The Method of Component-based Image Retrieval in Document

Sung-Ill Joo¹, Sun-Hee Weon¹, and Hyung-II Choi¹
¹ Department of Global Media, Soongsil Univ., Seoul, South Korea

Abstract - A document or paper contains various types of images. This paper presents a new method for comparing images in documents or papers. This method consists of three main steps. At the first step, Images are extracted by converting a document into html format. In the second step, the extracted image is divided into subcomponents and the category of each component is labeled. In the final step, descriptive feature values are computed and such values are inserted into a database in the form of descriptors. Experimental results show that the proposed approach works very well in terms of retrieving similar images. It can even retrieve images which are partially similar to a query image.

Keywords: Image retrieval, Document similarity, X-Y Cut, MPEG-7 descriptor, Spline descriptor

1 Introduction

In general, a content-based image retrieval (CBIR) is to find similar images by comparing a query image against images in a database. Many approaches of CBIR have been proposed in the literature [1,3,5]. However, most of them do give good results only when a query image has a kind of similarity against stored ones in overall sense. If a query image has some component that is quite similar against stored ones, most approaches fail to retrieve the one that matches the subpart of the query image. To solve this problem, we propose the method of component-based similarity retrieval. We divide an image into components using X-Y Cut [2] and then generate descriptors for each extracted components. These descriptors are used to compare similarities.

[Figure 1] shows the schematized flow of our approach. Firstly, an image detection module extracts images and texts separately by converting an input document (*.doc) into html format. The extracted image is divided into components by the image analysis module. The category of the components is then determined through some morphological analysis. This kind of categorizing allows a fast search during the time of retrieval.

Once the components are extracted and labeled, the next step to generate descriptors to express the components. We generate different descriptors according to the type of each category. Some of them are EHD (Edge Histogram Descriptor), CLD (Color Layout Descriptor), Spline Descriptor and Binary Descriptor.

These descriptors are inserted into database by DB insert module. The search module compares values of descriptors of a query image against those in DB at the component level, and it retrieves the most similar one.

2 Component division

As can be seen in [Figure 2], an image may contain several connected components. To extract out each component, we use so called X-Y Cut approach. This approach first converts an image into a binary image. It then uses a vertical and horizontal histogram of the binary image to segment out each connected component. The resulting components may have too many small components. We apply a size filter to remove small areas.

\[
\begin{align*}
\text{Background} & : \text{if } 3\sqrt{\sigma^2} < |I - \mu| \\
\text{Foreground} & : \text{otherwise} \\
\end{align*}
\]

The [Equation 1] is to make a binary image through thresholding operation. In [Equation 1], \( I \) is the intensity.
value of an image, \( \sigma^2 \) and \( \mu \) are its variance and mean value. Generally, a document has a white background. We found out the white background is likely to have the values of \( \sigma^2 = 25, \mu = 250 \) through experiments.

Figure 2: Result of X-Y Cut (left: original image, center: binary image, right: X-Y Cut image)

3 Category classification

We divide an image into components so that similarity can be tested at a component level. This approach gives a very good performance for finding the partially similar image. But it takes a long time for searching a similar image, because the approach needs to compute the similarity at several components. To solve this problem, we classify components into different categories according to their characteristics.

Figure 3: Hierarchical organization of categories of components

At first, morphological operations such as opening and size filtering are applied to connected components to refine their shapes. The refined components are then categorized into groups of Line, Partial Surface, and Surface. They are further categorized into detailed parts like Music Sheet, Bar Graph, Gray, Color, Binary and ETC, according to their characteristics. We have nine categories for components in total.

3.1 Morphological analysis

This analysis is to classify components into groups of Line, Partial Surface and Surface. The component is defined that each area is separated by X-Y Cut. [Figure 5] shows that the process of morphological analysis for category classification. Firstly, an opening and closing operation is applied to the binary image acquired in section 2. This operation tends to separate slightly touched components. [Figure 5(d)] presents that the result of preprocessed image in each component region. After that, we can obtain the MER (Minimum Enclosed Rectangle) in each component region through labeling. These MERs is shown in (e).

\[
\frac{\text{area}}{\text{MER}_w \times \text{MER}_h} > th_1, \text{area} > th_2, \text{MER}_w > th_3, \text{MER}_h > th_4
\]

(2)

\[
\max(\text{MER}_{\text{area}}) > \text{Component}_{\text{area}} \times 0.9
\]

(3)

[Equation 2] and [Equation 3] can be used to classify the category of component. In [Equation 2] and [Equation 3], \( \text{MER}_w \) is the width of MER, \( \text{MER}_h \) is the height of MER, and \( \text{Component}_{\text{area}} \) are the number of a foreground’s pixel in each. \( th_1, th_2, th_3 \) and \( th_4 \) are some fixed threshold values obtained through experiments.

