurrence with auto-encoder for image super-resolution, 2014 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA 2014, 2014.
[5] D. Glasner et al., Super-Resolution from a Single Image, Signal and Info. Proc. Assc. Annual Summit Conf., 2014.
[6] P. Vincent et al., Extracting and composing robust features with denoising autoencoders, ICML, 2008.

Facial Emotion Recognition for Motor Vehicle Operators

International Conference on Image Processing, Computer Vision and Pattern Recogition: ABSTRACT/POSTER

Geromar Hasta
COMputer Perception LAB (COMPLAB)
California State University, Bakersfield
Bakersfield, CA
ghasta@csub.edu

Albert C. Cruz
COMputer Perception LAB (COMPLAB)
California State University, Bakersfield
Bakersfield, CA
acruz37@csub.edu

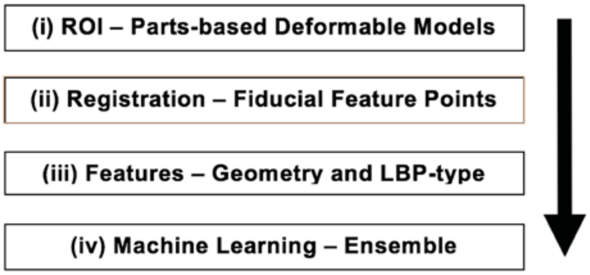
***Abstract*—Key factors of motor vehicle accidents are inattention and stress. Current technology lacks ability to track these factors. Using a single front-facing camera, a computer vision system can alert drivers by inferring mental state from expressions. We have used these techniques as a metric rating system to determine the Motor Trend Best Driver Car of the Year. This system has four phases: (i) Region of interest, (ii) registration, (iii) extraction of geometric and local-appearance features, and (iv) machine learning. Preliminary results with driver videos yields a correlation of 0.834. This result shows promise and future work will improve the accuracy of the system in testing.**

Keywords—Facial emotion recognition, facial feature extraction, face detection

I. Introduction

According to the U.S. Center of Disease Control and Prevention (CDC) motor vehicle accidents (MVA) are the leading cause death of teenagers in the U.S. Key factors to MVA are the driver's inattention and stress level. We propose a preventative strategy to track facial expressions from frontal face video. The driver's psychophysiological state are inferred to alert at-risk drivers. Currently, there is no system to date which can sufficiently understand human facial expressions for stress and attention for the analysis of human vehicle operators, from a single camera [1]. This is due to the shortcomings of the software algorithms for video frame processing. The focus of this work is a novel computer vision system that recognizes facial expression cues to detect any signs of stress, fatigue, or inattention. Initial results that analyze data from Motor Trend Magazine's Best Driver Car of the Year found that the major technical problems are: (1) the inability to accurately scale-up the system from training to testing live emotions and, (2) specifically, the inability to detect the frontal face from video frames. In this research plan, we discuss the applicability of state of the art methods, discuss future plans for building upon state of the art work, and give preliminary results. Example frames are given in Figure 1.

Figure 1: Sample frames from video.



(i) ROI – Parts-based Deformable Models

(ii) Registration – Fiducial Feature Points

(iii) Features – Geometry and LBP-type

(iv) Machine Learning – Ensemble

Figure 2: System overview. LBP: Local binary patterns. ROI: Region-of-interest.

II. Technical Approach and Research Plan

This system has four phases: (i) Region of interest extraction. An algorithm must locate the face within the image frame for analysis. This is a difficult challenge to contemporary approaches because of occlusion from the helmet and glasses. To this end, parts-based deformable models were developed and tested for face detection. (ii) Registration. After extraction, faces from phase (i) were aligned with fiducial feature points to ensure alignment of facial features across the data. (iii) Extraction of geometric and local-appearance features. The images are then succinctly represented in terms of facial point spatiotemporal locations. We use post-alignment positions of the facial points as features. These features are combined with Local Binary Pattern type features as a texture classifier. (iv) Machine learning. Finally, the frame is compared to training data of other videos that have been expertly labeled with psychophysiological indicators to make a prediction of the current frame. An overview of the method is given below in Figure 2.

The first phase of detection of emotion from facial features is the extraction of data points corresponding to the face. We investigated two methods for extracting the face: parts-based deformable models (Constrained Local Models) and parameterized appearance models (Supervised Descent Method). Both methods showed promise for detecting the region of interest with 70% detection rates. Constrained Local Models (CLM) [2] maximizes the likelihood of a set of data points represented as a mixture of trees. Supervised Descent Method (SDM) [3] fits the facial points in a supervised way with gradient descent. It was found that the model used in CLM provided a better track in general, but Supervised Descent Method provided better quality results when it worked. For this reason we use CLM. In the future we will build upon this work by incorporating supervised descent with Constrained Local Model's mixture of trees model for better face tracking.

After face tracking, facial configuration is aligned to specific registration points. State of the art methods are supervised and have excessive computational cost [4]. This can be avoided by carrying out a point-based alignment of the image for extraction of local appearance features. The three-dimensional points from tracking are used for face registration. Fiducial feature points are selected, e.g. the inner eye corners and the tip of the nose, because these points are the least effected facial muscle movement.

Facial features can be categorized into two types: geometric and local appearance [5]. Geometric features describe the shape of the face and relationship between facial parts. For quantification of geometric data, we take the aligned data points transformed in the previous step. In local appearance features, low-level intensity values and textures are captured. In this work uniform local binary patterns (LBP) were used and, in the future, spatiotemporal LBP features such as Volumetric LBP and Three Orthogonal Planes will be considered [6].

Finally, the last phase of emotion detection is classifying the samples with a regression ensemble. A regression ensemble will use a set of weak learners to develop a prediction for the extracted data. The results from this process will reveal the predicted valence and attention levels of the video frame. A support vector machine was also considered but it achieved a performance of only 0.7665 correlation. Fig. 4 shows the comparison of the computer predicted data with data examined by human experts. The results show promise by yielding with a high correlation between the predicted and examine values. The valence and attention values detected will depict the results of the driver's current state of emotion and can reveal their true expression. Valance in emotion is described by how attracted or aversive an individual is in a certain task or situation. Positive attractive valance corresponds to joy and happy emotion. Conversely, negative valence corresponds to emotions such as anger and stress.

III. Preliminary Results

Preliminary results from training data showed a promising correlation of 0.8343 in training. The system shows promise for preventing unnecessary MVA and further casualties in the young adult population by providing the vehicle to assist the driver when they are in poor psychophysiological states.

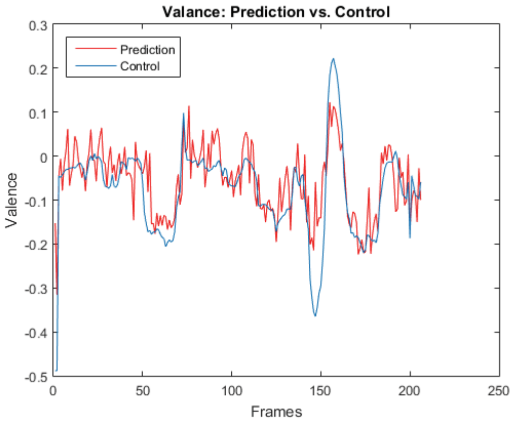


Figure 4: Machine Prediction compared to Control

IV. Conclusion

The emotion detection algorithm shows promise is predicting the valance and attention levels of vehicle drivers. This project will continue to improve processing time to be able to calculate affect on a real-time basis to offer live feedback for the driver. Other areas for improvement can be through the proficiency of facial detection and reducing the error from occlusion of gestures.

V. References

[1] A. Tawari, S. Sivaraman, M. M. Trivedi, T. Shannon, and M. Tippelhofer, "Looking-in and looking-out vision for Urban Intelligent Assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 115–120, 2014.

[2] S. Cheng, A. Asthana, S. Zafeiriou, J. Shen, and M. Pantic, "Real-time generic face tracking in the wild with CUDA," *ACM Multimed. Syst.*, pp. 148–151, 2014.

[3] X. Xiong and F. De La Torre, "Supervised descent method and its applications to face alignment," in *Proc. Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 532–539.

[4] S. Yang and M. Kafai, "Zapping Index : Using Smile to Measure Advertisement Zapping Likelihood," vol. 5, no. 4, pp. 432–444, 2014.

[5] A. Cruz, B. Bhanu, and N. Thakoor, "Vision and Attention Theory Based Sampling for Continuous Facial Emotion Recognition," *IEEE Trans. Affect. Comput.*, vol. PP, no. 99, pp. 1–1, 2014.

[6] T. R. Almaev and M. F. Valstar, "Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition," *Proc. - 2013 Hum. Assoc. Conf. Affect. Comput. Intell. Interact. ACII 2013*, pp. 356–361, 2013.

Calibration method Of Stereo Camera for Vehicle

Byoung-Ik Kim[1], Jang-wook Choi[1], Kyung-jin Na[1], and Soo-Young Ha[1]
[1]Advanced Research Team, AJIN industrial co., LTD., Gyeongsan-si, Gyeongsangbuk-do, Korea

Abstract – ***Camera calibration is an important prerequisite for recognizing 3D information of scene. In this paper, we proposed a more accurate and effective calibration method of stereo cameras mounted on the vehicle. Proposed method presented high precision alignment method of stereo camera. This method represents a camera calibration method which can overcome the limitations of the alignment mechanism.***

Keywords: *stereo camera, alignment mechanism, camera calibration.*

1 Introduction

Recently, the developed countries are competitively developing the fusion technique of advanced sensors such as radar, LIDAR and stereo camera. Especially, techniques using HD stereo camera are developed to correctly recognize pedestrians, vehicles, road conditions, etc. these fusion techniques to monitoring in real-time a situation around the vehicle to reduce traffic accidents caused by careless driving. Presently, driver assistance system using stereo camera were some commercialization and were focused on research and development to meet the laws of safety regulation to attach obligatorily. However, a stereo camera that is employed in the driver assistance system is mounted interposed between the room mirrors inside the vehicle. in this case, these will have respectively different tilt, rotation, movement, etc. the particular feature of Images obtained from the stereo camera, which is located in front of the vehicle should be positioned on the same horizontal line. Because, these data the distance estimated by the stereo image analysis is characterized by having a high precision. However, it is very difficult to accurately align the right and left cameras mechanically. In particular, the mechanical alignment of high-precision in the stereo vision greatly increases the time and cost. Because this operation is required the skill of the operator and the machining precision of the mechanism [1].

In this paper, we proposed a more accurate and effective calibration method of stereo cameras mounted on the vehicle. The proposed method contribute to align mechanically prior to installing the stereo camera to a vehicle. At same time, this method represent stereo camera calibration method and apparatus to can be overcome mechanism alignment problem.

2 Proposed calibration method

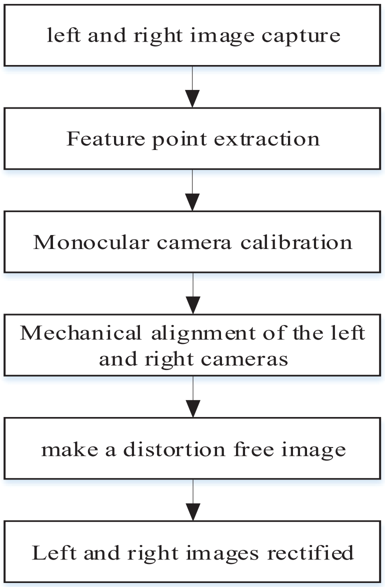
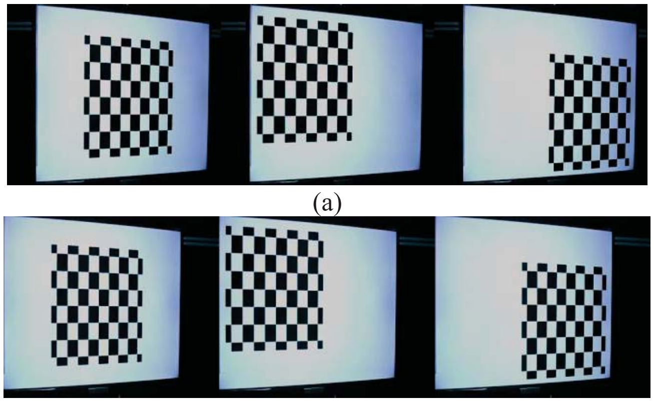

left and right image capture
Feature point extraction
Monocular camera calibration
Mechanical alignment of the left and right cameras
make a distortion free image
Left and right images rectified

Fig. 1. Proposed method for stereo camera calibration.

Figure 1 shows the proposed method for stereo camera calibration. Proposed method consist of six steps, the first is to acquire left and right image, the second is to extract feature point, the third is to do monocular camera calibration, the fourth is to do mechanical alignment of left and right camera, the fifth is to make distortion-free image, and the final is to perform alignment of left and right image.

Image acquisition using left and right cameras are performed for chess board having variety posture. Chess board is flat plate to be used for correction. This step should acquire a lot


(a)

(b)

Fig. 2. Chess board images; (a) left images, (a) right images.

of images having variety posture to ensure more accurate correction. Feature point in images is detected analyzed each of the left and right image, as shown in Figure 2. Monocular camera was calibrated respectively cameras using a monocular camera algorithm based on the extracted feature point while the distortion parameters and the internal and external parameters of the each cameras were acquired and were saved [2],[3].

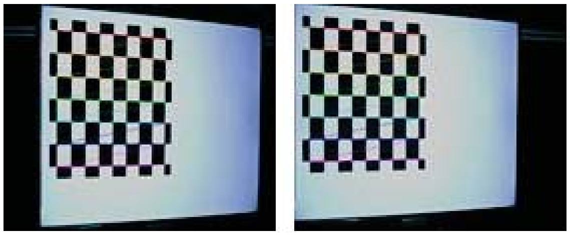
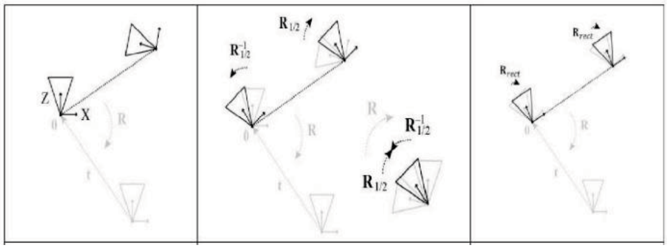

(a) (b)

Fig. 3. Feature points detection image; (a) left image, (a) right image.

Mechanical alignment of camera was performed for using the result which was calculated relative movement and rotation of between right and left camera using the external parameters calculated by calibration algorithm of monocular camera. If each cameras are calibrated using 10 of image pairs, movement and rotation are calculated for each image pair. Initial value is selected median value among the calculated values. And this method optimizes the movement and rotation by the LM method to minimize re-projection error of the feature points on a chess board. Finally, the right and left cameras are aligned mechanically using the Euler angle that was computed from the elements of the calculated rotation matrix R.



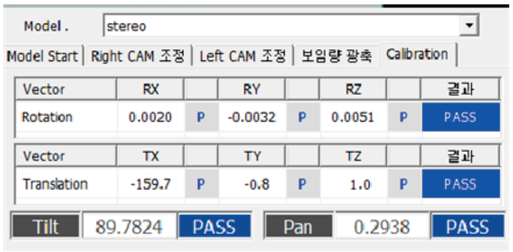
(a) (b) (c)

Fig. 4. Alignment using the relative rotation and movement of the camera; (a) before alignment, (a) the first alignment, (c) the second alignment.

The distortion free images is produced by removing the radial and tangential distortion using distortion parameter and internal/external parameter acquired and stored through the monocular camera calibration. Finally, left and right images are rectified using the relative rotation and movement of the camera calculated through a mechanical alignment, it takes place the vertical position of the left and right images of a non-distorted image.

