Filtering in Spatial and Frequency Domains: Examples & Tools

Shin S. Jou
Department of Computer Science, Winston-Salem State University, Winston-Salem, NC, 27110 USA

Abstract - In this paper, we intent to do some studies on filtering in the spatial and frequency domain of digital image processing. The choice is intentional as there is diverse amount of theory and application in Digital Image Processing. The Convolution Theorem establishes the powerful mathematical foundations to look into filtering in these two different domains. One needs to decide what domains are better to work with based on the complexity of the processes to deal with images and filters. Students with good foundations in these areas can easily turn to study other topics and applications based on the natures of the problems. MATLAB & ImageJ are used for program developments.

Keywords: Digital Image Processing, Signal Processing

1. Introduction

Digital image processing encompasses processes whose inputs and outputs are images and, in addition, encompasses processes that extract attributes from images, up to and including the recognition of individual objects. Since Digital Image Processing classes normally cover great amount of topics, hence we will only discuss parts of a Digital Image Processing course; on such topics as Point Processing with intensity transformation, Spatial Filtering and Filtering in Frequency domains, and some examples of filtering for applications on edge detections in segmentation etc. Since there are many way for implementing or experimenting image processing, we need to choose some software tools, which can be more user friendly for students and application users, while still can be effective for advanced users to develop more advanced applications and algorithms. The main tools that we use in the class are MATLAB with digital image processing tool box, and Java based ImageJ of National Institute of Health (NIH). MATLAB has good collections of Digital Image Processing, and Signal Processing toolboxes for applications. Many examples and pictures files from Gonzalez, Wood, Eddins [2, 3] and McAndrew [5] are used by students to work on the projects. References [2, 3, 4, 5] contain good sources for doing digital Image Processing Studies among others.

2. Some Concepts and Mathematical Principles

A digital image $S$ can be represented as a matrix of intensities. The intensities of each pixel can be among some color space. $S = [f(i, j)]_{MN}$, a matrix of pixel values, is represented as follow:

$$
\begin{array}{cccc}
  f(1,1) & f(1, 2) & \ldots \ldots & f(1, N) \\
  f(2,1) & f(2, 2) & \ldots \ldots & f(2, N) \\
  \vdots & \vdots & \ddots & \vdots \\
  f(M,1) & f(M, 2) & \ldots \ldots & f(M, N)
\end{array}
$$

2.1 Intensity Transformation Functions

Consider a single pixel $f(i, j)$ of the image $S$, its value is some value of intensity, such as gray level of the pixel. Of course, the intensity can be some component of the color spaces such as the (RGB) model. A mapping of the form, $G = T(F)$, can be considered as $G$ a new image of the same size with each pixels intensity modified to a new intensity to form the new image. $F$ serves as the input image and $G$ is the output image.

2.1A: Functions imadjust() & stretchlim()

MATLAB `imadjust()` and `stretchlim()` functions can adjust and stretch images. An original image can be converted to different images as follow:

Original image: $f$

![Original Image](image1.png)
Use \texttt{imhist()} to construct the histogram.

As for the MATLAB instructions used in the design of the MATLAB functions see ([3, 10]).

\subsection*{2.2 Spatial Filtering & Spatial Convolution}

In general, linear filtering of an image $f$ of size MxN with a filter mask $m$ of size mxn is given by the expression:

$$g (x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s, t) f(x+s, y+t)$$

The general implementation for filtering an MxN image $f$ with a weighted averaging filter \texttt{mask}, $m(s, t)$ of size mxn is given by the expression

$$g (x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s, t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}$$

Associated with spatial filtering is the spatial convolution. Let $m$ be a mask, and $f$ an image, then $m*f$ denotes the convolution of $m$ and $f$ and is defined as:

$$(m*f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(-s,-t) f(x+s, y+t)$$

It is equivalent to:

$$(m*f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} m(s,t) f(x-s, y-t)$$

