

Performance Evaluation of Edge Detection Techniques in Spatial and Frequency Domain

T. Shreekanth, R. Shashidhar, N. M. Basavaraju
 Sri Jayachamarajendra College of Engineering, Mysore, Karnataka, India

Abstract

This paper presents a comprehensive performance evaluation of different high pass filtering techniques for Edge detection in both time domain and frequency domain. The paper examines various kernels and compares the efficiency of the filtering technique against the computation time for various sizes of images with various sizes of high-pass filter kernels. We have made use of Sobel filter as the standard filter kernel against which other techniques are compared. The results clearly demonstrate that the implementation of edge detection in frequency domain is better in terms of computation time as compared to that of the time domain implementation.

1. Introduction

Image sharpening and smoothening is perhaps one of the most ubiquitously used techniques in any image processing application. While these are considered inexpensive operations, multiple use of the same may become computationally intensive, and hence may make the overall application slow. In this paper we examine and compare the performance of sharpening filters in time-domain and frequency domain. While there are many apparent computational advantages in converting an image into frequency domain to apply a filter, it is not very apparent whether the conversion is required in smaller images.

Applying the filter in time domain is straight forward, and requires the convolution operation of the filter kernel over the entire image, but applying the filter in frequency domain requires the steps as shown in figure [1].

The primary advantage of converting an image into frequency domain is the change in the operation from convolution to multiplication. Convolution is an expensive procedure requiring an asymptotic time of $O(n^2)$, but multiplication in certain architectures can be implemented in constant time.

However, the image needs to be preprocessed and converted into frequency domain using FFT, and reconstructed using Inverse FFT. Further, the use of Fourier transform works well for large images that are correctly padded. Padding the image with appropriate number of zeros is essential when

applying Fourier Transform because images are infinitely tiled in the frequency domain and filtering may produce wrap around effects [1]-[5].

Image processing \equiv filtration of 2D signals.

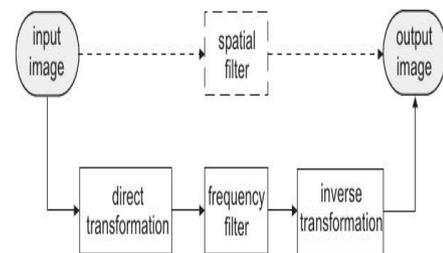


Figure 1. Image processing in frequency domain vs spatial domain

2. Filtering in Frequency Domain

Joseph Fourier said that any function can be written as a weighted sum of sines and cosines of different frequencies. The frequency domain is a space in which each image position A represents the intensity value in image B vary over a specific distance related to A. Suppose there is a value 30 at the point that represents the frequency 0.2 (or 2 periods very 10 pixels). This means that in the corresponding spatial image, the intensity values vary from dark to light and vice versa over a distance of 10 pixels, and the difference between dark and light colors would be $60 = 30 * 2$. The frequency domain is particularly interesting because it makes explicit periodic relationships in the spatial domain, and some image processing operators are more efficient and practical only when applied in the frequency domain.

$$F(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(x\frac{m}{M} + y\frac{n}{N})}$$

$$f(m,n) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(x,y) e^{j2\pi(x\frac{m}{M} + y\frac{n}{N})}$$

An image can be converted from spatial domain to the frequency domain and from frequency domain to the spatial domain respectively using the 2D discrete

Fourier Transform and inverse discrete Fourier Transform as defined by the above equations[6], [7]. While the formula shows a naïve implementation of $O(n^2)$, most implementations use Fast Fourier Transform with an asymptotic time complexity of $O(n \log n)$. The visualization of filters is more intuitive in the fact that low pass filters (image smoothing filters) and high pass filters (image sharpening filters) are better understood in the frequency domain.

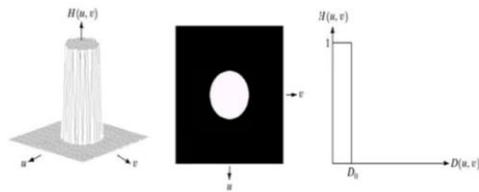


Figure 2. Perspective plot of the Ideal low pass filter in frequency domain

3. Filtering in Spatial Domain

In spatial domain, the value of the pixel in the enhanced image is some function of the pixels in the neighborhood of the pixel. The neighborhoods can be of any shape but are usually rectangular, and are called kernels.

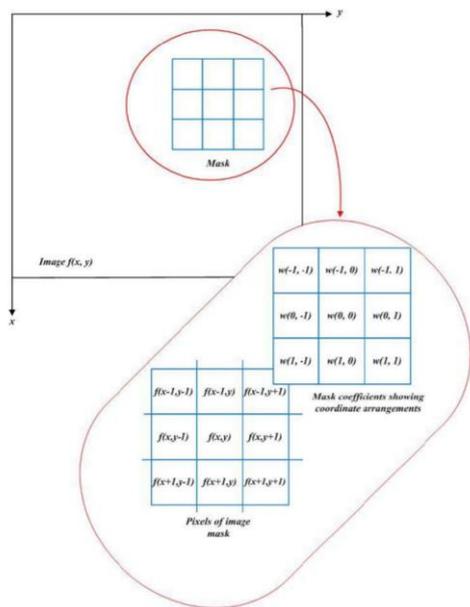


Figure 3. Spatial Filtering Illustration

Image processing in the spatial domain may be expressed as: $g(m, n) = T(f(m, n))$. Where $f(m, n)$ is the input image, T is the spatial transform operator for the modification process, $g(m, n)$ is the processed/enhanced image. The operator T is typically a single valued and monotone function that can operate on the single pixel or a selected portion of neighborhood around the pixel of interest. The basic

spatial processing consists of simply moving the filter mask (equivalent to operator T) from point to point in image.