3 Experiments

The experiment was conducted according to the following procedure. 1) The base plate is mounted on the optical axis calibration equipment, 2) is to adjust the focus of the monocular camera to have a uniform resolution, 3) is to acquire chess board images having variety posture for calibration, 4) Calibration results confirmed. Experiment results could confirm that mechanism alignment problem overcomes.



Model . stereo

Vector	RX		RY		RZ		결과
Rotation	0.0020	P	-0.0032	P	0.0051	P	PASS

Vector	TX		TY		TZ		결과
Translation	-159.7	P	-0.8	P	1.0	P	PASS

Tilt	89.7824	PASS	Pan	0.2938	PASS

Fig. 5. Calibration results

4 Conclusions

In this paper, we proposed a more accurate and effective calibration method of stereo cameras mounted on the vehicle. The proposed method contribute to align mechanically prior to installing the stereo camera to a vehicle. At same time, this method represent stereo camera calibration method and apparatus to can be overcome mechanism alignment problem.

This work was supported by ICT R&D program of MSIP/IITP. [1005438, Development of multi-function collision avoidance ADAS SoC and camera system supporting EURO-NCAP2020]

5 References

[1] Hongshan YU and Yaonan Wang, "An Improved Self-calibration Method for Active Stereo Camera," Proceedings of the 6th World Congress on Intelligent Control and Automation, vol. 22, no. 11, pp. 5186-5190, 2006

[2] Zhang, Z., "A Flexible New Technique for Camera Calibration," Technical Report, Microsoft Research, Mar. 1999

[3] Zheng you Zhang, "A Flexible New Technique for Camera Calibration," IEEETransactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, 2000

Wide Dynamic Range CMOS Image Sensor with Dual Mode Operation

Sanggwon Lee, Byung-Soo Choi, Myunghan Bae, and Jang-Kyoo Shin*

School of Electronics Engineering, Kyungpook National University,
80 Daehak-ro, Buk-gu, Daegu 41566, Korea

Phone: +82-53-950-5531, E-mail: jkshin@ee.knu.ac.kr

Abstract

This paper presents a wide dynamic range complementary metal oxide semiconductor (CMOS) image sensor (CIS) using column capacitors and feedback structure. The prototype sensor is designed and fabricated by using 0.18μm standard CMOS technology. This sensor is operated under dual mode scheme; an active pixel sensor (APS) mode for normal light condition and a passive pixel sensor (PPS) mode for high light condition. For checking the light conditions, a node capacitance controlled by the reference voltage is added in every column line. By utilizing the proper sensing mode depending on the light conditions, the dynamic range can be adjusted with a linear response. By using the proposed structure, it is possible to store more electric charge, which results in a wider dynamic range. The simulation and measurement results demonstrate the feature of wide dynamic range (WDR).

Keywords: CMOS image sensor, linear response, dual mode, feedback structure, wide dynamic range

Recently, CMOS image sensors (CISs) have been implemented in various applications such as safety systems and security monitors [1]. WDR CISs are required for the camera market of security systems to capture lots of image information. Therefore, the characteristics of image sensor are more important than before. However, the dynamic range of conventional CIS is lower than 70dB, and thus cannot capture a scene both bright and dark conditions. So, a lot of researches have been tried in order to improve dynamic range [2].

We propose a method to extend the dynamic range of a CIS using the dual mode operation technique. The pixel is operated in the APS mode under normal-light conditions and the PPS mode under high-light conditions using column capacitors, which are equalized for detecting high-light information. In addition, the use of a feedback circuit is helpful in improving high-light detection. Therefore, the combination of the APS mode and the PPS mode leads to a wider dynamic range.

Fig. 1 shows the schematic of the proposed pixel and the readout circuit. To achieve dual mode operation, the pixel is designed with four transistors based on the 3T structure. The proposed CIS structure consists of a pixel array, a scanner, an upper column circuit, and a lower column circuit. By using lower column circuit, the proposed image sensor can be captured high light imaging without saturation.

The timing diagram and potential well diagram of proposed image sensor are shown in Figure 2 and Figure 3, respectively. Based on the light conditions, the operation mode are decided. In APS mode, the pixel operates using M1, M2, and M3 transistors and the switch S_1 is turned on. The photocurrent is integrated in FD node capacitor at normal light conditions as shown in Fig. 3(a). When the pixel is saturated as shown in Fig. 3(b), the PPS mode is operated. In PPS mode, the pixel operates using M1 and M4 transistors and switch S_2 is turned on. The photocurrent is integrated in integration node using column capacitors as shown in Fig. 3(c). Fig. 3(d) shows the potential well with a reference voltage of 0.5 V. When the reference voltage is biased, the charge is divided by the feedback capacitor immediately. From this mechanism, the dynamic range is controlled by the reference voltage with the feedback comparator.

Fig. 4(a) shows the simulation results for the pixel output when the photocurrent is varied. Fig. 4(b) shows the measurement results when the light source is varied from low light conditions to high light conditions. Using the proposed method, the image sensor can be captured over 12,000 [lux] light intensity. Fig. 5 shows the chip layout. A CIS with a chip area of 1.8 mm × 2.1 mm is designed with a resolution of 176 × 144 (QCIF) using a 5.6 × 5.6 μm^2 pixel pitch and it is fabricated with a 0.18-μm 1-poly 6-metal (1P6M) standard CMOS process.

References

[1] E. Raphael, R. Kiefer, P. Reisman, and G. Hayon, “Development of a camera-based forward collision alert system,” SAE International Journal of Passenger Cars, Vol.4, No.1, pp.467-478, Apr., 2011.

[2] N. Akahane, S. Adachi, K. Mizobuchi, and S. Sugawa, “Optimum design of conversion gain and full well capacity in CMOS image sensor with lateral overflow integration capacitor”, IEEE Trans. Electron Devices, Vol. 56, No. 11, pp. 2429-2435, 2009.

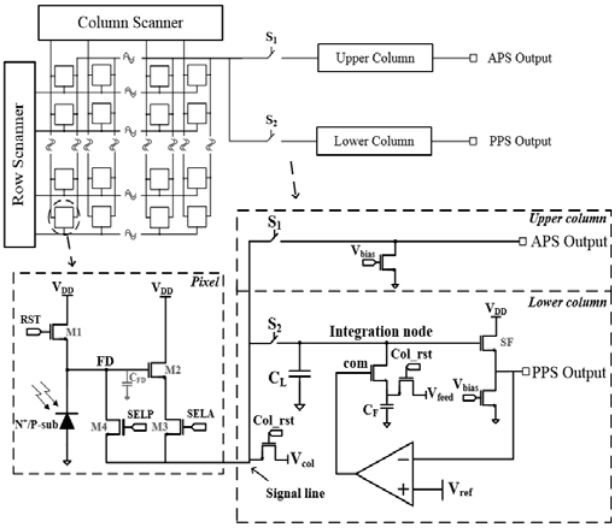


Fig. 1. Structure of the proposed pixel and readout circuit.

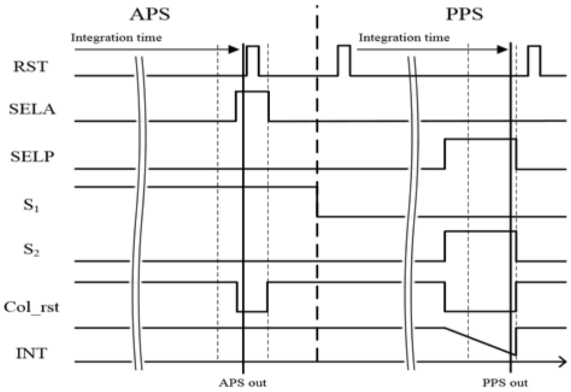
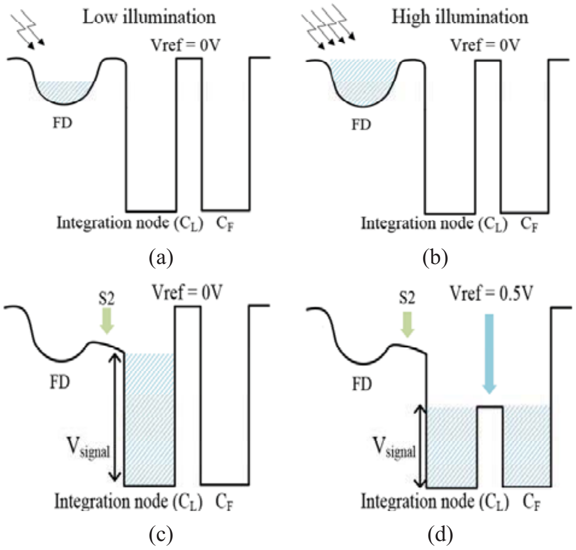


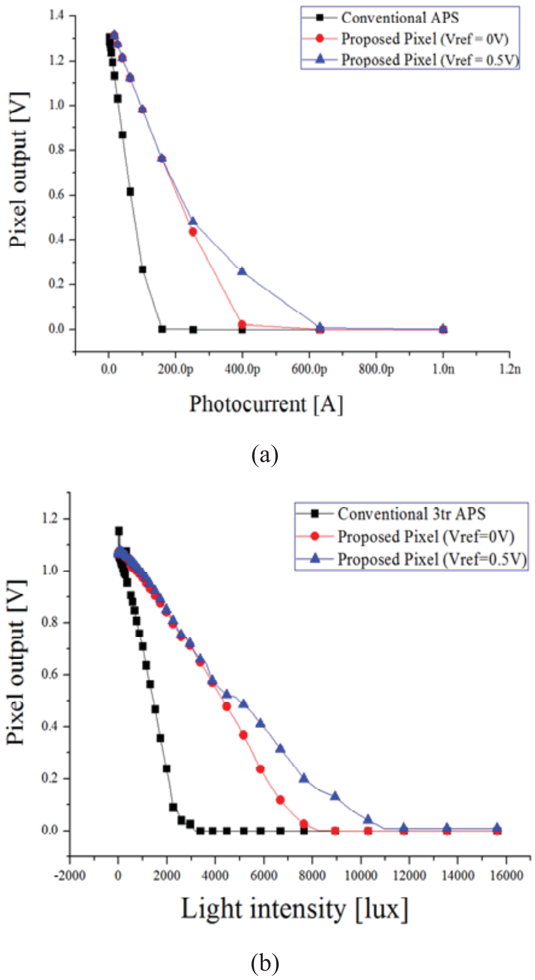
Fig. 2. Timing diagram of the proposed image sensor.



(a) (b)

(c) (d)

Fig. 3. Potential well diagram of proposed image sensor.



(a)

(b)

Fig. 4. Pixel output of (a) simulation result variation of sweeping photocurrent and (b) measurement result variation of controlling comparator reference voltage.

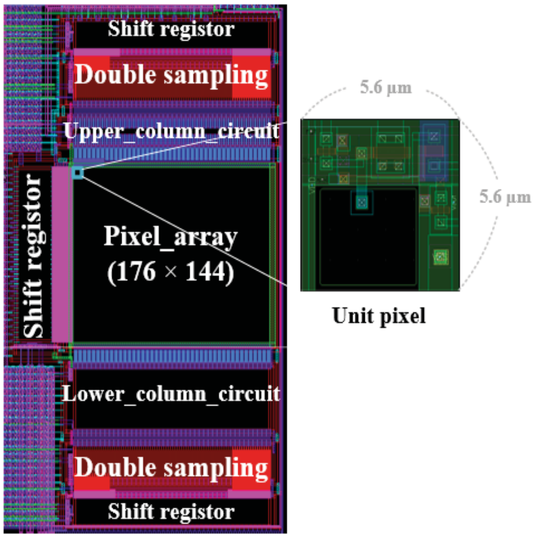


Fig. 5. Layout of the proposed image sensor.

Feature Reduction in Heterogeneous Data Sets via Sequential Search Techniques

Ronald C Anderson[1,2], **Mary C. Baker**[1]
[1]Department of Electrical & Computer Engineering, Texas Tech University, Lubbock, TX, USA
[2]Department of Math & Computer Science, Thiel College, Greenville, PA, USA
Correspondence: ronald.c.anderson@ieee.org

Abstract – *In the clinical setting, pattern recognition techniques promise significant improvement for disease state detection. However, many medical conditions are not reliably described by a single defining feature. In such heterogeneous groups, capturing synergistic effects among features becomes vital to classification success. Additionally, many commonly employed feature selection techniques, such as statistical tests, may not work well when applied to problems involving heterogeneous groups. This treatise explores the application of Sequential Forward Floating Search (SFFS) techniques in the high-dimensional data sets common to the neuroimaging field. Using both multimodal neuroimaging data and calibrated synthetic data sets, the project seeks to characterize SFFS's effectiveness versus established statistical approaches common in the literature.*

Keywords: EEG; Feature Selection; Sequential Forward Floating Search; SFFS

1 Introduction

1.1 The Challenge of Heterogeneous Data Sets

In certain clinical diagnoses, such as autism, multiple features may characterize the illness, indicating a wide range of symptomology. For instance, in a hypothetical sample of 20 autistic subjects, eleven may exhibit verbal deficiencies, seven may exhibit abnormal lack of empathy, and four may exhibit heightened sensitivity to sound. Some members of the group may exhibit more than one of the properties. These three features are thus characteristic of the group, but no one feature is a strong classifier for all members. This simple example illustrates a heterogeneous data set: the autistic class is comprised of several subgroups.

The best classification scheme, then, should be one where features are considered in combination, rather than individually. Unfortunately, many preferred methods for feature selection ignore synergistic effects among groups. The common t-test and Wilcoxon rank sum tests are examples. Also, the presence of subgroups in the data could "skew" the feature vector away from the near-normally distributed property assumed by many statistical tests, reducing their effectiveness or invalidating it.

1.2 The Sequential Forward Floating Search

Pudil, et. al., first presented the SFFS technique to the pattern recognition community in 1994 [1]. Since its inception, it has seen many adaptations and modifications to improve its effectiveness for certain problem classes [2].

SFFS is a wrapper method, meaning the classifier's objective function is evaluated as part of the search process. This bestows several added benefits, including the ability to assess multiple features in a synergistic manner. Also, the underlying objective function can be optimized to the classification problem at hand. Both of these benefits are essential to this dissertation.

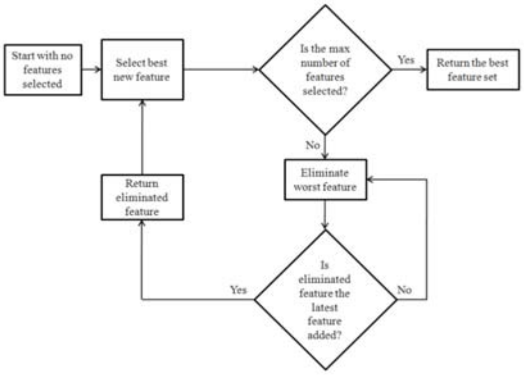

Start with no features selected
Select best new feature
Is the max number of features selected?
Yes
Return the best feature set
No
Eliminate worst feature
Return eliminated feature
Is eliminated feature the latest feature added?
Yes
No

Figure 1: SFFS Flowchart [2]

One basic implementation of the SFFS algorithm is illustrated in Figure 1. The inputs to the algorithm are a set of *N* features to search and an integer *K* indicating the maximum number of features that may be chosen. The method output is a subset of the input features (no more than *K*) that provides the best classification rate seen during the search. The value *K* is chosen based on the problem and serves two purposes: dimensionality reduction and prevention of overfitting the data.

2 Literature Overview

2.1 SFFS Applications in Neuroimaging

Despite SFFS's popularity in many fields of research, including microarray analysis and remote sensing, its use in the neuroimaging discipline has been minimal to date [2][3][4][5][8].