Figure 4: The result of categorizing components

Figure 5: The process of morphological analysis
The shape of a component is to be analyzed through its MER. If there is no MER that is satisfied with [Equation 2], then it will be classified into Line. On the other hand, if component have one or more MER satisfies the condition of [Equation 2] and it satisfies the condition of [Equation 3], then the component can be classified into the group of surface, or it is classified into the partial surface. A component of left-top in (f) is classified into the Line, because foreground is removed with morphological operation. Two MERs are extracted from component of right in (f), but a biggest one can not satisfied with the [Equation 3]. Therefore, that component is classified into the partial surface.

3.2 Music Sheet analysis

We use linear features of music papers for analyzing music sheets. The linear features are extracted by Hough transform. We inspect the alignment of detected lines in order to judge whether they form a music paper or not. If five lines exist in a constant interval, we regard them as music staffs. So if a component contains more than one staffs, it is categorized as Music Sheet. Otherwise, the component is categorized as ETC.

3.3 Bar Graph analysis

We apply a bar graph analysis to the component classified as Partial Surface. (1) We first carry out color quantization to reduce the number of color levels. It has the effect of removing noises. (2) We perform labeling operation. We discard the color corresponding to the background. In general, a background has the most dominant color or the color touching the border of a component. We extract MER for each labeled color. (3) If vertical sides of MERs are aligned regularly, they are classified to belong to the same group. (4) We consider the group with the largest number of MERs as a bar group. (5) If the total area of the presumed bars is more than 70% of the whole area of a component, we take the component as a bar graph.

3.4 Dominant Color analysis

This analysis is applied to components that are categorized as Surface or components that do not meet the conditions of Bar graph. We extract the dominant color of a component[3]. The color is represented in HSV. The dominant color is used to classify the component into Color or Gray. It becomes Gray if the value of saturation S is smaller than some threshold. Otherwise, it becomes Color. Also, if the two dominant colors cover more than 92% of a component, we label the component to be Binary.

4 Descriptor generation

Once each component is classified into some category, we need to create proper descriptors to characterize the component. In this paper, we use MPEG-7 Descriptors, Binary Descriptor, and Spline Descriptor. Although there are many kinds of MPEG-7 Descriptors, we use EHD(Edge Histogram Descriptor) and CLD(Color Layout Descriptor) [4,5]. There are many literatures about EHD and CLD, so we do not explain about them here.

4.1 Spline descriptor

A Spline descriptor is extracted for components that have been classified as a Bar Graph only. (1) We extract peak points from all MERs. (2) We execute cubic spline interpolation[6] using the extracted peak. (3) We select 32 samples from the interpolated curve. (4) We normalize the samples so that they have values between 0 and 1 using the max and the min values of y position. Therefore we can obtain normalized descriptor of 32 dimensions in this way.

![Spline descriptor generation](image)

Figure 6 : Spline descriptor generation (left : original image, center : result of bar detection, right : spline descriptor)

4.2 Binary descriptor

We generate Binary descriptor for components that are classified as Binary. (1) We select the background color. The background color is the most dominant color. (2) We divide an image into 16x16 sub-blocks. (3) We count the number of pixel belonging to the background and the foreground at each sub-block. (4) We label the sub-block as 1 if its number of foreground pixels is greater than the number of background pixels. Otherwise, it is labeled as 0. So, a binary descriptor of 256 bit is generated. We can use the hamming distance to compare the similarity between two binary descriptors.

![Binary descriptor](image)

Figure 7 : Binary descriptor (left : original image, center : binary image, right : binary descriptor)

5 Search

So far we have described how to generate descriptors for components that are properly categorized. The purpose of this paper is to propose a new approach for retrieving a partially similar image. This section shows how to search the database in a suitable and stable manner in order to retrieve the image that is partially similar to a query image.

[Table 1] shows the structure of our descriptor-table in a database. The descriptor-table has a record for each
component. So, it can have several records of same filename when an image has several components.

Table 1. Descriptor table

<table>
<thead>
<tr>
<th>FileName</th>
<th>Category</th>
<th>Descriptor1</th>
<th>Descriptor2</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC1.bmp</td>
<td>Gray</td>
<td>0.023-1.0</td>
<td>0.927-1.2</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>Surface</td>
<td>0.012- ...</td>
<td>0.182- ...</td>
<td>.</td>
</tr>
</tbody>
</table>

Once a query is given, it is subdivided into components and their categories are labeled. When we search the database, we try to find the record that is the most similar to each component of a query. In this way, we can find an image that is similar at a component level.

\[
Similarity = \max(S_0, S_1, ..., S_{m-1})
\]  

(4)

\[
S_i = \frac{1}{\sum_{k=0}^{n} |Desc_D(k) - Desc_Q(k)|} (i \forall m)
\]  

(5)

[Equation 5] shows how to compute similarity between same descriptors of database and a query. For the case of Binary descriptor, the hamming distance is used for similarity. A subscription i of S is the component number, Desc_D denotes descriptor of database and Desc_Q represents the descriptor of the i-th component of a query image. Also, m is the number of components of a query image and n is the dimension of descriptors. So a similarity is calculated by the sum of differences between any two descriptors. [Equation 5] may need a certain normalizing process for some types of descriptors because they may have quite different amount of values.