4. Implementation

For the purpose of robust and quick implementation, we have implemented spatial and frequency domain filters in MATLAB. The filtering pipeline used is quite simple. The paper implements filtering comparison for Sobel Kernel, Laplacian Kernel, and Prewitt Kernel. For spatial filtering we have obtained the necessary kernel of 3x3 or more as required. The image is then filtered using the convolution operation. The time taken for the entire procedure is noted using tick.

For frequency domain filtering in MATLAB, the images are first preprocessed to be padded with the required no. of zeros. FFT is applied on both the kernel and the image, and displayed. The image and filter kernel are multiplied in frequency domain and then the inverse Fourier Transform returns the processed image. Time of computation is measured using tick. In both cases the absolute values of the pixels are taken after filtering to produce the appropriate intensity values. After filtering is obtained, threshold Binarization is done to improve the contrast of the image.

5. Results

In this section, we present the materials on which the edge detection technique in frequency and time domain is evaluated, the performance measures used to evaluate the algorithm, and the results obtained.



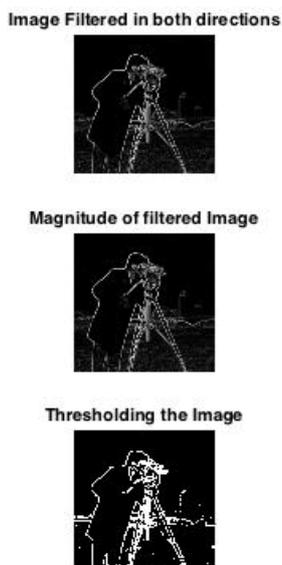


Figure 4. Example image filtered in spatial and frequency domain

As shown in Tables 1 and 2, we compared the computation time under three different criteria. There are three parameters: image size, kernel Size, and kernel itself. In each comparison, while one of the parameter is varied, the other two is kept constant. Table 1 shows the execution time for different Image sizes for Sobel filter of 3x3 kernel. It was observed that computation in frequency domain was about ten times faster and remained constant with image size, as opposed to the convolution based spatial filtering. Similarly,

Table 2 compared execution times with different kernels on an image of 256x256 and a kernel size of 3x3. Here also, it was seen that across different kernels, the frequency domain computation time was comparatively faster by about 10 times. Further, we also measured the computation time with different kernel sizes, while maintaining an image size of 256x256 and the Sobel Kernel (see Table 3). In the spatial domain with increase in kernel size, the computation also increased rapidly, on the other hand, computation in frequency domain remained constant more or less.

Table 1. Comparison of Execution Time with Sobel Kernel for Images of Different Sizes

SI No.	Image Size	Spatial Domain Time(s)	Frequency Domain Time(s)
1	256x256	0.1970	0.0203
2	128x128	0.1949	0.0198
3	64x64	0.1845	0.0196

Table 2. Comparison of Execution Time with Different Kernels for the Same Image

SI No.	Kernel	Spatial Domain Time(s)	Frequency Domain Time(s)
1	Sobel	0.1970	0.0203
2	Laplacian	0.2062	0.0205
3	Prewitt	0.2183	0.0214

Table 3. Comparison of Execution Time for the Same Image with Sobel Kernel of Different Size

SI No.	Kernel size	Spatial Domain Time(s)	Frequency Domain Time(s)
1	3x3	0.1970	0.0203
2	5x5	0.2047	0.0206
3	7x7	0.4454	0.0207

6. Conclusions

In the above results we obtained decisive data about differences in spatial domain computation and frequency domain computation. Across different comparisons we could observe that frequency domain computation is at least 10 times faster when appropriately padded. Furthermore, the computation time remained constant with changing parameters.

The computation speed may be attributed as expected to the asymptotic time complexity improvement from $O(n^2)$ in spatial domain to $O(n \log n)$ in frequency domain. Further, an interesting result was obtained with different sized kernels: As kernel size was increased computation time in spatial domain increased, but time in frequency domain remained fairly constant. With increase in kernel size the number of multiplications required in Spatial domain increases exponentially, while in frequency domain, since kernel padding is independent of kernel size, the multiplications required remain constant, and hence the constant time.

7. References

[1] Shanmugam, K. Sam, Fred M. Dickey, and James A. Green. "An optimal frequency domain filter for edge detection in digital pictures." IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1979): 37-49.

[2] Davis, Larry S. "A survey of edge detection techniques." Computer graphics and image processing 4.3 (1975): 248-270.

[3] Shrivakshan, G. T., and C. Chandrasekar. "A comparison of various edge detection techniques used in image processing." *IJCSI International Journal of Computer Science Issues* 9.5 (2012): 272-276.

[4] Sonam Saluja, Aradhna Kumari Singh, Sonu Agrawal, "A study of Edge-detection methods." *IJARCCCE International Journal of Advanced Research in Computer and Communication Engineering* Vol 2, Issue 1, January 2013.

[5] Mamata Juneja, Parvinder Singh Sandhu, "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain." *International Journal of Computer Theory and Engineering*, Vol 1, No. 5, December 2009.

[6] R.C.Gonzalez and R.E. Woods, *Digital image processing*, "Digital Image Processing", 2nd edition, Prentice Hall, 2002.

[7] Anil K.Jain, "Fundamentals of Digital Image Processing", First edition, PHI Learning, 2011.