2.2 SFFS Application to Heterogeneous Data

General studies comparing SFFS to other methods have been conducted[6][7]. Hua et. al. present a thorough study of SFS and SFFS with first stage feature pre-filtering using t-tests or ReliefF in microarray analyses [7]. Their studies use both real data and synthetically-generated data with possible (mutually-exclusive) subgroups within the feature vector. However, the feature vector sizes in this study range from 8,000 – 20,000. Currently, this is beyond the size range for most neuro-imaging datasets. Also, their algorithms were trained using 60 or more sample members, which is an unobtainable goal for many neuroimaging studies. Last, the addition of pre-filtering instills concern over the survivability of subgroup integrity after filtering, possibly interfering with classification.

3 Research Objectives

The impact of in-class subgroups on SFFS classification effectiveness is not well understood. Also, a sound understanding of algorithm effectiveness should come from a combination of trials on real data as well as carefully-tuned synthetic data. To date, this combination of practicality and ground truth does not exist in the neuroimaging literature.

Hence, the research objectives can be formulated as follows: generate synthetic data sets that adequately simulate the presence of subgroups, such as those that could be found in neurological disease states; quantitatively establish the effectiveness of SFFS on those synthetic data sets when compared with statistical methods; apply the algorithms to actual multimodal neuroimaging data to assess the impact of algorithm optimizations.

4 Methodology

4.1 Synthetic Data Set Formulation

Synthetic data set formulation is fundamental to this project. A ground-truth assessment of algorithm performance requires input classes of known parameterization. The data will be generated from several distribution types and incorporate multiple subgroups spanning features. These synthetic data can then be processed via the common EEG/fMRI analysis tools to generate features for classification.

4.2 Evaluation of Methods on Synthetics

Synthetic data can be used to generate many features for classification with SFFS and typical statistical approaches. Since the input data are deliberately generated with given properties, the outcomes from each method can be used to assess the merits of each feature selection technique. Preliminary results from analyses on heterogeneous normally-distributed synthetic features indicate that SFFS holds advantage over the t-test in identifying embedded subgroups.

4.3 Application to Multimodal Data Sets

Last, the application of finalized SFFS classification systems to real multimodal neuroimaging data sets will be explored. For this step, task-oriented and resting state EEG/fMRI are the expected approach problems. This stage of the research is crucial since over-reliance on synthetic data could lead to algorithms unfit for use on biologically complex data with which they are expected to deal.

5 Conclusions

Key objectives of this project are to establish the impact of subgroups in heterogeneous data on SFFS and statistical approaches by employing a combination of simulated data sets and multimodal neuroimaging data. The hypothesis is that SFFS methods can be optimized for classification in problems where subgroup structure requires considering multiple features together.

6 References

[1] Pudil, Pavel, Jana Novovičová, and Josef Kittler. "Floating search methods in feature selection." *Pattern recognition letters* 15.11 (1994): 1119-1125.

[2] Kerr, Andy. “An EEG Feature Selection Toolbox for EEGLAB in the MATLAB Environment”, Master’s Thesis (2011)

[3] Akrofi, Kwaku, et. al.. "Classification of Alzheimer's disease and mild cognitive impairment by pattern recognition of EEG power and coherence." *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010.

[4] Baker, Mary C., et. al.. "An SFFS technique for EEG feature classification to identify sub-groups." *Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on*. IEEE, 2012.

[5] Burrell, Lauren, et. al.. "Evaluation of Feature Selection Techniques for Analysis of Functional MRI and EEG." *DMIN*. 2007.

[6] Balli, T., and R. Palaniappan. "Classification of biological signals using linear and nonlinear features." *Physiological measurement* 31.7 (2010): 903.

[7] Hua, Jianping, Waibhav D. Tembe, and Edward R. Dougherty. "Performance of feature-selection methods in the classification of high-dimension data." *Pattern Recognition* 42.3 (2009): 409-424.

[8] Yang, Ye, Ranadip Pal, and Michael O'Boyle. "Classification of cognitive states using functional MRI data." *SPIE Medical Imaging*. International Society for Optics and Photonics, 2010.

SESSION

LATE BREAKING PAPERS

Chair(s)

TBA

2D Hand Tracking with Motion Information, Skin Color Classification and Aggregated Channel Features

J. H. Hammer[1,2], C. Qu[1,2], M. Voit[2], and J. Beyerer[2,1]
[1]Vision and Fusion Laboratory, Karlsruhe Institute of Technology Karlsruhe, Germany
[2]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany

Abstract— *In this paper we present our latest approach for 2D hand tracking in video streams of head-worn monocular color cameras. Those are lightweight and embedded in many head-mounted devices (HMDs) like eye trackers, Augmented Reality glasses or Virtual Reality devices to capture the field of view from the ego perspective. Interaction with virtual elements of the augmented or virtual reality or the interaction with the device itself can be intuitively performed using the hands. Our new approach called AfM fuses our previous method called Motion Segmentation and Appearance Change Detection based Skin color detection (MACS) with the results of a hand detector using Aggregated Channel Features. It tracks the hand more robustly and reduces the deviation from the ground truth paths by 50 % on our benchmark with more than 25,000 frames consisting of different hand gestures.*

Keywords: hand tracking, hand gestures, object detection, skin color, optical flow

1. Introduction

Virtual reality (VR), as it is realizable already today, is going to change how we will be entertained, how we will learn and how we will collaborate. One day Augmented Reality (AR) devices will have overcome their current limitations of weight, small field of view and unimpressive visualization capabilities. Those devices will replace all sorts of auxiliary monitors as tablets or smartphones which are currently often used e.g. in industry or museums to bring information closer to the user. The user then has to capture this information and transfer it to the real object. With AR the step of transferring the information is not needed any more, since information can be visualized referenced to the world directly in the field of view on the objects. AR glasses have already proven its necessity in different applications like e.g. the picking process in warehouses[19] or industrial assembly since the hands are free when no device needs to be hold in the hands.

When the visualization capabilities of AR glasses will be impressive enough in the near future one wants to interact with the virtual contents visualized in front of oneself. Hand movements and gestures are the most intuitive modality for manipulating such virtual contents. In today's VR applications controllers are used instead of the hands. Having no need for those controllers would be a great benefit. Therefore, the hands must be tracked reliably even in difficult environments with highly challenging lighting conditions. Many researcher focus on hand pose estimation using depth data [1], [2], [21], [22], but in both, industrial and commercial applications, the weight of the HMD is one of the most important factors to allow for daily usage in real life applications. Stereo cameras and depth sensors always add additional weight and especially construction complexity. That is why many VR devices have only one monocular camera embedded. Hence, monocular hand tracking remains important to be focused on [3], [5], [6], [7], [20].

In the following sections we will first concentrate on related work (Section 2). Then we will summarize one of our previous 2D hand tracking approaches that yields as basis of the work on hand in Section 3.1. Afterwards we present the created hand detector in Section 3.2 and show in Section 3.3 how we embedded it into our previous approach to achieve a much more accurate hand tracking, which is quantified in the evaluation in Section 4. We end with a conclusion in Section 5.

2. Related Work

In mobile applications with a head-mounted camera, the background is not static. Hand localization can therefore not be achieved by simple frame subtraction. The lighting conditions may change all the time and direct sunlight can be illuminating the scene. 3D sensors are widely and successfully used for hand tracking and hand pose estimation [1], [2] because depth information easily results in an accurate segmentation of the scene. Nowadays active depth sensors are not heavy any more but are restricted to indoor applications. A light-weight passive stereo camera system would be another option to acquire depth information, but robust stereo reconstruction heavily increases the computational load. To sum up, every type of depth sensor adds additional needs to the size and weight of a HMD as well as complexity concerning the electrical wiring and increases power consumption. Accordingly, it is achievable to use only one single RGB camera. Gloves [9], markers [10], accelerators [11] or thermal cameras [12] have been used. But the optimal solution would be to not need to attach further devices to the hands of the user. Pisharady et al. [6] proposed a method for hand posture detection from single view even in front of a complex background but their method is not applicable for real-time processing. Hammer and Beyerer [7] compared different hand tracking methods with already established approaches [13][14]. In [8] we presented a 2D hand tracking approach called Motion Segmentation and Appearance Change Detection based Skin color detection (MACS) where we embedded the pixelwise skin-color classification approach of Li et al. [5] to better handle different environments with different lighting conditions. Since our new approach uses MACS as basis we will summarize the most important steps of MACS in a part of the next section. For a detailed explanation of MACS we refer to [8]. The main contribution of this work is the training of a hand detector based on Aggregated Channel Features for object detection presented by Dollar et al.[25] and the fusion process of its hand position candidates with the hand position computed by MACS, which decreases the tracking deviation by about 50% on our benchmark also presented in [8].

3. Hand Tracking Using Motion, Color and Aggregated Channel Features

The new hand tracking approach called AfM (ACF feat. MACS) relies on MACS and refines its hand position guess with the candidates computed by the hand detector using Aggregated Channel Features (ACF). In the following sections we briefly summarize MACS, describe the hand detector and the tested developed strategies.

3.1 Brief Summary of MACS

For a detailed description of MACS we refer to [8]. Its workflow can be summarized as follows: First the optical flow is estimated (Figures 1a, 1b, 1c) and clustered. From the determined clusters the foreground region assumed to contain the hand is extracted (view Figures 1d, 1e) using k-means and knowledge about the size of the hand in the images of a head-worn camera. To more robustly track this motion segment, we observe its appearance computed from the masked image (Figure 1f). This technique is essential for working with corrupted motion information especially at image boarders and when no optical flow is estimated or could not be clustered. We named this *Appearance Change Detection*. The skin color classification approach of Li et al.[5] is a further indicator for the hand segment. MACS fuses the resulting skin color segmentation (Figure 1g) and the moving foreground segmentation yielding a fused segmentation (Figure 1h) by only keeping those blobs of the skin color segmentation that overlap with areas of foreground motion. This fused segmentation is the basis for the tracking with the *shape*-particle filter (Figure 1i) that was introduced in [7]. The hand position is determined as the median position of all particles. If the average amount of skin-colored pixels per particle in a square local neighborhood of 50×50 pixels centered at the particle's position is above a certain threshold (750 pixels in our case), the hand is assumed to be in the image. Then enough skin colored pixels are in place of the particles. Otherwise the hand is assumed as not visible. This threshold is one of MACS limitations since a wrong segmentation can easily produce false positives.

3.2 Hand Classification Using Aggregated Channel Features

In order to facilitate hand detection the Aggregated Channel Feature (ACF) algorithm [25] is employed, which exploits rich color feature descriptors in conjunction with an effective classifier, following the classical approach of the Histograms of Oriented Gradients (HOG) [26].

Specifically, rather than solely relying on the gray-scale information, ACF leverages the LUV color channels combined with the gradient magnitude and 6 HOGs, leading to totally 10 feature channels. Pre- and post-smoothing with Gaussian filtering followed by 4×4 pooling are conducted to aggregate the pixels for compact and efficient representation.

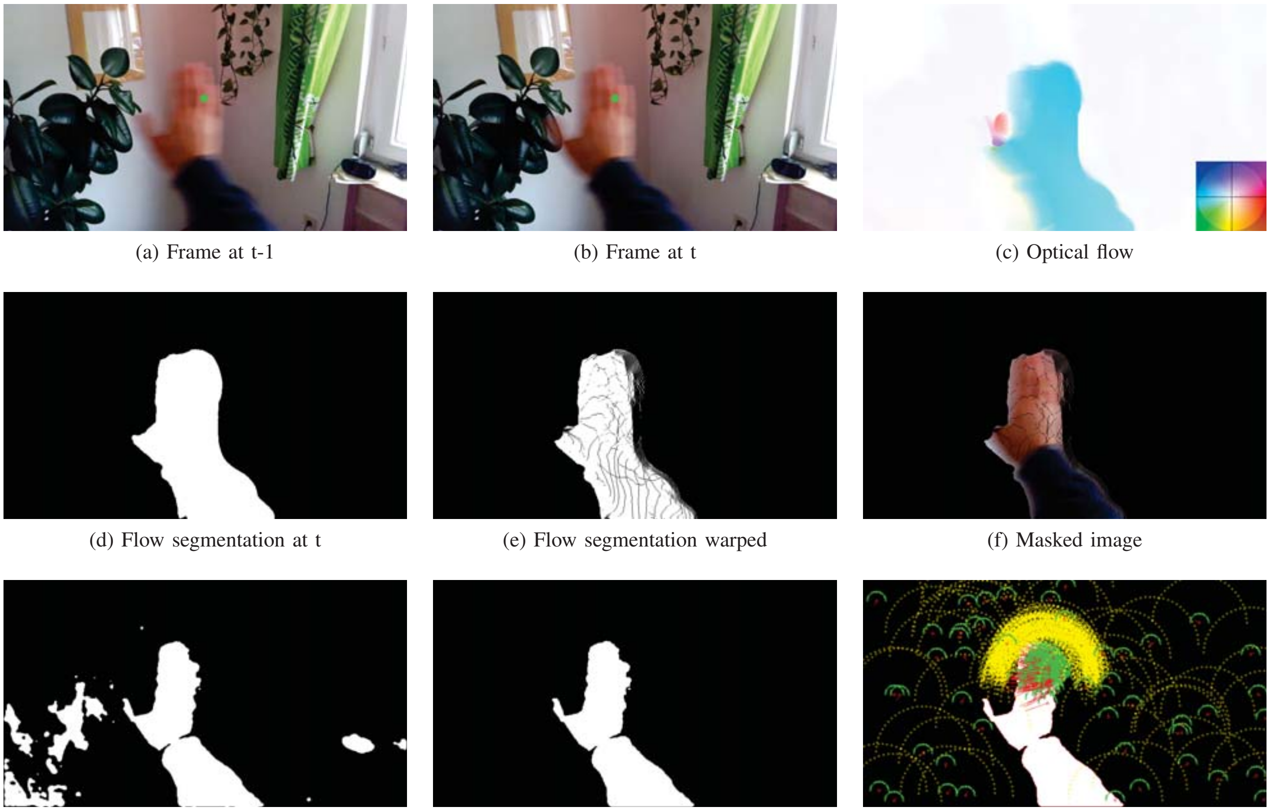
To circumvent the computational complexity of recomputing dozens of different feature pyramids, ACF only performs few of them with exact values and approximates both the color and HOG channels for the remaining scale spaces by simple rescaling of the known ones, which is justified by the power law of the natural image statistics.

Training is carried out on hand crops of 50×50 pixels extracted from a subset of frames of our sequences. Since the problem setup resembles that of the original pedestrian detection task with the palm like the upper body and the non-rigid fingers analog to the legs, we apply the same 2048 depth-two boost trees for learning with bootstrapping.

Example output of the hand detector can be seen in Figure 2. The hand detector produces results on the hands and fingers but unfortunately also on the arm and the background. Simply using the candidate with the highest score is therefore no option. But we can use MACS as guess for the hand location and refine it by the ACF candidates as described in the next section.

3.3 Fusion of MACS and the ACF-Based Hand Detector

To fuse MACS and the ACF-based hand detector we need to look at what both produce: On the one side there is the hand position guess $h_{MACS} \in \mathbb{R}^2$ produced by MACS. Furthermore MACS can decide that no hand was found. On the other side there are many hand candidates i centered at pixel p_i computed by the ACF-based hand detector where each guess comes with an associated score s_i. And as we have seen there are often guesses computed even if no hand is visible. Fortunately those guesses come with low scores.


(a) Frame at t-1

(b) Frame at t

(c) Optical flow

(d) Flow segmentation at t

(e) Flow segmentation warped

(f) Masked image

(g) Color segmentation

(h) Fused segmentation

(i) Tracking with *shape*-particle filter

Fig. 1
MACS OVERVIEW

Below we are describing different strategies for the fusion process.

