

Implementation of Image Deblurring Techniques in Java

Peter Chapman

Computer Systems Lab 2007-2008

Thomas Jefferson High School for Science and Technology
Alexandria, Virginia

January 22, 2008

Abstract

Families, friends, professionals, and enthusiasts take countless numbers of photographs every day, and inevitably, many images suffer from some sort of "blurring." A program with the power to take a blurred image and create a much crisper and clearer "deblurred" form would be immensely valuable. Law enforcement attempting to read the license plate off a blurred photo, or a family attempting to improve the clarity of their grandfather's smile would find such a piece of software useful. In my implementation, I attempt to deblur images suffering from simple types of motion blur using the alternate domains granted by the use of Fourier transformations and a basic understanding of image deconvolution.

Keywords: Fourier Transformations, Spatial Domain, Frequency Domain, Phase Domain, Blind Image Deconvolution, Fast Fourier Transformation, Cooley-Turkey Fast Fourier Transformation Algorithm, Discrete Fourier Transformation, Image Deblurring

1 Introduction

Photographs are utilized in many different fields for a wide variety of purposes; Regardless of the subject area, a blurred image is often a useless one.

A program with the ability reverse the such damages would be extremely useful. Such functionality could be bundled into the software of cameras with adjustments performed automatically after each shot, or an available feature on standard photo manipulation software. Due to the complexities involved in the image deblurring process my research is focused on blind image deconvolution, where the application is given a general overview of how the image was blurred presumably by the user. In order to further simplify the project further, my application will only be built to handle images suffering from motion blur.



Figure 1: Photograph from a taxi suffering from motion blur (Raskar, Agrawal and Tumblin).

2 Background

Due to the value of a program that can deblur images, many have tried to create an all-purpose deblurring program, but few have found much success with a general approach. The tendency in the field is to focus on motion blur and narrow the application of the program in order to get a more effective method that often applies to a smaller range of tasks. In one such project, the researchers used a modified camera with motion-sensing technology. Upon taking a picture, the researchers were able to read the data collected from the motion sensors and calculate how the image was blurred (Raskar, Agrawal and Tumblin). The results were phenomenal (Figures 1 and 2).

However, it is possible to have some success with more general applications such as that found in the work of M. D. Cahill. His program, called Unshake, attempts to reverse any type of motion blur. Although the application works effectively on relatively minor motion blurring, such as those less than ten pixels, the general solution, however, simply cannot handle blurs as severe as more specialized programs can.

Another paper, Image Deblurring with Blurred/Noisy Image Pairs, conquers blurred images by taking two photographs. The first has a very low exposure, resulting in a dark, noisy photograph with close to zero blurring. The second is a long exposure photo that gets the color and brightness in the image with lot of motion blurring. The software developed by the team combines the two by performing image deconvolution techniques on the blurred image using the short exposure photo as a reference that reveals how the high exposure image was blurred. With excellent results, Yuan, Sun, and their colleges plan to implement their findings in video cameras.



Figure 2: Deblurred photograph (Raskar, Agrawal and Tumblin).

In order to reverse the blur on an image, it is necessary to approach the task mathematically. If the process that blurs the image is considered a mathematical function, it must be reversed in order to restore the image; however, to do so, it is necessary to understand how the image was blurred, characteristics such as direction, type (motion, out of focus image, etc.), and magnitude. The best way to approach such a complex task is to convert the image into a different domain. The way in which we normally view images is known as the spatial domain, but if the image is converted

into a series of sin functions through a mathematical technique known as a Fourier transformation (Figure 3) it is possible to view the image in the frequency domain (Gonzalez and Wintuz). Once in the frequency domain, it is now possible to perform advanced analysis and mathematic operations on the image in a generalized fashion. It is understood that using the Fourier transformation of an normal image and the Fourier transformation of the blur (a five pixel horizontal line corresponds to a five pixel blur) with a process known broadly as image convolution the Fourier transformation of the blurred image is produced (Figure 4). Thus, by performing the inverse, a deconvolution on the image, the original image can be restored. The most difficult part of this process is determining what the blur factor was when the picture was taken. In theory, if one can determine how the image was blurred, it is possible to deblur the image (Cahill).

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-2\pi i (\frac{xu}{N} + \frac{yv}{M})}$$

Figure 3: Equation for a two-dimensional Discrete Fourier transformation (Bracewell).

