Applications of Fourier Analysis in Image Recovery
TJHSST Senior Research Project
Computer Systems Lab 2009-2010

Kang Guo

January 20, 2010

Abstract

The goal of this project is to explore and implement image deblurring techniques. Many of these techniques involve some sort of an image transform, and the most commonly used one is the Fourier Transform. A Point Spread Function, which is the Fourier Transform of the blur kernel, can be applied to an image in the frequency domain after it has been transformed to create a blurry image. The opposite of this can be done by applying a transform to a blurry image, and after removing the point spread function from the frequency domain, a deblurred image can be obtained.

Keywords: Discrete Fourier Transform, Fast Fourier Transform, Point Spread Function, Blur Kernel

1 Introduction

Motion blur in images is a common problem for professionals in various fields. When the image is deblurred, the usefulness of the image increases. Parts of the image that were difficult to identify can be rendered to effective clarity. This project will explore and implement image deblurring techniques. By implementing these techniques, users can efficiently remove blur from an image.

2 Background

Regarding many image processing techniques, transforming the image from a spatial domain to a frequency domain is very helpful. In the area of image deblurring, a Fourier Transform is useful in creating the frequency domain image. The Discrete Fourier Transform only describes the frequencies contained in the spatial domain of the image, as opposed to a Continuous Fourier Transform which will describe a continuous range of frequencies.

The Discrete Fourier Transform is not sufficiently fast for an image of size 256x256. Instead, speed improvements can be obtained by implementing a Fast Fourier Transform. This transform relies on the fact that the Discrete Fourier Transform is seperable. Originally, N, 2-dimensional transforms would have to ben done. With the Fast Fourier Transform, 2N 1-dimensional transforms have to be done, resulting in an efficiency on the order of $N \log N$, compared to N^2 for a Discrete Fourier Transform. On a 256x256 pixel image, speed improves from upwards of ten minutes to just five seconds.

After applying the Fourier Transform, a blur kernel can be applied to the frequency domain image, and after applying an inverse transformation, a blurred image will be obtained. This process is called convolution and requires that the Fourier Transform of the blur kernel, known as the Point Spread Function be multiplied to the Fourier Transform of the original image. The same process can be done to remove blur from an image. This is known as deconvolution. Instead of applying the blur kernel to the frequency domain image, it can simply be removed, and then transformed back to spatial domain to produce the deblurred image. This process is simply the inverse, instead of multiplying the transformed images, they can be divided. Note that when working with images that have been Fourier Transformed, image values are stored as complex numbers. Therefore, the identity $e^{i\theta} = \cos\theta + i\sin\theta$ is useful.

Attempting to traditionally deblur an image will result in unwanted noise and ringing artifacts. However, a finite number of Fourier basis functions are able reconstruct the image without much data loss. In determining the blur kernel, iterating between updating the blur kernel and the estimated latent image will ultimately allow the two to converge and produce an acceptable deblurred image. The Richardson-Lucy algorithm is sufficient in blind deconvolution, in which the PSF is not known. In the absence of noise, this algorithm functions well, and by increasing the number of iterations, the quality of the image increases. The formula is as follows, where h is the PSF,

f is the original image, and g is the blurred image. H^* is the adjoint operator of H, where $Hf = g$.

$$f^{n+1} = f^n H^* (\frac{g}{(Hf^n)})$$

Deblurring can also be done with a pair of images: a blurry one and a noisy one. By removing noise from the noisy image, an estimate of what the final image should look like can be obtained. This helps in the estimation of the blur kernel. Again, iterations between estimating the blur kernel, residual deconvolution, and de-ringing will ultimately allow the image to converge and produce an image of acceptable quality.

3 Development

Code for my project will be done primarily written in the C programming language. I will also need to use imagemagick to convert various images into the pgm format, which I can use to directly read color values. Testing will revolve around applying forward and inverse Fourier Transforms to test the functionality of the image transforms. Ultimately, after applying blur kernels to the frequency domain image, I will be able to visually verify the effect of the deblurring program.