3.3.1 Simple MACS Refinement

This method is the most simple approach for using the hand detector guesses. The MACS guess h_{MACS} is refined by a hand detector guess i_{best} that is close to h_{MACS} and at the same time has a high score. This i_{best} is computed as

$$i_{best} = \arg\max_i c_i \tag{1}$$

with $i \in \{1, \ldots, n\}$ based on a confidence score c_i with

$$c_i = d_i + v_i \tag{2}$$

consisting of a distance weight d_i

$$d_i = 1 - \frac{||p_i - h_{MACS}||_2}{\sum_j ||p_j - h_{MACS}||_2} \tag{3}$$

with $j \in \{1, \ldots, n\}$ and a score weight v_i

$$v_i = \frac{s_i}{\sum_j s_j}. \tag{4}$$

The final hand position h_t at the current point of time t is then set to be $p_{i_{best}}$. In the presented work we considered at the maximum the ten best hand detector guesses per frame ($n = 10$). If no guess is computed, the hand is assumed to have left the image.

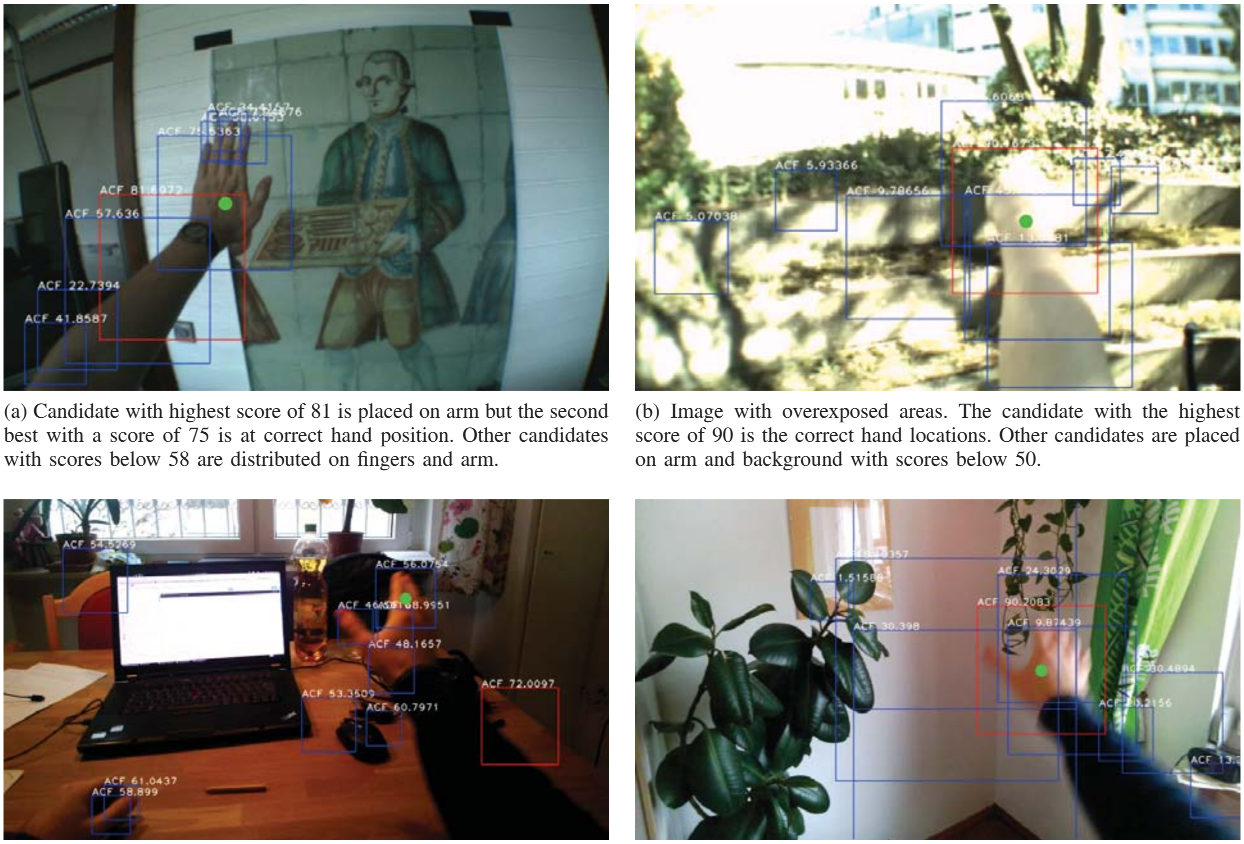
3.3.2 Refinement with Propagation

Since the MACS guess h_{MACS} itself is often not accurate with a median distance to the ground truth path of 34 pixels [8] the distance weight (3) is weighting wrong hand detector guesses p_i better because they are closer to the MACS guess. To solve this problem we use a simple propagation step that takes the last two hand positions h_{t-1} and h_{t-2} into account. The distance weight (3) is then adjusted as

$$d_i = 1 - \frac{||p_i - h_g||_2}{\sum_j ||p_j - h_g||_2} \tag{5}$$

with

$$h_g = \begin{cases} h_{t-1} + (h_{t-1} - h_{t-2}) & \text{if } h_{t-1} \text{ and } h_{t-2} \text{ available} \\ h_{\text{MACS}} & \text{otherwise} \end{cases} \tag{6}$$



(a) Candidate with highest score of 81 is placed on arm but the second best with a score of 75 is at correct hand position. Other candidates with scores below 58 are distributed on fingers and arm.

(b) Image with overexposed areas. The candidate with the highest score of 90 is the correct hand locations. Other candidates are placed on arm and background with scores below 50.

(c) Correct hand position not under the ten best candidates. But at least some candidates are on the hand and their scores are above 45. The best candidate at the right table margin has a score of 72. The candidates close to the computer mouse have scores between 53 and 60.

(d) Hands are even detected at the wall but the correct hand location has the best score with 90 compared to 30 of the second best candidate.

Fig. 2

RESULTS OF THE HAND DETECTOR. AT MOST TEN CLASSIFICATION RESULTS ARE SHOWN. THE ONE WITH THE RED COLOR HAS THE HIGHEST SCORE.

and the rest of the computation equals the computation described in 3.3.1.

3.3.3 Robust Distance Weighting

The problem with the above described distance weight is, that due to the normalization the value d_i depends on the number of hand detector guesses p_i taken into account and their distances. Therefore we change the distance rating computation to be related to a maximally accepted distance $d_{\max}$. The adapted computation of d_i is as follows:

$$d_i = \begin{cases} 0 & \text{if } ||p_i - h_g||_2 \geq d_{\max} \\ 1 - \frac{||p_i - h_g||_2}{d_{\max}} & \text{otherwise.} \end{cases} \quad (7)$$

Accordingly, if the distances of a hand detector guess to the position h_g is greater or equal $d_{\max}$ the guess is penalized with a distance weight of 0. If it is 0, it gets the highest distance weight of 1 and if it is in between, the weight is linearly lowered. This makes the distance weight independent of the number and local distribution of the hand detector guesses. Our current implementation uses a $d_{\max}$ of 100 pixels, which is a rough guess of the maximal displacement of the hand position between two consecutive frames.

3.3.4 Median Rejection of Hand Detector Guesses

To add more robustness against hand detection guesses that are too far away from the current track, we can do even more than the robust distance weighting described previously. Additionally we compute the median of the last m hand positions and state that the next hand position must not have a higher distance to this median position than $d_{\max}$. Therefore, all hand detector guesses p_i with a higher distance to the median position are discarded. We consider $m = 3$ in our implementation to not be thrown back from the current state too much.

Table 1
RESULTS FOR THE COMPLETE BENCHMARK

	TPR	**FPR**	**PREC**	**F1**	**ACC**	**qu25**	**qu50**	**qu75**
MACS	0.907	0.015	0.980	0.942	0.950	19.7 px	34.1 px	52.4 px
AfM REF	0.908	0.007	0.990	0.947	0.955	8.1 px	13.6 px	22.8 px
AfM PROP	0.904	0.007	0.990	0.945	0.953	8.1 px	13.6 px	22.5 px
AfM PROP ROB	0.905	0.007	0.990	0.946	0.954	8.5 px	14.1 px	24.7 px
AfM ALL	0.911	0.005	0.993	0.950	0.958	8.1 px	13.5 px	22.5 px

3.3.5 Adaption of the Segmentation Mask

Since the particle filter used for producing the MACS hand position guess p_{MACS} gets a binary segmentation, we can simply adjust this binary segmentation by setting all pixels to background whose distance to the median position introduced in 3.3.4 is greater than some distance. Due to the shape particle we experienced good results with such a distance of the 1.5-fold of $d_{\max}$, so in our case 150 pixels.

3.3.6 Rejecting Candidates based on the ACF Score

One of MACS limitations is that it is not tracking hands but a moving skin-colored foreground object. The decision if a hand is found is simply made upon the average number of skin pixels $\bar{s}_{\text{skin}}$ in the local 50×50 pixels neighborhood of each particle. If $\bar{s}_{\text{skin}}$ is above a certain threshold MACS assumes to have found a hand because this is the case when most of the particles condense on the hand blob. Unfortunately, when the segmentation is not good, this decision rule is not robust and highly adapted to the used data sets of [8]. We can improve this by instead using the ACF scores of the hand detector results. So, instead of using the threshold for the decision making if a hand is visible in the image or not, we set it to a very low value of $\bar{s}_{\text{skin}} = 100$ to let MACS find even more often a hand position guess. This produces more false positives but we can reject all hand detector guesses with lower scores than a threshold $t_{\text{ACF}} = 20$ to detect if a hand is visible or not.

4. Evaluation

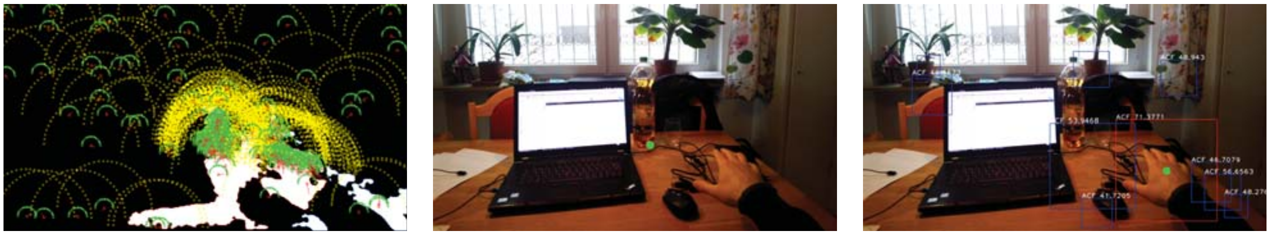
In [8] we presented our new benchmark consisting of 29 videos with more than 25,000 frames, different and challenging lighting conditions and wooden elements that make skin color classification difficult. We described our evaluation methodology and a good tracking result: The true positive rate (TPR) should be at least above 80 %. The false positive rate (FPR) should be as low as possible. The F1-measure (F1), the precision (PREC) and the accuracy (ACC) should be as high as possible. Additionally, the distances between all true positives and their corresponding ground truth hand positions should be as low as possible. But due to the subjective trajectory annotation [8] a deviation from the ground truth of less or about seven pixels per frame can be seen as perfect result. Even tracks with a deviation of up to 18 pixels per frame would subjectively be considered as very good tracking. To compare the accuracy of the tracking we are going to look at the 0.25 (qu25), 0.50 (qu50) and 0.75 (qu75) quantile of the distances.

In the evaluation we take into account from [8] only MACS since all other algorithms it was compared to failed to produce good tracking results due to their static skin color models. Then we compare MACS to the different variants of the new AfM-algorithm:

- MACS: Motion segmentation and Appearance Change Detection based skin color detection ([8])
- AfM REF: Simple MACS Refinement (Section 3.3.1)
- AfM PROP: Refinement with propagation (Section 3.3.2)
- AfM PROP+ROB: Robust Distance Weighting (Section 3.3.3)
- AfM ALL: Refinement with propagation (Section 3.3.2), robust distance weighting (3.3.3), median rejector (Section 3.3.4), mask adaption (3.3.5), ACF score rejection (Sections 3.3.6)

The detection rates, including the true positive rate, the false positive rate, the precision, the F1-measure, the accuracy, and additionally the statistics for the deviation from the ground truth path are computed in a manner as if all sequences would have been concatenated to one long sequence. The results are illustrated in Table 1.

MACS reaches a good true positive rate of 90.7 %. It shows a low false positive rate of 1.5 % and a high precision of 98 %. The F1-measure of 94.2 % is also high and the accuracy of 95.0 % is high as well. When looking at the AfM REF results, we see a better true positive rate of 90.8 %, which results in seven correctly detected hands more. The false positive rate of 0.7 % states that about 100 false positives less as MACS were produced. Accordingly the precision and the F1-measure are slightly better. The results of AfM PROP and AfM PROP ROB show that the refinement with propagation and the robust distance weighting on their own do not improve the results. Variant AfM ALL, using additionally the median rejection of hand detector guesses, the adaption of the segmentation mask and the rejection of ACF candidates based on their score, produces about 40 true positives more and 30 false positives less than AfM REF, resulting in an even better true positive rate of 91.1 %, a



(a) Condensation of the particles

(b) Input image with false hand position estimation by MACS

(c) Input image with correct hand position estimation by AfM ALL

Fig. 3

MACS FAILS BUT AFM TRACKS CORRECTLY

better false positive rate of 0.5 %, a precision of 99.3 % and an F1-measure of 95.0 %.

When we look at the distance values, we see the real difference between MACS and AfM All. The deviations from the ground truth of the tracks computed by MACS show a 0.25 quantile of 19.7 pixels, a 0.50 quantile of 34.1 pixels and a 0.75 quantile of 52.4 pixels. AfM ALL shows more than 50 % smaller deviations with a 0.25 quantile of 8.1 pixels, a 0.50 quantile of 13.5 pixels and a 0.75 quantile of 22.5 pixels. This tells us that AfM ALL produces a much more accurate tracking. Regarding that 18 pixels deviation are subjectively seen as accurate tracking [8], we can say that with AfM ALL we have come much closer to the goal of accurate tracking on our videos.

Comparing the variants of AfM the differences are marginal when looking at the complete benchmark. But the results also show that AfM ALL produces even better results without the need for a highly adapted threshold for the average number of skin pixels used by the particle filter for deciding if a hand is present or not. This decision has been delayed to the rejection of ACF candidates based on their score (view Section 3.3.6), and thus makes AfM ALL more general than the other AfM variants.

To show how AfM improves tracking accuracy we look at the example where MACS suffered from poor distinction of skin and wood as shown in Figure 3. The particles move to the upper left part of the segmentation result and the estimated hand position is distracted as visualized by the green dot in Figure 3b. This problem is solved by AfM because of its hand detector as can be seen in Figure 3c where the green dot is placed on the hand.

5. Conclusion

To sum up, we have shown a new 2D hand tracking algorithm called AfM that improves our previous method MACS by the usage of hand detector based on Aggregated Channel Features. We proposed several approaches for the fusion process of the MACS hand position guess and the candidates of the hand detector. AfM is the first approach to produce good tracking results on our benchmark consisting of more than 25,000 frames under challenging conditions, quantified by the usual detection values and deviations from the ground truth paths. AfM reaches better detection values than MACS on its own and reduces the deviations from the ground truth path by more than 50 %. Furthermore the AfM ALL variant moves the decision if a hand is visible from the particle filter of MACS to the fusion process by rejecting candidates of the hand detector by their score, through which AfM ALL yields much more generality.

Future work will concentrate on a better and more automated scene segmentation that does not need any assumptions about the number of motion layers or the size of the hand as still needed in MACS to make AfM more general. One should even think about an online training phase of the hand detector. Since AfM comes close to good tracking, the benchmark needs to be improved with further sequences containing other moving objects and more people performing the gestures in different scenes.