Convolution & Fourier Transform are fundamentally important mathematical entities which have great importance in mathematics, science and engineering applications. From the formulas, we see that if we rotate the filter mask by 180 degrees and multiply, then convolution is the same as filtering. Filtering and convolution have been implemented in MATLAB tool boxes. Convolution and filtering are related to each other in MATLAB as:

\texttt{conv2(image, mask) = filter2(rot90(mask, 2), image)}
2.3 Fourier Transforms and Frequency Domain Filtering

The Frequency domain filtering depends on the Convolution Theorem on Convolutions and Fourier Transforms. Here we like to mention some important features of Fourier Transforms.

Fourier Transforms are represented as follows:

\[ F = \mathcal{F}(f)(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i (ux + vy)} dx dy \]

\[ f = \mathcal{F}^{-1}(F)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{2\pi i (ux + vy)} du dv \]

Here, \( \mathcal{F} \) and \( \mathcal{F}^{-1} \) serve as the Fourier Transform and its Inverse Transform between Spatial and frequency domains. For finite discrete digital image, DFT are represented in the following form:

\[ F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-2\pi i \left( \frac{xu}{M} + \frac{yv}{N} \right)] \]

\[ f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[2\pi i \left( \frac{xu}{M} + \frac{yv}{N} \right)] \]

Fourier Transforms establish the relationships between Spatial and Frequency Spaces.

One application of the Convolution Theorem is that we can perform time-domain convolution using frequency domain multiplication.

The **CONVOLUTION THEOREM** can be best represented by the following two formulas.

\[ \mathcal{F}(f \ast g) = \mathcal{F}(f) \mathcal{F}(g), \quad \text{likewise,} \]

\[ f \ast g = \mathcal{F}^{-1}(\mathcal{F}(f) \mathcal{F}(g)), \quad \text{here } \ast \text{ is the convolution.} \]

3. Spatial filtering Examples

We will use MATLAB as the main tool for showing examples on Spatial and Frequency Domains filtering.

Two types of filters are important for image filtering. High-frequency components are those areas with high changes of intensities over distance. The other type is the low-frequency with slower changes with respect to distances. Hence, a filter is a high-pass if it passes over high-frequency components and reduces or eliminate low-frequency components. Equivalently, it is a low-pass filter if it passes over the low-frequency components, and reduces or eliminates high frequency components. Here we show some low-pass and high pass filters examples in MATLAB.

An average filter is a low-pass filter and it tends to blur the edges. The following are some pictures with the original picture \( f \) marked as original. Using fspecial MATLAB function, with the following average filters:

**AVERAGE FILTERS**

\( f1=\text{fspecial('average')}; \) \( f9x9=\text{fspecial('average',9)}; \)

\( f17x17=\text{fspecial('average',17)}; \)

Filtering original picture \( f \) with \( f1 \), \( f9x9 \), and \( f17x17 \) filters will produce the following 4 pictures with different degrees of blurring.

**First one: Original picture \( (f) \)**

![Original picture](image1)

![Filtered picture](image2)
This last one has higher blur

Here are some examples of high-pass filtering. High-Pass Filter will normally sharpen the edges, as shown from the following two pictures.

**Laplacian and Laplacian of Gaussian Filters**
```
laplacianJ = fspecial('laplacian');
flaplacian = filter2(laplacianJ, f);
figure, imshow(flaplacian/100)
laplacianOfGaussian = fspecial('log');
flaplacianofGauJ = filter2(laplacianOfGaussian, f);
figure, imshow(flaplacianofGauJ/100)
```