In applying image deblurring techniques, I will be primarily focusing on eliminating ringing artifacts from a transformed image. A secondary focus will involve determining the blur kernel that is applied to the frequency domain image. This will allow for an image to be deblurred without knowledge of how it was blurred in the first place.

The Fourier image is often displayed with F(0,0) in the center of the image. In the frequency domain with this particular shifting, the further away from the center of the image, the higher the frequency is. Visually, one can detect gradients in the Fourier transform that corresponds to the original image in the spatial domain. For example, this image (Fig. 1) has two dominating directions, one along the x-axis and another along the y-axis. This can be seen in the Fourier image (Fig. 2) shown by the strong lines intersecting at the middle, and in the original image shown by the border of the mirror.

Fig. 3 outlines the image blurring process, in which the Point Spread Function and the transformed image are multiplied. Different types of blur can be modeled with the Point Spread Function, as seen in Fig. 5 and Fig.

6. Fig. 4 outlines the image deblurring process, which is simply the inverse of the blurring process. The values are divided instead of multiplied, and the image is returned to its normal state. In this process, some very small Fourier values are divided, resulting in amplification of noise.

4 Discussion

Images serve various purposes in many different fields, and the clarity of an image is almost universally preferred. Thus, a program to effectively remove blur in images would be useful in any subject area. Such functionality would allow photographers and image editors to be able to remove blur and increase clarity of images. Casual image enhancement would allow photographers to take more presentable pictures with less blur. Deblurring photographs taken by roadside cameras would allow law enforcement to clearly read license plate numbers.

References

[1] Q. Shan, J. Jia, and A. Agarwala. "High-quality Motion Deblurring from a Single Image" *ACM Trans. Graph. 27*, 3, Article 73 (August 2008).

[2] L. Yuan, J. Sun, L. Quan, and H. Shum. "Image Deblurring With Blurry/Noisy Image Pairs" *ACM Trans. Graph. 26*, 3, Article 1 (July 2007).

[3] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. "Image Transforms - Fourier Transform", 2003. `http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm`

[4] T. Chan and J. Shen. "Theory and Computation of Variational Image Deblurring" *WSPC/Lecture Notes Series* (November 2005).

[5] S. Allon, M. Debertrand, and B. Sleutjes. "Fast Deblurring Algorithms", 2004. `http://www.bmi2.bmttue.nl/image-analysis/Education/OGO/0504-3.2bDeblur/OGO3.2b_2004_Deblur.pdf`