References

[1] Oikonomidis, I., Lourakis, M.I.A., Argyros, A.A.: Evolutionary quasi-random search for hand articulations tracking. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3422–3429. CVPR '14, IEEE Computer Society, Washington, DC, USA (2014)

[2] Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and Robust Hand Tracking from Depth. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1106–1113 (Jun 2014)

[3] de La Gorce, M., Fleet, D., Paragios, N.: Model-Based 3d Hand Pose Estimation from Monocular Video. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(9), 1793–1805 (Sep 2011)

[4] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision 92, 1–31 (2011)

[5] Li, C., Kitani, K.M.: Pixel-level hand detection in ego-centric videos. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 3570–3577. IEEE (2013)

[6] Pisharady, P., Vadakkepat, P., Loh, A.: Attention based detection and recognition of hand postures against complex backgrounds. International Journal of Computer Vision 101, 403–419 (2013)

[7] Hammer, J. H., Beyerer, J.: Robust hand tracking in realtime using a single head-mounted rgb camera. In: Kurosu, M. (ed.) Human-Computer Interaction. Interaction Modalities and Techniques, Lecture

Notes in Computer Science, vol. 8007, pp. 252–261. Springer Berlin Heidelberg (2013)

[8] Hammer, J. H., Voit, M., Beyerer, J.: Motion Segmentation and Appearance Change Detection Based 2D Hand Tracking. In: To appear at the 19th International Conference on Information Fusion (July 2016).

[9] Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ACM Trans. Graph. 28(3), 63:1–63:8 (Jul 2009)

[10] Mistry, P., Maes, P.: Sixthsense: a wearable gestural interface. In: ACM SIGGRAPH ASIA 2009 Sketches. pp. 11:1–11:1. SIGGRAPH ASIA '09, ACM, New York, NY, USA (2009)

[11] Prisacariu, V., Reid, I.: Robust 3d hand tracking for human computer interaction. In: Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on. pp. 368 –375 (march 2011)

[12] Appenrodt, J., Al-Hamadi, A., Elmezain, M., Michaelis, B.: Data gathering for gesture recognition systems based on mono color-, stereo color- and thermal cameras. In: Proceedings of the 1st International Conference on Future Generation Information Technology. pp. 78–86. FGIT '09, Springer-Verlag, Berlin, Heidelberg (2009)

[13] Bradski, G.R.: Real time face and object tracking as a component of a perceptual user interface. In: Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98). pp. 214–. WACV '98, IEEE Computer Society, Washington, DC, USA (1998)

[14] Kölsch, M., Turk, M.: Fast 2d hand tracking with flocks of features and multi-cue integration. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on. p. 158 (june 2004)

[15] Phung, S., Bouzerdoum, A., S., Chai, D., S.: Skin segmentation using color pixel classification: analysis and comparison. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(1), 148 –154 (jan 2005)

[16] Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognition 40(3), 1106 – 1122 (2007)

[17] Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l1 optical flow. In: Proceedings of the 29th DAGM conference on Pattern recognition. pp. 214–223. Springer-Verlag, Berlin, Heidelberg (2007)

[18] Needham, C.J., Boyle, R.D.: Performance evaluation metrics and statistics for positional tracker evaluation. In: Proceedings of the 3rd international conference on Computer vision systems. pp. 278–289. ICVS'03, Springer-Verlag, Berlin, Heidelberg (2003)

[19] DHL Customer Solutions & Innovation. Augmented Reality in Logistics. Changing the way we see logistics - a DHL perspective. http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/csi_augmented_reality_report_290414.pdf, 2014, Accessed: 2016-03-14.

[20] Nils Petersen and Didier Stricker. Morphing billboards: an image-based appearance model for hand tracking. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pages 1–13, January 2014.

[21] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3633–3642, New York, NY, USA, 2015. ACM.

[22] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2015.

[23] Deqing Sun, E.B. Sudderth, and M.J. Black. Layered segmentation and optical flow estimation over time. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1768–1775, 2012.

[24] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal Video Object Segmentation From Persistence of Occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015.

[25] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, August 2014.

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

Multidimensional Affective Model-based Multimodal Complex Emotion Recognition System using Image, Voice and Brainwave

Byung-Hun Oh[1], **and Kwang-Seok Hong**[1]
[1]School of Information and Communication engineering , Sungkyunkwan University, Suwon,
16419, Korea

Abstract - *This paper proposed a multimodal complex emotion recognition system using image, voice, and brain-wave based on multidimensional affective model. Based on face image, voice, and brain wave of users, features corresponding to the explicit response degree in multidimensional affective model known as emotional response elements were extracted and scored to construct human emotions in the field of psychology and cognitive science. By mapping three-dimensional emotion model consisting of the multidimensional affective model using results from our scoring method, human emotions and the intensity of emotions are recognized.*

Keywords: Emotion Recognition, Affective Model, Scoring Method, AVD Model

1 Introduction

In the field of human interface technology development, the interactions between human and machine are important. Researches on emotion recognition will help us understand these interactions. Emotion recognition is one of the most important factors in human-centric human-to-machine interface [1].

Emotion is a subjective and conscious experience primarily characterized by psychophysiological expressions, biological reactions, and mental states. Emotion is often associated with mood, temperament, personality, disposition, and motivation that can reciprocally influence emotion [2]. Emotion information is transmitted through a wide range of modalities, including facial expression, voice, electroencephalogram, posture, head movement, and body movement. Even though the expression of simultaneity of multi modalities related to emotion is important for the recognition of emotions, most studies have performed emotion recognition through single modality such as facial expression, voice, or body movement [3].

In most state of the art technologies on emotion recognition, features extracted from image, speech, and bio signals are classified into specific emotional categories based on pre-trained recognition models. However, the exact state of emotion expressed as a contiguous vectors in multi-dimensional is quite contrived. Therefore, the objective of this study was to extract features based on user's face image, voice, and brain waves to determine if any component of emotion based on the extracted features is contained to some extent by extracting sensitivity values of Arousal, Valence, and Dominance (AVD). By mapping the multidimensional model through scoring the extracted sensitivity value, user's emotions are recognized.

2 System Architecture

This section describes a multimodal complex emotion recognition system based on the multidimensional affective model. A simple block diagram for the proposed multimodal complex emotion recognition system is shown in Figure 1.

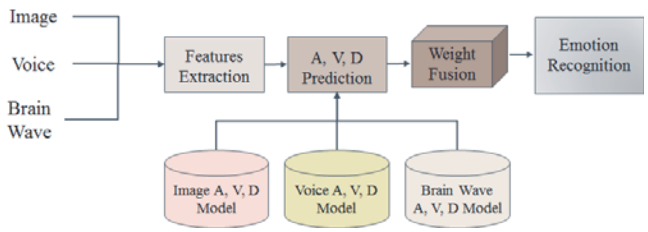


Figure 1. Flowchart of the proposed multimodal complex emotion recognition system

First, user's face image, voice, and brain-wave features influencing the emotion are extracted. The multimodal complex emotion recognition system is composed of facial, voice, brain-wave feature extraction, affective model (Arousal, Valence, Dominance) prediction, emotion model mapping, and emotion recognition phase in sequential steps.

3 The Proposed Emotion Recognition System

3.1 Multidimensional Affective Model

Dimensional models of emotion are used with attempt to conceptualize human emotions by defining where they lie in two or three dimensions. Most dimensional models incorporate valence and arousal or intensity dimensions. Dimensional models of emotion suggest that a common and interconnected neurophysiological system is responsible for all affective states.

The AVD emotional state model [4] is a psychological model to describe and measure emotional states. AVD uses three numerical dimensions to represent all emotions. AVD dimensions are shown in Figure 2. Valence scale measures how pleasant an emotion might be. For instance, both anger and fear are unpleasant emotions. Therefore, they have high unpleasant scores on the Valence scale. {Editor's Note: Please double check the highlighted sentence and make sure it is acceptable. The original sentence was unclear.} However, joy is a pleasant emotion [5]. Arousal scale measures the intensity of the emotion. For instance, anger and rage are both unpleasant emotions, but rage has a higher intensity or higher arousal state compared to anger. Boredom is also an unpleasant emotion, but it has a lower arousal value compared to anger. Dominance scale represents the controlling or dominant nature of the emotion. For instance, while both fear and anger are unpleasant emotions, anger is a dominant emotion while fear is a submissive emotion.

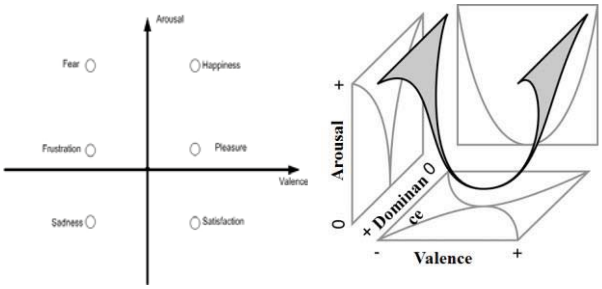


Figure 2. Arousal, Valence, and Dominance (AVD) dimensions in the multidimensional Affective Model

3.2 Extraction of Voice Features

To recognize emotion, we selected many features due to their demonstrated superior performance [6]. A total of 55 detailed voice features of pitch, pitch derivative, energy, speech rate, and Mel-frequency cepstral coefficients (Table 1) were extracted in this study.

Table 1. Extracted voice features

Feature	Detailed Feature	Feature number
Pitch	Average, Maximum, Minimum, Standard deviation, variance	5
Pitch derivative	Average, Maximum, Minimum, Standard deviation, variance	5
Energy	Average, Maximum, Minimum, Standard deviation, variance	5
Speech rate	Rate of speech	1
MFCC	13thMel-frequencycepstralcoefficients	13

3.3 Extraction of Facial Features

For face-based emotion recognition, each video frame contained only the person of interest with a frontal view of the face (the SKKU emotion database provides video data in this form). We first detected the face. After that, background information was discarded (i.e., a box was placed around the face). Different-sized boxed faces were obtained due to random perturbations in camera zoom. To compensate for this effect, facial images were re-scaled into a standard size suitable for feature extraction. To accomplish this, bilinear interpolation was employed, resulting in standardized facial images with the same number of pixels in each image. For correlation analysis, the distance to main features reflecting emotions were determined. A total of 11 facial features (Table 2) were extracted.

Table 2. Extracted facial features

Example Image	Description	Feature number
	1. Inner brow raising	2
	2. Outer brow raising	2
	3. Brow lowering	1
	4. Lip corner pull	2
	5. Upper lip raising	1
	6. Mouth stretching	1
	7. Lower lip depression	1
	8. Nose Wrinkling	1

3.4 Extraction of Brain-wave Features

The lower regions behind the ears were used for EEG measurements. A total of 14 measurement points denoted as F7, AF3, FC5, F3, T7, P7, O1, F8, AF4, FC6, F4, T8, P8 and O2 were used. They were placed according to the international 10-20 electrode system. Gyro X-axis and Y-axis were assigned to channels 8 and 16, respectively. We used features from the measured raw data. The 14 points used in EEG measurement according to the international 10-20 electrode system are shown in Figure 3.

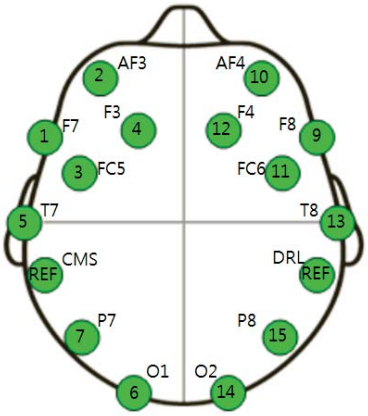


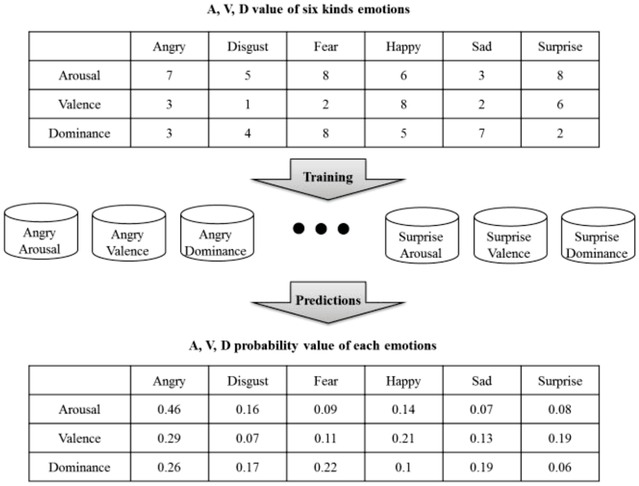
Figure 3. The 14 points in the international 10-20 electrode system

3.5 Sensitive Value Scoring

In this paper, we predicted emotions by applying the AVD model. First, user's face, voice, and brain-wave were captured by camera, microphone, and brain-wave measurement devices, respectively. The features of facial, voice, and brain-wave were then extracted. The extracted features were inputted to the emotional prediction module. The level of emotional response predicted was then compared to the emotional response generated by the model using Random Forest algorithm. The similarity score (from 1 to 9 points) between emotional response values predicted (as shown in the predicted results table in Fig. 4) and the emotional response obtained by the AVD model was obtained. The higher the score is, the more similarity between the prediction and the AVD model exists.

3.6 Emotion Mapping

In this paper, we predicted the emotion by applying the AVD model. First, user's face and voice were inputted to the camera and microphone. Features of facial and voice were then extracted. The extracted features were inputted to the emotional prediction module. The predicted level of emotional response was then compared to the level previously generated by the emotional response model using Random Forest algorithm. If the emotional response value predicted was similar to that shown in the predicted result table in Fig. 4, AVD similarity scores (from 1 to 9 points) were obtained. The higher the score, the more similarity between the prediction and the AVD model. An example of the scoring system based on AVD emotional model is shown in Figure 4.



A, V, D value of six kinds emotions

	Angry	Disgust	Fear	Happy	Sad	Surprise
Arousal	7	5	8	6	3	8
Valence	3	1	2	8	2	6
Dominance	3	4	8	5	7	2

A, V, D probability value of each emotions

	Angry	Disgust	Fear	Happy	Sad	Surprise
Arousal	0.46	0.16	0.09	0.14	0.07	0.08
Valence	0.29	0.07	0.11	0.21	0.13	0.19
Dominance	0.26	0.17	0.22	0.1	0.19	0.06

Figure 4. Example of scoring based on AVD emotional model

Using the result of Arousal and Valence derived from the emotional response scoring module, various feelings were mapped to a two dimensional plane. After performing emotion mapping in the two dimensional plane, we can see the intensity of the emotion using the result of Dominance derived from the emotional response scoring module. An example of complex emotion recognition based on the AVD emotional model is shown in Figure 5.

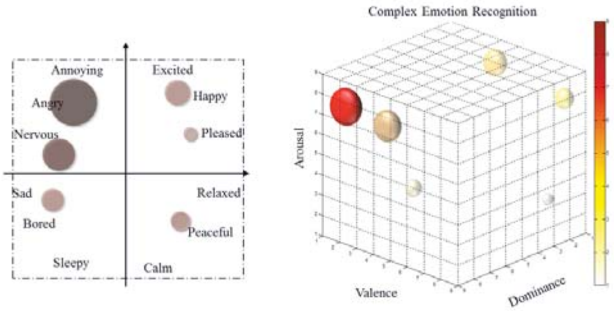


Figure 5. Example of complex emotion recognition based on AVD emotional model

4 Experimental Results

To evaluate the proposed algorithm, a self-production emotion DB at natural state was used in the laboratory. As shown in Table 3, each unimodal of facial, voice, and brain-wave, and a combination of the three modalities was used in the experiment. Multimodal experiments were performed with various weight values. The highest recognition rate is shown in Table 3. In recognition rate, when combined with multimodal of facial, voice, and brain-wave, the proposed algorithm had the best performance (recognition rate of 65.98%).