Table 2. The descriptors for each category and the method of combining individual similarities

<table>
<thead>
<tr>
<th>Category</th>
<th>Desc1</th>
<th>Desc2</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Music Sheet</td>
<td>EHD</td>
<td>-</td>
</tr>
<tr>
<td>Line</td>
<td>ETC</td>
<td>Spline Descriptor</td>
<td>Desc1<em>0.3 + Desc2</em>0.7</td>
</tr>
<tr>
<td>Partial Surface</td>
<td>IBar Graph</td>
<td>EHD</td>
<td>CLD</td>
</tr>
<tr>
<td>Partial Surface</td>
<td>IColor, Partial Surface</td>
<td>EHD</td>
<td>Binary Descriptor</td>
</tr>
<tr>
<td>Partial Surface</td>
<td>IBinary</td>
<td>EHD</td>
<td>CLD</td>
</tr>
<tr>
<td>Surface</td>
<td>Color, Surface</td>
<td>EHD</td>
<td>CLD</td>
</tr>
<tr>
<td>Surface</td>
<td>IBinary</td>
<td>EHD</td>
<td>Binary Descriptor</td>
</tr>
</tbody>
</table>

[Equation 4] expresses the overall similarity of each row in [Table 1]. The \(S_{m-1}\) denotes similarity of the \((m-1)\)-th components of a query image. In this way, all the similarities between all components of a query image and a record are calculated, and the largest one among the calculated similar rates is stored in the similarity column of the [Table 1]. If all works are finished, the Similarity column is sorted in a descending order. If a certain record shows a similarity of over the predefined threshold, we can easily retrieve the corresponding image and its sub-components.

The important point is that we need to compute similarity between a record and each component of a query image only when their categories are same. [Table 2] presents which types of descriptors are used for each category. It also shows the method of combining descriptors with proper weights when computing the overall similarity. For instance, if the category is Surface and Color, we can get the overall similarity by combining similarities of EHD and CLD with weights of 0.2 and 0.8, respectively.

6 Experiment results

To evaluate the effectiveness of our approach, we have formed the Database that includes 2480 images. These images are extracted from web documents and technical papers. We have implemented our proposed Component-based Image Retrieval System(CIRS) with Visual Studio 6.0 C++ and Oracle 10g.

[Table 3] summarizes the search results of our system. For these evaluations, [Equation 6] is used.

\[
\text{Ratio of ground truth image} = \frac{R_G}{T_G}
\]  

(6)

<table>
<thead>
<tr>
<th>Category</th>
<th>Ratio of ground truth image</th>
<th>Category</th>
<th>Ratio of ground truth image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar Graph</td>
<td>0.83</td>
<td>Block Diagram</td>
<td>0.85</td>
</tr>
<tr>
<td>Shape</td>
<td>0.92</td>
<td>Map image</td>
<td>0.88</td>
</tr>
<tr>
<td>Music Sheet</td>
<td>0.75</td>
<td>ETC</td>
<td>0.91</td>
</tr>
<tr>
<td>Natural Image</td>
<td>0.91</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In [Equation 6], \(R_G\) is the number of ground truth images that are retrieved from the search module of our system. \(T_G\) is the total number of ground truth images. If an image in database has a component that is the identical with some component of a query, then it is defined to be a ground truth image.

We have confirmed that our system gives very good performance in most case. For music sheets, the retrieval rate is relatively low. It seems to be that the used descriptor is not effective for music sheet. It shows very high retrieval rate for shape and natural image. Their descriptors seem to work well for such categories.
7 Conclusion

In this paper, we have proposed the efficient approach for retrieving partially similar images. It decomposes an image into components and computes the similarity at a component level. To improve the efficiency of search process, we have classified components into several categories and we have attached different descriptors to components according to their categories.

We evaluated the performance of our system through experiments with various types of images. We have confirmed that our system gives very good results for most types of images. But it gives somewhat unsatisfactory results for images like music sheets. We hope that our approach may give better results if we can develop more appropriate descriptors.

Acknowledgement This work was supported by the Seoul R&BD Program(SS110013).

8 References