Figure 1: Original Image

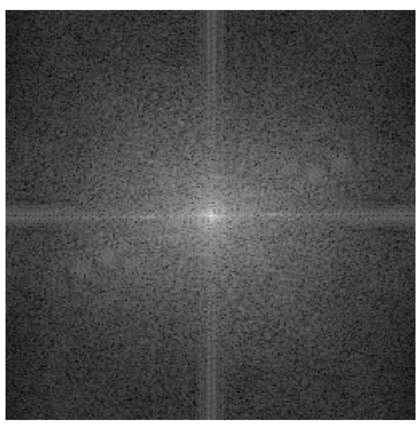

Figure 2: Fourier Transformed Image

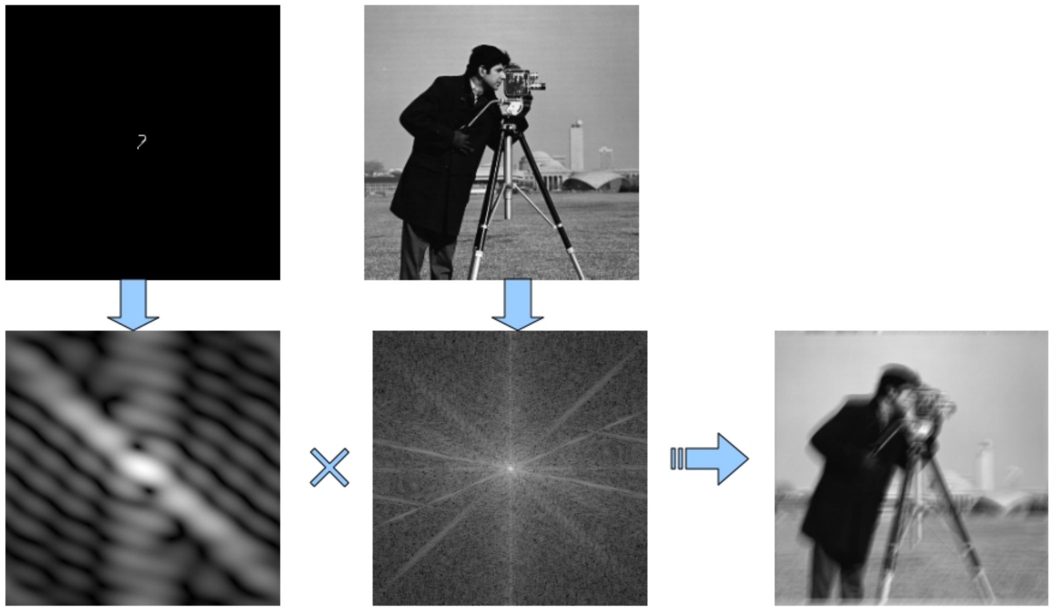

Figure 3: Blurring an Image

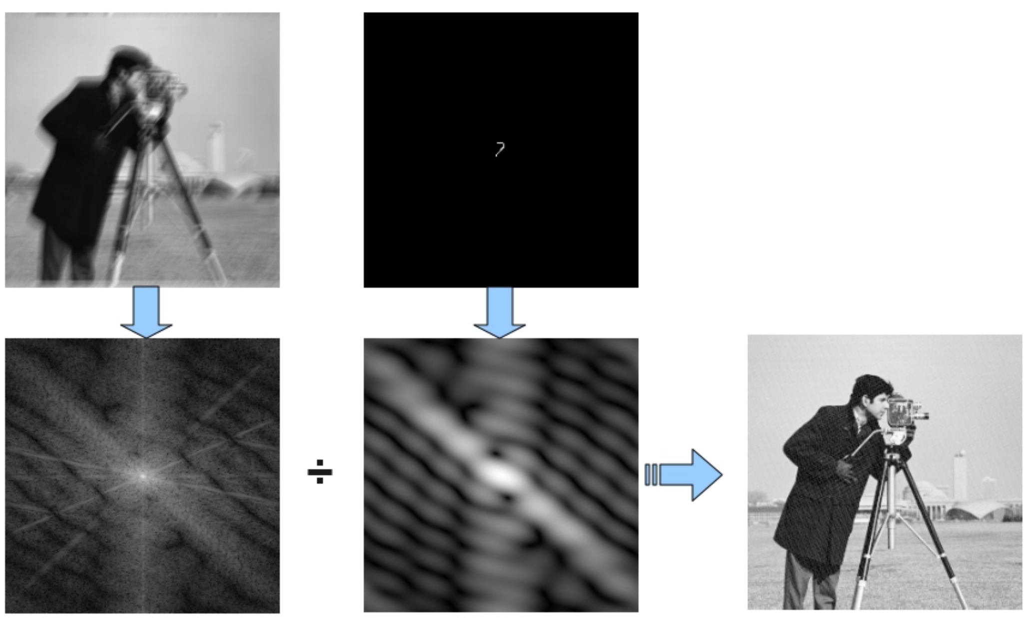

Figure 4: Deblurring an Image

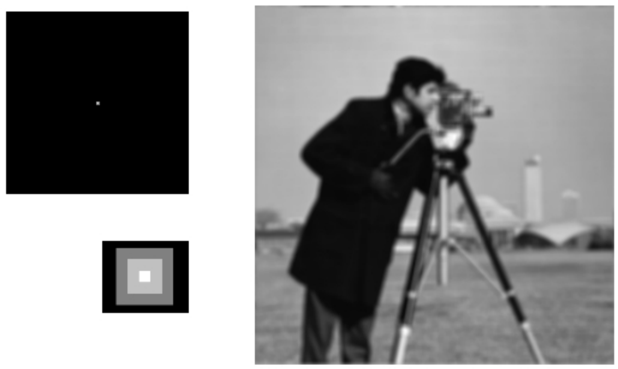

Figure 5: Gaussian Blur

Figure 6: Linear Blur