Table 3. Emotion recognition experimental results

Modalities	Ranking 1	Ranking 2	Ranking 3
Facial	58.07%	69.77%	80.91%
Voice	61.11%	71.11%	82.03%
Brain-wave	51.75%	61.14%	70.19%
Facial + Voice	64.06%	73.93%	85.10%
Facial + Brain-wave	58.21%	71.16%	81.13%
Voice + Brain-wave	61.28%	70.89%	82.03%
Facial + Voice + Brain-wave	65.98%	75.21%	85.73%

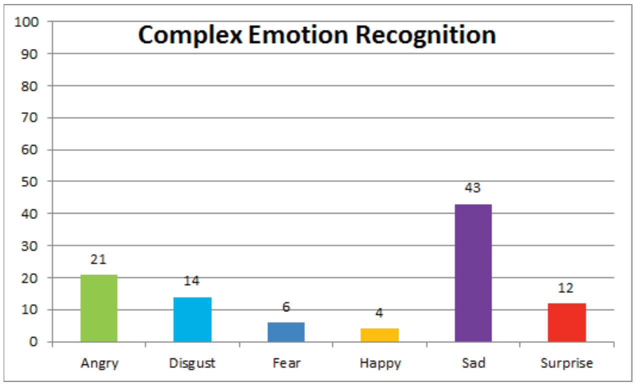


Figure 6. Example of displaying the recognized complex emotion

In addition, by ranking the six emotions from one to six, the recognized emotions could be expressed in a graph. As shown in Figure 6, the values of multiple emotions that user feel could be seen.

5 Conclusion

In this paper, we proposed a multimodal complex emotion recognition system using image, voice, and brain-wave based on multidimensional affective model. This proposed system could detect complex emotions through multi-modalities of facial image, voice, and brain waves. By applying multi-dimensional emotion model to extract features from each of the multi-modalities, the intensity of emotions can be measured.

6 Acknowledgments

This research was supported by Basic Science Research Program through NRF of Korea, funded by MOE(NRF-2010-0020210) and This research was supported by the MSIP, Korea, under the G-ITRC support program (IITP-2015-R6812-15-0001) supervised by the IITP.

7 Reference

[1] K.R. Scherer, "Emotion, Introduction to Social Psychology: A European perspective," Oxford: Blackwell, pp.151-191, 2000.

[2] J. A. Harrigen, "The new handbook of methods in nonverbal behavior research," Series in Affective Science, Oxford University Press, pp. 137-198, 2005.

[3] Ayadi. El. Moataz, S. Kamel. Mohamed, Karray. Fakhri, "Survey on speech emotion recognition: Features, classification schemes, and databases," Elsevier, Patter Recognition, Vol. 44, No. 3, pp. 572-587, Mar, 2011.

[4] Ishihara. Hidenori, Toshio. Fukuda, "Individuality of agent with emotional algorithm," Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and System, pp. 1195–1200, 2001.

[5] Albert. Mehrabian, "Framework for a comprehensive description and measurement of emotional states," Genet. Soc. Gen. Psychol. Monogr, pp. 339–361, 1995

[6] Aleksic, P. S. and Katsaggelos, A. K., "Audio-visual biometrics," Proc. IEEE, vol. 94, no. 11, pp. 2025-2044, Nov. 2006.

A Two-Phase Fuzzy System for Edge Detection

Azzam Sleit*, Maha Saadeh, Wesam AlMobaideen
Department of Computer Science, King Abdulla II School for Information Technology, P.O. Box 13898
University of Jordan, Amman 11942, Jordan
azzam.sleit@ju.edu.jo, maha.k.saadeh@gmail.com, almobaideen@inf.ju.edu.jo
*Corresponding Author

Abstract: Edge detection is a preliminary process in many image processing and computer vision applications. It detects important events in the image where sharp discontinuity in pixels' intensity is found. Several edge detection techniques have been proposed including Sobel, Canny, Prewitt, and many others. Since fuzzy logic is a powerful tool to manage the uncertainty efficiently, it can be used in edge detection to decide whether a certain pixel is an edge pixel or not. In this paper, a two-phase fuzzy inference system is proposed to detect edges in gray level images. In the first phase the discontinuity in pixels' intensity is evaluated according to various directions, while in the second phase the final decision is determined based on the results obtained from the first phase. The proposed algorithm is implemented using MATLAB and experimental results show improvement when compared with other edge detection techniques.

Keywords: *Edge Detection; Fuzzy System; Sobel; Canny; Gradient.*

I. INTRODUCTION

Digital image processing is the process of analyzing image pixels in order to extract the needed information such as image texture, objects features, color information, and image edges [1- 3]. Image edge detection refers to the process of extracting the sharp discontinuity in pixels intensity in a certain image.

The importance of edge detection is that it is considered as a basic step in various image operations such as boundary detection, object recognition, and object classification. Several edge detection techniques have been proposed such as Sobel, Roberts, Canny, Prewitt, and Laplacian of Gaussian [4]. The detection of image edges using these techniques is based on the calculation of image first or second derivatives. Gradient-based edge detection methods is performed by calculating image first derivative and then looking for maximum and minimum. On the other hand, Laplacian-based edge detection methods searches for zero crossing in the second derivative [4].

The fuzzy set, first proposed by Zadeh in 1965 [5], is defined as a set of elements with a degree of membership between 0 and 1. The mathematical function that describes the membership of elements in a fuzzy set is called a membership function [5]. The advantage of fuzzy logic is that it describes the problem in terms of linguistic variables which makes it a powerful tool for managing the vagueness and uncertainty efficiently [6]. Since the decision whether a pixel should be considered as an edge or not is uncertain, fuzzy logic has been successfully used in image edge detection [6 - 10].

In this paper, a new edge detection technique is proposed. It uses a 3χ3 mask and according to the relationship between mask's pixels, a two-phase fuzzy system is applied to determine whether the center pixel is an edge or not. In the first phase each pixel is tested to detect edges according to four directions; horizontal, vertical, diagonal, and inverse diagonal. The second phase aims to combine the results from the first phase in order to provide the final decision on a certain pixel. The proposed algorithm is implemented using MATLAB and the results of the new technique is compared with those of Sobel, and Canny.

The rest of the paper is organized as the following. Section II reviews related works. Then, section III presents an overview on fuzzy logic. Section IV discusses the proposed technique. Section V exhibits experimental evaluation of the proposed technique. Finally, section VI concludes the paper.

II. RELATED RESEARCH

Many techniques have been proposed for image edge detection. Sobel method [4] uses two kernels, shown in Figure 1, to obtain edge intensities in the vertical and horizontal directions. The kernels can be applied separately to the input image to produce separate measurements of the gradient components in each direction then they can be combined to find the magnitude of the gradient at each pixel. Eq. (*1*) shows how to calculate the magnitude of the gradient [4].

$$|G| = \sqrt{\left(G_x^2 + G_y^2\right)} \quad \dots (1)$$

Where G_x and G_y are the gradient in directions x, y respectively. If the pixel intensity exceeds a specific threshold, the pixel will be regarded as an edge point.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Figure 1: Sobel's Vertical and Horizontal Masks.

Canny method [4] finds the edge strength by taking the gradient of the image using the same kernels used by Sobel operator to find the gradient in both directions, i.e. x and y, and then the magnitude is calculated. Next, it computes the edge direction according to the gradient in the x and y directions using Eq. (*2*) [4].

$$\theta = tan^{-1}\left(\frac{G_x}{G_y}\right) \quad \dots (2)$$

Once edge direction is known, the next step is to resolve the orientation of the edge into one of the four directions shown in Figure 2, based on which direction is closest to θ. Finally, a Non-Maxima Suppression (NMS) method is used to extract the edge point with the largest edge intensity along its direction which gives a thinner edge line in the output image.

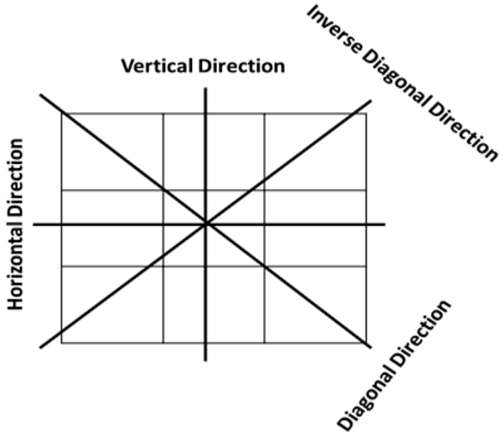


Figure 2: Edge Directions.

In [7] the authors proposed a fuzzy edge detection approach in which each pixel is compared with neighbors' intensities by applying a 3χ3 mask on the image and then all mask pixels' intensities are entered to the fuzzy system. Afterward, and according to inference rules the center pixel will be considered as black, white, or edge pixel.

Another fuzzy approach proposed in [8] which is based on the calculation of standard deviation and gradient values. For each center pixel of a 3χ3 mask, these two values are entered in to the fuzzy system and are evaluated as low, medium, or high edge intensity.

Liang et al. proposed an algorithm which operates on gray images via three passes on image pixels [9]. In the first pass, a fuzzy classifier is used to classify the pixels according to gray level variation in various directions to 6 classes; edge in direction-1, edge in direction-2, edge in direction-3, edge in direction-4, background, and speckle. The second pass applies a competitive process to compare the edge pixels with its neighbors to obtain a thinner edge. Finally, the last pass is to de-speckle the speckle pixels.

Anver et al. proposed a fuzzy system that decides on image edges based on different mask sizes [10]. They use 3χ3, 5χ5, and 7χ7 masks and for each mask they calculate line-edginess and step-edginess according to intensities differences between the center pixel and its neighbors. For example when using the 3χ3 mask, they compare the step-edginess$_{3x3}$ and the line-edginess$_{3x3}$ for each pixel in the image and the winner is called edginess$_{3x3}$. After this step, each pixel in the image will have three values; edginess$_{3x3}$, edginess$_{5x5}$, and edginess$_{7x7}$. The three values are entered in to a fuzzy system to find the final edginess strength.

Chung-chia kang and Wen-June Wang [11] proposed an edge detection technique for both gray and color images. They use a 3χ3 mask divided into two sets S_0 and S_1, which are used to compute objective functions corresponding to four directions as per Figure 3. The objective function that corresponds to edge direction j is defined by Eq. (3.a), Eq. (3.b) and Eq. (3.c).

$$f_j = (L-1) \times \frac{Nf}{Df} \qquad \ldots (3.a)$$

$$Nf = min\left(1, \frac{|m_0 - m_1|}{w_1}\right) \qquad \ldots (3.b)$$

$$Df = 1 + \frac{1}{15}\sum_{S_0} min\left(1, \frac{|p_m - p_n|}{w_2}\right) + \frac{1}{3}\sum_{S_1} min\left(1, \frac{|p_m - p_n|}{w_2}\right) \qquad \ldots (3.c)$$

Where L is the gray level of the digital image, w_1= 90, w_2= 40, m_0 and m_1 are the average of pixels' intensities in S_0 and S_1 respectively. p_m, p_n are two pixels' intensities in the same set such that m ≠ n and m > n. For each direction j, the objective function f_j is calculated and then edge intensity is calculated by Eq. (*4*). Finally, a non-maxima suppression method is applied for pixels with intensity ≥ specific threshold to extract edge points along the direction, which is used to obtain a thinner edge along the direction.

$$Edge = Max(f_1, f_2, f_3, f_4) \qquad \ldots (4)$$

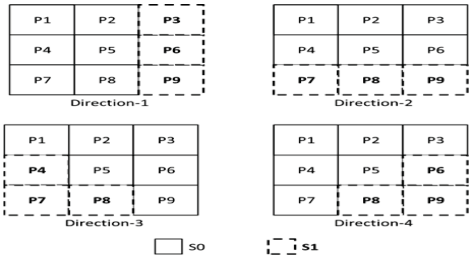


Figure 3: Directions' Sets.

Another Edge detection technique is proposed by Jiang et al. in which a mathematical model is used to extract thin edges in low-contrast images [12]. They developed a quad-decomposition edge enhancement process, a thresholding process, and a mask-based noise filtering process to enhance thin edge features, extract edge points and filter out some meaningless noise points.

Although fuzzy systems have been used in some previous techniques such as in [7, 8, 10], the proposed technique considers the relation between pixels' intensities in four different directions within the same mask. In [9] although the edges are considered in four directions, two additional steps are needed to obtain thinner edges and remove speckle pixels which are not needed in the proposed technique.

III. FUZZY LOGIC OVERVIEW

Fuzzy logic was introduced by Lotfi Zadeh in 1965 [5]. The power of fuzzy logic is that it presents set membership as a value between 0 and 1 rather than traditional crisp value which can be 0 or 1. Fuzzy logic can be used in many systems to reflect the uncertainty by define the degree to which an element belongs to a particular set [6]. The difference between fuzzy value and crisp value is illustrated in Figure 4.

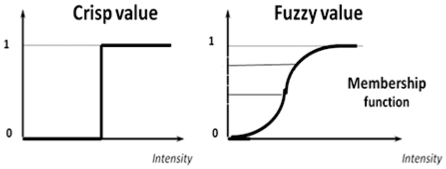


Figure 4: Difference between Crisp and Fuzzy Sets.

Fuzzy inference systems have three main components; inputs, output(s), and rules. Two main fuzzy inference styles are known; Mamdani and Sugeno**.** In this section we will discuss Mamdani-style inference [13] since it has been used for the proposed fuzzy system.

Each input and output is described as linguistic variable such as Pressure. The values of input or output variable are represented by linguistic terms. Each linguistic term is represented by one membership function; in our example the Pressure can be Low or High. Both Low and High are linguistic terms which can be presented by a membership functions such as Trapezoidal, Triangular, or Gaussian. Fuzzy rule is an If-then rule that consists of two parts; antecedent(s) and consequent(s). The general format of fuzzy rule is:

If Input-variable is Value$_n$

Then Output-Variable is Value$_m$

For example: If Pressure is High Then Volume is Small. After defining the inputs, and outputs along with the membership functions and the needed fuzzy rules, the fuzzy system is ready to be used. The evaluation process can be performed in four steps illustrated in Figure 5.

Firstly, the fuzzification step in which the crisp input value is mapped into a fuzzy value using the corresponding membership functions. Secondly, in rules evaluation step, fuzzified inputs are applied to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator AND or OR (correspond to minimum and maximum respectively) is used to obtain a single number that represents the result of the antecedent evaluation. This number, called antecedent truth, is then applied to the consequent membership function by clipping the consequent membership function at the level of the antecedent truth. Next, in rules aggregation step, the membership functions of all rules' consequents previously clipped are combined into a single fuzzy set. Finally, the output value is mapped from fuzzy value to crisp value in the defuzzification step [13].

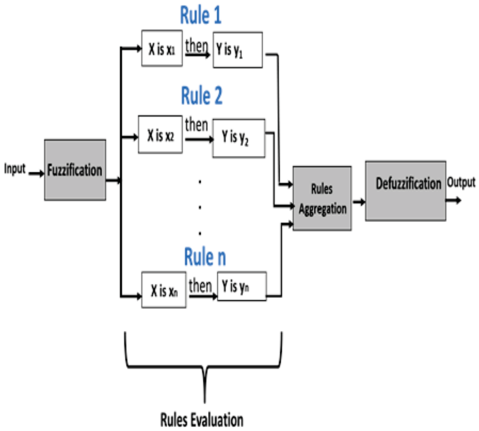


Figure 5: The Main Steps in Inference System Evaluation.

IV. PROPOSED TECHNIQUE

In this paper a new edge detection technique for grayscale images is proposed. Our technique consists mainly of two steps; the first step is the pre-processing step in which the parameters that are needed for the fuzzy system, will be calculated. These parameters represent the distances in pixels' intensity in 3χ3 mask. In the second step, the calculated distances will be evaluated by two-stage fuzzy system to decide whether the mask center pixel is an edge pixel or not. Figure 6 illustrates the general structure of the proposed fuzzy system along with the needed parameters for each stage. In the next subsections the proposed technique will be discussed in more details.