Here is an unsharped filter

**Unsharp Filter**
```
unsharp = fspecial('unsharp', 0.5);
fu = filter2(unsharp, f); imshow(fu)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```

4. Filtering in Frequency Space

From the Convolution Theorem, we know that the filtering in the frequency domain shall follow the following procedures ([2, 3]). Assume that we are looking for the convolution (or filtering) of the image f with the filter h. They basically need to go through the processes of the following diagram. Preprocessing, post-processing and other adjustments need to be designed in the algorithms such as the sizes of images and filters etc.

Adjustments need to be done in the algorithms in things such as the resize of images and filters etc. Here we will give some examples on filtering with low-pass filters in Frequency Domain. We use examples mainly from ([3]).

4.1 Low-Pass Filtering: (Original image f)
```
(A4.1)  f = imread('airfield auto Jou.tif'); imshow(f)
```
(B4.1)

\[
[f, \text{revertclass}] = \text{tofloat}(f);
PQ = \text{paddedsize(size}(f));
[U, V] = \text{dftuv}(PQ(1), PQ(2));
D = \text{hypot}(U, V);
DO = 0.05*PQ(2);
F = \text{fft2}(f, PQ(1), PQ(2));
H = \exp(-((D.\text{^2})/((2*(DO.\text{^2})))));
g = \text{dftfilt}(f, H);
g = \text{revertclass}(g);
\text{figure, imshow(fftshift(H))}
\]

Gaussian low-pass filter image

(C4.1) FFT Spectrum

\[
\text{figure, imshow(log(1 + abs(fftshift(F))), [ ])}
\]

(B4.2)

\[
PQ = \text{paddedsize(size}(f));
DO = 0.05*PQ(1);
H = \text{hpfilter('gaussian', PQ(1), PQ(2), DO)};
g = \text{dftfilt}(f, H);
\text{figure, imshow(g)}
\]

With value: \( DO = 0.05*PQ(1); \)

The pictures and MATLAB instructions in (A), (B), (C), (D) show how the processes of the filtering in the frequency domain as shown in the flow chart diagram above.

There are lots of design details of algorithms to achieve, for example the resize of images or filters etc. We find that ([2, 3]) contain good sources for image processing studies, with many programs implementation. Many instructions above are not parts of MATLAB toolbox programs.

4.2 High-Pass Filtering

In frequency domain, if \( H_{LP} \) is a low pass filter, then \( H_{HP} = 1 - H_{LP} \), \( H_{HP} \) a High-Pass filter

Here is one High-Pass filtering: (Original image \( f \))

(A4.2) \( f = \text{imread('airfield auto Jou.tif')}; \text{imshow(f)} \)

(D4.1) Filtered Image: \( \text{figure, imshow(g)} \)
For lighter image: $\text{imshow}(g+60)$

For lighter image: $\text{imshow}(g+100)$

The MATLAB functions such as $\text{toFloat}$, $\text{paddedsize}$, $\text{dftuv}$, $\text{dftfilt}$, $\text{hpfilter}$, $\text{dftfilt}$ etc. shown above are not parts of the Math Works DIP toolbox. They are from ([3]).

5. Conclusions

Image processing involves the manipulation, processing of the images in order for better pictorial information for human, and for other applications that can even for machines to better detect the images. Since digital images are basically a matrix of digital values, with values value in color spaces. The basic needs include the display of images for visualization, processing of images for better application etc. The processing will involve manipulations of numerical data. Although, there are many different theories and applications in DIP (digital image Processing), the first needed steps will be the filtering of image data. Since the Fourier theory tells us that we can consider processing of images and signals in more than just the Spatial Domain. It is nature to see how processing are also done in frequency domain. From the learner point of view, a good processing & development tools are of vital importance. MATLAB and NIH’s ImageJ are used for software experiments. C++, can also be an important tool for implementing algorithms. Some emphasize the needs for parallel processing ([9]) for special DIP applications. Mathematics like convolution, Fourier transforms is mentioned to establish the connections among different application domains. However, mathematics can be maintained to the minimum while examples help learner gain insights. Discrete approaches are used as Digital Image is discrete in nature.

6. References


[9] Classic HPC Development using Visual C++,
http://resourcekit.windowshpc.net/sampletutorial1.htm