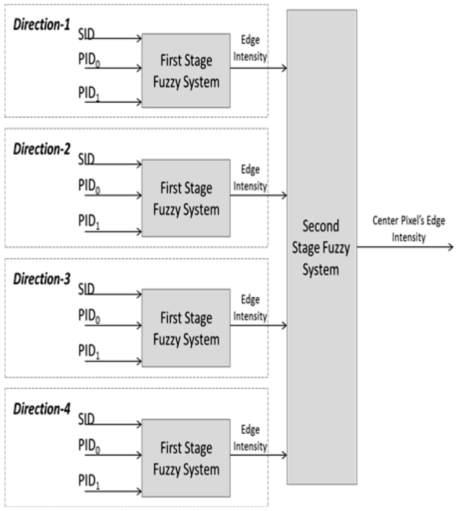


Figure 6: Proposed Fuzzy System General Structure.

A. Step 1: Parameters Calculation

In the first step, all fuzzy parameters are calculated. For each pixel in the gray image, a 3χ3 mask is applied. Then, the needed parameters are calculated depending on the pixels' intensity variation according to four directions; vertical, horizontal, diagonal, and inverse diagonal, as shown in Figure 2. For each direction two sets are defined; S_0 and S_1 which are shown in Figure 3. Each set consists of a number of pixels which varies according to a specific direction, for example, Direction-1 has S_0= {p1, p2, p4, p5, p7, p8}, and S_1= {p3, p6, p9}.

Two different intensity distances are defined to recognize the relation between the sets' pixels. The first one is the Set Intensity Distance (SID) which characterizes the inter-set intensity distance between S_0 and S_1. It is calculated using Eq. (5.a), Eq. (5.b), and Eq. (5.c).

$$SID = | \overline{s_0} - \overline{s_1} | \qquad \ldots (5.a)$$

$$\overline{s_0} = \frac{\sum_{i=1}^{6} Pi_{s_0}}{6} \qquad \ldots (5.b)$$

$$\overline{s_1} = \frac{\sum_{i=1}^{3} Pi_{s_1}}{3} \qquad \ldots (5.c)$$

$\overline{\boldsymbol{s_0}}$ and $\overline{\boldsymbol{s_1}}$ are the intensity average for pixels in S_0 and S_1 respectively. Pi_{S0} is the ith pixel's intensity in S_0 set, and Pi_{S1} is the ith pixel's intensity in S_1 set. The second distance type is the Pixels Intensity Distance (PID) which describes the pixels' intensity variation between the pixels within the same set (Intra-set intensities variance). Two PID are defined; PID_0 and PID_1 corresponding to the sets S_0 and S_1 respectively. To find PID_0 and PID_1 we calculate the Standard Deviation (SD) for pixels' intensities within each set as per Eq. (6.a) and Eq. (6.b), respectively. Lower standard deviation implies smaller intensities variance and thereby lesser possibility to have an edge.

$$PID_0 = SD_{s_0} \qquad \ldots (6.a)$$

$$PID_1 = SD_{s_1} \qquad \ldots (6.b)$$

B. Step 2: Fuzzy Evaluation

After finding intensity distances, we use a fuzzy system to evaluate these distances. As mentioned before, the proposed fuzzy system is applied to decide on edge pixels using two stages. Firstly, decide if the center pixel, in the mask, is an edge according to each direction. Secondly, combine the effect of all directions and decide whether to consider the center pixel as an edge or not.

For each direction, the first fuzzy stage is applied to find the edge intensity which reflects the possibility to have an edge in that direction. It is a three inputs one output fuzzy inference system. It takes SID, PID_0, and PID_1 as inputs and returns the edge intensity as an output. This intensity reflects the possibility to have an edge in that direction.

The SID variable has three values; *Small*, *Medium*, and *Large*. Both PID_0 and PID_1 have also three membership functions; *Small*, *Medium*, and *Large*. Each input parameter will be fuzzified according to its corresponding membership functions. The trapezoidal and triangular functions are used to represent the membership functions. These functions are represented in Eq.(7) and Eq.(8), respectively.

After fuzzification process all values will be evaluated according to the Mamdani fuzzy rules of this stage which are listed in

Table I. Finally, the output is defuzzified according to the rules evaluation result, and the value is represented by *Low*, *Medium*, and *High* membership functions.

Table I: First Stage Fuzzy Rules (the shaded area is the output).

SID	PID_{s0}	PID_{s1}		
		Small	**Medium**	**Large**
Small	**Small**	Low	Low	Medium
Medium	**Small**	Medium	Medium	Medium
Large	**Small**	High	High	High
Small	**Medium**	Low	Low	Medium
Medium	**Medium**	Medium	Medium	High
Large	**Medium**	High	High	High
Small	**Large**	Medium	Medium	Medium
Medium	**Large**	Medium	High	High
Large	**Large**	High	High	Low

After finding the edge intensities for all directions, these values are entered to the second stage which is a four to one fuzzy inference. As first stage the second stage uses Mamdani fuzzy inference to evaluate its inputs with rules listed in .

Table II.

The inputs are the four outputs from the first phase (one output for each direction) and the output is the final edge intensity according to the four directions results. Both input and output are represented as *Low*, *Medium*, and *High* membership functions.

Table II: Second Fuzzy Stage Rule (the shaded area is the output).

D1	D2	D3	D4		
			Low	**Medium**	**High**
Low	**Low**	**Low**	Low	High	High
Low	**Low**	**Medium**	High	Low	High
Low	**Low**	**High**	High	High	High
Low	**Medium**	**Low**	High	Low	High
Low	**Medium**	**Medium**	Low	High	Medium
Low	**Medium**	**High**	High	Medium	High
Low	**High**	**Low**	High	High	High
Low	**High**	**Medium**	High	Medium	High
Low	**High**	**High**	High	High	High
Medium	**Low**	**Low**	High	Low	High
Medium	**Low**	**Medium**	Low	High	Medium
Medium	**Low**	**High**	High	Medium	High
Medium	**Medium**	**Low**	Low	High	Medium
Medium	**Medium**	**Medium**	High	Low	High
Medium	**Medium**	**High**	Medium	High	Low
Medium	**High**	**Low**	High	Medium	High
Medium	**High**	**Medium**	Medium	High	Low
Medium	**High**	**High**	High	Low	High
High	**Low**	**Low**	High	High	High
High	**Low**	**Medium**	High	Medium	High
High	**Low**	**High**	High	High	High
High	**Medium**	**Low**	High	Medium	High
High	**Medium**	**Medium**	Medium	High	Low
High	**Medium**	**High**	High	Low	High
High	**High**	**Low**	High	High	High
High	**High**	**Medium**	High	Low	High
High	**High**	**High**	High	High	Low

$$\mu(x,a,b,c,d) = \begin{cases} 0 & x < a, x > d \\ \frac{x-a}{b-a} & a \le x \le b \\ 1 & b < x < c \\ \frac{d-x}{d-c} & c \le x \le d \end{cases} \quad \ldots(7)$$

$$\mu(x,a,b,c) = \begin{cases} 0 & x < a, x > c \\ \frac{x-a}{b-a} & a \le x \le b \\ \frac{c-x}{c-b} & b < x \le c \end{cases} \quad \ldots(8)$$

V. EXPERIMENTAL RESULTS

In this section, we present the results of the proposed technique and compare them with those obtained from Sobel, and Canny. We tested the proposed technique on images which can be categorized as images that contain many details (Figure 7 and Figure 8). The original images are shown in (a) part. Images (b) part show the results of the proposed technique. The results of Sobel and Canny are shown in (c) and (d) parts, respectively.

Canny results were calculated according to a proper threshold which is chosen by MATLAB heuristically in a way that depends on the input data image. Regarding Sobel technique we have chosen the threshold to be 0.08 after conducting some experimental results, since the threshold chosen by MATLAB gives bad results compared with the chosen threshold.

As illustrated in the figures the proposed technique generally lays between Sobel and Canny operators in that it provides more detailed edges than Sobel but not as many details as Canny. Moreover, one can notice that the edges obtained by the proposed method are smoother and have less noise than the ones obtained by Canny and Sobel. When comparing between our results and that of Canny we notice that the provision of more detailed edges by Canny is not always appropriate such as the case with Butterfly image. This is due to giving more detailed edges could scramble the general shape of the original image. Consequently, for Butterfly image, our technique gives results that are better than the results of both Sobel and Canny methods.

VI. CONCLUSION

Fuzzy logic is a powerful tool that can be used to manage ambiguity and uncertainty. Since the decision whether to consider a pixel as an edge or not is based on uncertainty, fuzzy logic can be used to detect image edges. In this paper a new fuzzy approach is proposed in which two fuzzy phases are used to detect edges. Image pixels are evaluated according to four directions; horizontal, vertical, diagonal, and inverse diagonal.

A 3χ3 mask is applied on the image and for each mask we define two types of intensity distances. According to these distances the fuzzy inference will locate edges according to the four directions.

Experimental results show the merits of the proposed approach compared with Sobel and Canny especially as it tends to show more details than Sobel and less edges, which could be noisy, than Canny.

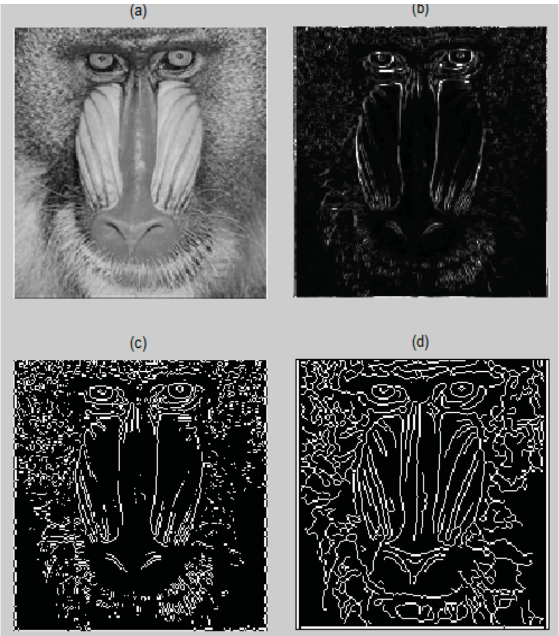


Figure 7: Baboon Image, (a) The Original Image, (b) Proposed Technique Result, (c) Sobel with Threshold = 0.08, (d) Canny with Proper Threshold Chosen by MATLAB.

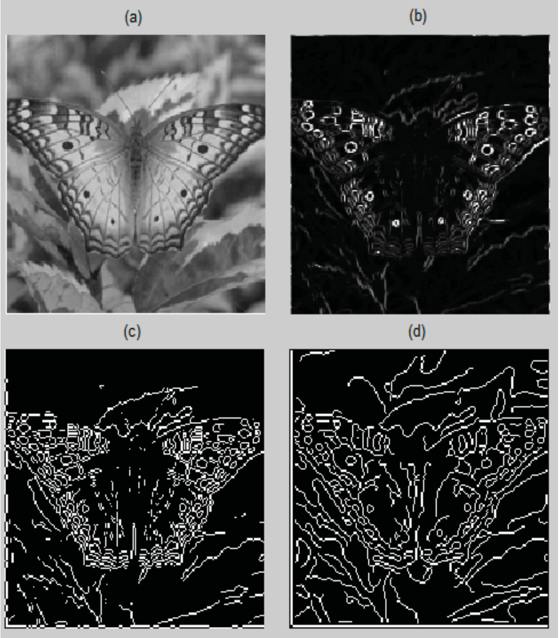


Figure 8: Butterfly Image, (a) The Original Image, (b) Proposed Technique Result, (c) Sobel with Threshold = 0.08, (d) Canny with Proper Threshold Chosen by MATLAB.

REFERENCES

[1] A. Sleit, A. Abu dalhoum, M. Qatawneh, M. Al-Sharief, R. Al-Jabaly, and O. Karajeh, "Image clustering using color, texture and shape features," KSII Transactions on Internet and Information Systems (TIIS), vol. 5, no.1, 2011, pp. 211-227.

[2] A. Sleit, H. Saadeh, I. Al-Dhamari, and A. Tareef, “An Enhanced Sub image Matching Algorithm for Binary Images," Recent Advances In Applied Mathematics, 2010, pp.565-569.

[3] A. Sleit, S. Abusharkh, R. Etoom, and Y. Khero, "An Enhanced Semi-Blind DWT–SVD-Based Watermarking Technique For Digital Images. The Imaging Science Journal, vol.60, no.1, 2012, pp.29-38.

[4] R. Maini, and H. Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques," International Journal of Image Processing (IJIP), vol.3, no.1, 2009, pp. 1-11.

[5] L.A. Zadeh, "Fuzzy sets," Information and Control, vol.8, no.3, 1965, pp. 338–353.

[6] L. Hu, H. Cheng, and Z. Zhang, "A High Performance Edge Detector Based on Fuzzy Inference Rules," Information Sciences, vol. 177, 2007, pp. 4768–4784.

[7] A. Alshnnawy and A. Aly, "Edge Detection in Digital Images Using fuzzy Logic Technique," World Academy of Science, Engineering and Technology, vol. 51, 2009, pp. 178-186.

[8] W. Barkhoda, F. Akhlaqian, and O. Shahryari, "Fuzzy Edge Detection Based on Pixel’s Gradient and Standard deviationValues," Proceedings of the International Multiconference on Computer Science and Information Technology, Mrągowo, Poland, October 12–14, 2009, pp. 7 – 10.

[9] L. Liang and C. Looney, "Competitive Fuzzy Edge Detection," Applied Soft Computing, vol. 3, 2003, pp. 123-137.

[10] M. Anver and R. Stonier, "Evolutionary Learning of a Fuzzy Edge Detection Algorithm Based on Multiple Mask," Complexity International, vol. 12, 2008, pp. 1-13.

[11] C. Kang and W. Wang, "A Novel Edge Detection Method Based on the Maximizing Objective Function," The Journal of the Pattern Recognition Society, vol. 40, 2007, pp. 609-618.

[12] J. Jiang, C. Chuang, Y. Lu, and C. Fahn, "Mathematical-Morphology-Based Edge Detectors for Detection of Thin Edges in Low-Contrast Regions," IET Image Process, vol. 1, 2007, pp. 269–277.

[13] http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm, accessed on 8/6/2016.

Optimal Design and Material selection of Heat Exchanger in Nylon 66 Plant

Saad Ahmed

Mechanical and Aerospace Engineering Department

College of Engineering

University of Missouri Columbia, Columbia, USA

Abstract

Nylon 66 is one of the most important synthetic polymer nowadays. Although the production system is well established and economized, the operating cost still challenging because of high loss of heat exchanging and frequent replacement of the main heat exchangers. The optimal design of Shell and Tube heat exchanger reactor is based on design model parameters along with physical properties for materials of construction. Aspen Tech Dynamic software is one of the computerized tool of process equipment design and it utilizes the process conditions and materials of construction to conclude for the optimal materials that enchases heat exchanging and product quality. Heat transfer models, fouling factor and other resistances are combined with process condition, physical properties of materials and mechanical design to yield the optimal design of shell and tube heat exchanger reactor.

Keyword: *Shell and Tube heat exchanger, material, optimal design, Aspen Tech.*

1. Introduction

Beginning with the first patent issued March 3, 1931, the heat exchanger as we know it today has evolved to become a thermal tool used in a variety of situations [1]. Heat exchangers are essential element of industry to economize the heat required for production. The cost benefit of the whole process depends on heat transfer and recovering efficiency. However, optimization of design and material of construction is challenging and time consuming in calculation of operating costs. Performance of Shell and Tube (SAT) eat exchanger depends on the operation parameters and materials of construction. Other minors like fouling factor, thermal conductivity, etc. are also required to achieve accurate results for the design requirements[2] The most commonly used type of heat-transfer equipment is the ubiquitous shell-and-tube SAT heat exchanger[3]. Material selection is one of the challenging tasks in designing of SAT heat exchanger. Several researchers have conducted studies and projects on this topic and considered different approaches to achieve the optimal design. Arsenyeva et al [4] considered the optimal design of a multi-pass plate-and-frame heat exchanger with mixed grouping of plates. The optimizing variables include the number of passes for both streams, the numbers of plates with different corrugation geometries in each pass, and the plate type and size[4]. Benarji et al. [5] has presented a considerable methodology for optimal design of shell and tube heat exchanger for high heat performance and low pressure drop with a new design of finned tubes. Saunders [6] presented practical approach on heat exchanger simple design factors that enable the method of fixing set of geometrical parameters. Gaddis and Gnielinski [7] presented a new procedure for calculating the shell-side pressure drop, which is based principally on the Delaware method. However, there are no studies conducted on optimal material of SAT heat exchanger for high temperature and pressure polymer production. The aim of the present study is to develop a comprehensive optimization study on materials used based on severe operation condition.

2. Methodology

Aspen Shell & Tube Exchanger Software [8] has been used to obtain the optimal design of SAT heat exchanger for Nylon 66 polymer production. Figure 1 shows a flow diagram for Nylon 66 plant.

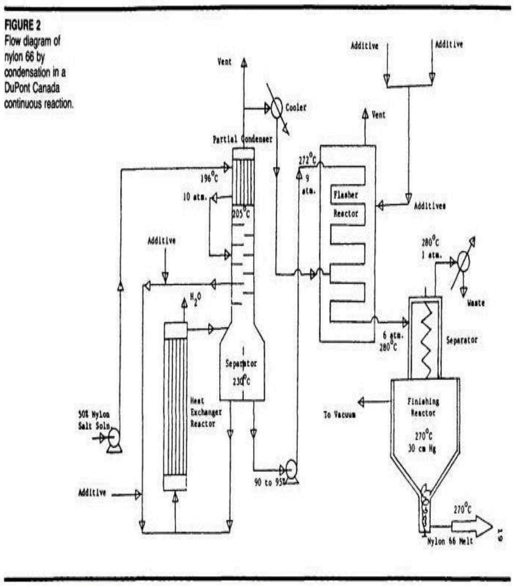


Figure 1. Process flow diagram of Nylon 66 plant

Nylon 66 (N) is manufactured from two raw materials, 1, 6 hexamethylene diamine (A) and adipic acid (B) both contains six carbon atoms from petroleum. These materials are combined to form nylon salt that is dissolved in water and transferred to rotary mill.

The nylon salt solution is concentrated by evaporation in large evaporators and sent to a SAT heat exchanger reactor, the target equipment of the present study, that operates at 210 C and 1.8 MPa The chemical reaction, polymerization to Nylon 66, takes place in this SAT reactor heat exchanger and the molten product then has sent to the following finishing processes.

Operating at the above conditions of temperature, pressure and viscosity requires an effective capital and operating cost due to high grade materials that should be used to meet the operation requirements. It has been reported that designers save between 10-30% on equipment costs by effectively designing their exchangers using Aspen Exchanger Design & Rating[9]. The following inputs and software calculations will try to implement a comprehensive design to obtain the optimum design and cost arrangement meet the process limitations. The software solves and produce SAT heat exchanger geometry and performance data. Figure 2 show input data to the SAT exchanger reactor.

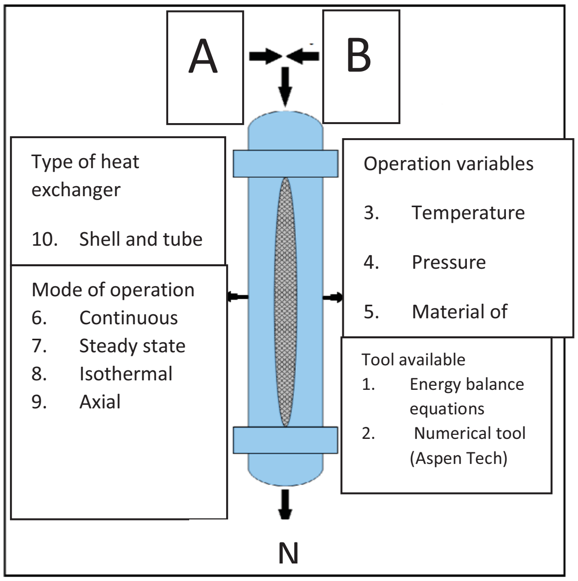


Figure 2. Schematic diagram of Heat Exchanger parameters

3. Mathematical modelling of SAT heat exchanger reactor

Based on literature review, in this study 407, 440 and 447 C steel material have been selected for the shell and 316 and 316 L modified Stainless steel for the tube bundle. These materials has been proved as an excellent materials for high temperature and pressure operation of viscous solutions. Aspen Plus Dynamics version is a computer-aided tool that utilizes the input data shown in Figure 2 to solve for the optimum design results.

Coulson and Richardson had developed a correlation for SAT heat exchanger, that is:

Db = Do.*(Nt / a)^(1 / b)

Where

Db: diameter of tube bundle,

Do: outside diameter of a tube,

Nt: number of tubes,

a and b are constants of tube passes.

For the present study 4 tube passes is considered, corresponds to a= 0.643 and b = 3.103. This calculation yield the sizes of tube bundle only. The clearance between the shell and the bundle has to be added to obtain the shell diameter. A clearance of 60 mm[9] is considered for the present study.

The heat balance of SAT heat exchanger reactor is given by

$$\mathbf{Q = U\ A\ \Delta T_{lm} = w\ C_{p(t)}(t_2 - t_1) = W\ C_{p(s)}(T_1 - T_2)} \qquad (1)$$

Where:

Q heat transferred per unit time (kJ/h)

U the overall heat transfer coefficient (kJ/h-m^2 °C)

A heat-transfer area (m^2)

ΔT_{lm} log mean temperature difference (° C)

Cp(t) tube side heat capacity ((kJ/kg-°K)

Cp(s) shell side heat capacity ((kJ/kg-°K)

w liquid flow rate, tube side (kg/h)

W liquid flow rate, shell side (kg/h)

The log mean temperature difference $\mathbf{\Delta T_{lm}}$ **f**or countercurrent flow is given by:

$$\mathbf{\Delta T_{lm} = \frac{(T_1 - t_2) - (T_2 - t_1)}{\ln \frac{(T_1 - t_2)}{(T_2 - t_1)}}} \qquad (2)$$

Where

$\mathbf{T_1}$ inlet shell side fluid temperature

$\mathbf{T_2}$ outlet shell side fluid temperature

$\mathbf{t_1}$ inlet tube side temperature

$\mathbf{t_2}$ outlet tube-side temperature

The corrected temperature difference is obtained by applying a correction factor to log mean temperature for true counter current.

$$\mathbf{\Delta T_m = Ft\ \Delta T_m} \qquad (3)$$

Where Ft is the correction factor

With even number of tube passes, Kern relationship can be applied and temperature plot with correlation. The correction factor Ft for the two heat exchangers which has 1 shell pass and 4 tube passes can be estimated using the following relationship[10]:

$$F_t = \frac{(R^2+1)^{0.5}\ \ln[(1-S)/(1-RS)]}{(R-1)\ln[(2-S\,(R+1-(R^2+1)^{0.5})/(2-S\,(R+1-\sqrt{(R^2+1)}))]} \qquad (4)$$

The fouling factors estimated will have a significant effect on the design of SAT heat exchanger reactor.

Pressure drop has to be calculated and considered in the design of SAT heat exchanger reactor. It can be calculated in the tube side of SAT heat exchangers by correlations for pressure drop in pipelines. Beggs-Brill correlations would be used for this purpose. Then, it could be connected to the flow rate of Nylon 66 via a correlation for a flow rate. These correlations of flow rate do not require data on design geometry but acquiring the heat exchanger effect on pressure in one parameter as follows[11]:

$$\Delta P = \frac{K(Mass\ flow)^2}{density} = kW^2\,\frac{\left(\frac{1}{\rho_{in}} + \frac{1}{\rho_{out}}\right)}{2} \qquad (5)$$

Input: Parameters of Modeling

Inputs

Hot Side

Fluid name		Demineralized Water
Flow		
Total	Kg/hr	543,000
T, in	°C	210
T, out	°C	55

Pressure, in	Mpa	1.2
Press Drop	Mpa	0.05
Fouling Resistance	m^2-hr-°C/kJ	0.0007

Cold Side

Fluid name		Nylon 66
Flow Total	Kg/hr	2,342,000
T, in	°C	25
T, out	°C	65
Pressure, in	Mpa	1.3
Press Drop	Mpa	0.02
Fouling Resistance	m^2-hr-°C/kJ	0.0007

Inside surface area	m^2	528.12
Overall U based on inside area	kJ/h-m^2-°C	543.93
Average temperature, shell side	°C	45.15
Average temperature, tube side	°C	63.83
Mean wall temperature, inside	°C	67.82

Physical Properties 304 Stainless steel

Specific Heat (0-100°C)	500	J.kg-1.°K-1
Thermal Conductivity	16.2	W.m -1.°K-1
Thermal Expansion	17.2	mm/m/°C
Modulus Elasticity	19.3	GPa
Electrical Resistivity	7.23	μohm/cm
Density	8.0	g/cm^3

Physical Properties 316 Stainless steel

Specific Heat (0-100°C)	500	J.kg-1.°K-1
Thermal Conductivity	16.3	W.m -1.°K-1
Thermal Expansion	15.9	mm/m/°C
Modulus Elasticity	193	GPa
Electrical Resistivity	7.4	μohm/cm
Density	7.99	g/cm^3

Physical Properties A105 carbon steel

Density (lb / cu. in.)	0.284
Specific Gravity	7.9

Specific Heat (Btu/lb/Deg F - [32-212 Deg F])	0.107
Melting Point (Deg F)	2740
Thermal Conductivity	360
Mean Coeff Thermal Expansion	6.7
Modulus of Elasticity Tension	30
Modulus of Elasticity Torsion	11

Physical Properties A106 carbon steel

Density (lb / cu. in.)	0.284
Specific Gravity	7.9
Specific Heat (Btu/lb/Deg F - [32-212 Deg F])	0.107
Melting Point (Deg F)	2740
Thermal Conductivity	360
Mean Coeff Thermal Expansion	6.7
Modulus of Elasticity Tension	30
Modulus of Elasticity Torsion	11

No of tubes = 116

Length of the tubes = 3.2 m

Tube diameter = 75 mm

Tube pitch = 45mm

Clearance = Pt – do =75 – 45 =30

Tube layout = 120

Shell length = 3.5 m

Shell diameter = 2.6 m

Thickness = 9mmm

Materials: 105, 106 and 107 C steel for the shell

304 and 316 Stainless Steel for the tube bundle.

The input parameters have processed in the CAD of Aspen Dynamic and each option for materials of construction has been considered for the mechanical design targeting the optimal performance. Figure 3 shows the simulation flow diagram of the process with the input parameters available for each equipment.

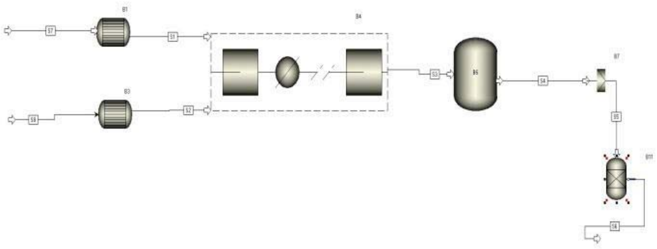


Figure 3. Aspen Dynamic simulation of SAT heat exchanger reactor

Aspen SAT Exchanger uses Microsoft Excel to explore the SAT exchanger reactor geometry and performance detail. Through the output of the software, one will be able to identify the possible mechanical design that gives the most economical production of Nylon 66 with the least energy requirements. This approach is similar to a statistical approached that has been consider by Habimana [12]. **Product parameters: Output** for 105 steel for the shell and 316 Stainless Steel for the tube bundle:

Fluid Name:	Nylon 66			
Manufacturer:				
Description:	pure component		Heat Vap	
Operating range:		°C minimum	°C maximum	BP
K	Temp. °C	Density kg/cu.m.	Sp. Heat KJ/kg-°K	Conduct. W/m-°K
253.15	-20	732.52	2.13	0.142
293.15	20	702.52	2.22	0.131
333.15	60	670.59	2.32	0.118
373.15	100	636.16	2.44	0.106
393.15	120	617.75	2.51	0.099
413.15	140	598.35	2.59	0.092

Thermal Conductivity	Btu/h-ft-°F	
Latent Heat	Btu/lb	
Inlet Pressure	psig	29.00754
Velocity	ft/s	

Press Drop Allowed	psig	1.5	10
Fouling Resistance	ft²-h-°F/Btu	0.000499687	0.000499687

The obtained results showed that using 316 Stainless steel for tube side and 105 C steel for shell side have enhanced the heat exchanging in SAT heat exchanger reactor and make the reaction process easier in terms of controlling temperature and viscosity of Nylon 66.

3. Conclusions

In this work, new approach in selecting materials for heat exchanger design has been considered and examined for the production of Nylon 66 polymer. Heat transfer equation, design parameters and physical properties are principal elements in this approach and the aid of Aspen Tech software is vital. Material

References

1. Derek J. Quade1, Michael A. Meador1, Euy-Sik E. Shin2, James C. Johnston1, Maria A. Kuczmarski1. The Design, Fabrication, And Testing Of Composite Heat Exchanger Coupons
2. S.Y.Sawant, Mr.Sagar E. Experimental analysis of advanced materials for Anticorrosive Heat Exchanger Journal of Mechanical and Civil Engineering (IOSR-JMCE) ISSN: 2278-1684, PP: 52-57.
3. Sinnott, R. K. Coulson & Richardson's Chemical Engineerings Chemical Engineering Design, revised 2nd ed.; Butterworth- Heinemann: Oxford, U.K., 1996; Vol. 6
4. Olga P. Arsenyeva b,*, Leonid L. Tovazhnyansky a, Petro O. Kapustenko a, Gennadiy L. Khavin. Optimal design of plate-and-frame heat exchangers for efficient heat recovery in process industries. Energy 36 (2011) 4588e4598
5. N. Benarji , C. Balaji & S. P. Venkateshan. Optimum Design of Cross-Flow Shell and Tube Heat Exchangers with Low Fin Tubes . Heat Transfer Engineering, 29(10):864–872, 2008
6. Saunders, E. A. D., Heat Exchangers: Selection, Design and Construction, Longman Scientific and Technical, New York, 1986.
7. Gaddis, S. E. and Gnielinski, V., Pressure Drop on Shell Side of Shell-and-Tube Heat Exchangers with Segmental Baffles, Chem. Eng. Process, vol. 36, pp. 149–159, 1997.
8. Aspen Technology, Inc. 1994-2016,
9. B.Jayachandriah, K. Rajasekhar. Thermal Analysis of Tubular Heat Exchangers Using ANSYS International Journal of Engineering Research. Volume No.3 Issue No: Special 1, pp: 21-25 22nd March 2014
10. A. Sommers, Q. Wang b, X. Han b, C. T'Joen c, Y. Parkd, A. Jacobi. Ceramics and ceramic matrix composites for heat exchangers in advanced thermal systemsd A review, Applied Thermal Engineering xxx (2010) 1-15
11. Juan Gabriel Cevallos. Thermal and Manufacturing Design of Polymer Composite Heat Exchangers. PhD Dissertation. University of Maryland, College Park, USA, 2014.
12. Dominique Habimana. Statistical Optimum Design of Heat Exchangers. MSc Thesis. Faculty of Technology Department of Technical Physics and Mathematics. Lappeenranta University of Technology. Finland, 2009