3 Rendering the Fourier Transformation

The first step is to render the blurred image in the frequency domain. This is accomplished using a Fourier transformation. The general formula for a Fourier transformation involves integration of a continuous function. Since an image can seldom be represented as a continuous function, it is necessary to treat the image as a set of values in a limited domain. Using the formula for the discrete Fourier transformation (Figure 3) it is possible to render the Fourier transformation of the image. A 2D discrete Fourier transformation requires a calculation with every combination of points on the image, resulting in an extremely slow $O(N^3)$.

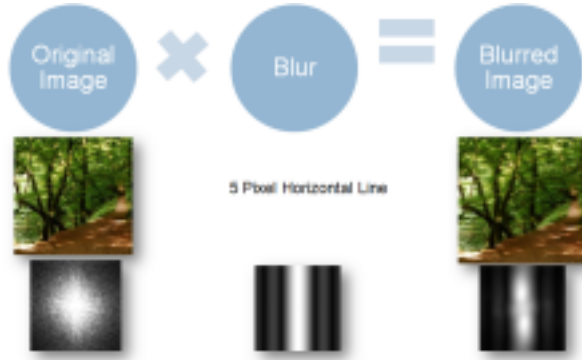


Figure 4: The blurring process with images taken from Cahill.

As a result, it is necessary to use a faster implementation of the Fourier transformation. The fast Fourier transformation (FFT) is a process that allows one-dimensional data sets to be rendered in the frequency domain in $O(N \log N)$ time. Since the sums in the discrete Fourier transformation can be separated, a two-dimensional Fourier transformation can be rendered quickly by applying an FFT to the rows and then to the columns. The speed of the FFT is derived from the symmetric nature of the Fourier transformation, requiring significantly fewer calculations thus decreasing the run-time (Figure 5 and Figure 6). (Jones)

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} \left(x(n) e^{-j \frac{2\pi n k}{N}} \right) \\
 &= \sum_{n=0}^{\frac{N}{2}-1} \left(x(2n) e^{-j \frac{2\pi (2n) k}{N}} \right) + \sum_{n=0}^{\frac{N}{2}-1} \left(x(2n+1) e^{-j \frac{2\pi (2n+1) k}{N}} \right) \\
 &= \sum_{n=0}^{\frac{N}{2}-1} \left(x(2n) e^{-j \frac{2\pi n k}{\frac{N}{2}}} \right) + e^{-j \frac{2\pi k}{N}} \sum_{n=0}^{\frac{N}{2}-1} \left(x(2n+1) e^{-j \frac{2\pi n k}{\frac{N}{2}}} \right) \\
 &= \text{DFT}_{\frac{N}{2}}[x(0), x(2), \dots, x(N-2)] + W_N^k \text{DFT}_{\frac{N}{2}}[x(1), x(3), \dots, x(N-1)]
 \end{aligned}$$

Figure 5: The derivation of the fast Fourier transformation, taken from Jones.

The inverse of the FFT, a step necessary for returning the deblurred image back to the spatial domain, is easily performed by essentially taking the conjugate of the image in the frequency domain, realizing that the data

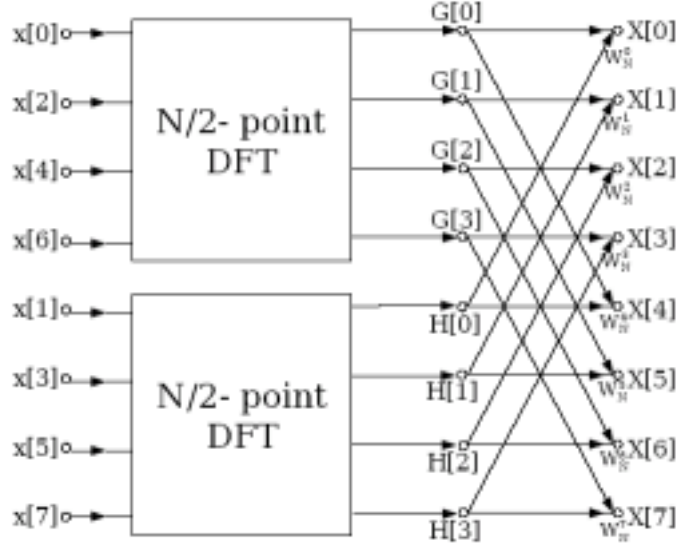


Figure 6: Chart illustrating the increased efficiency provided by the fast Fourier transformation, taken from Jones.

resulting from the FFT is a series of complex numbers; then performing a FFT; and finally calculating the conjugate once again. The product of the entire process results in a significant level of noise for which must be compensated.

References

- [1] Bracewell, Ronald N. The Fourier Transform and Its Applications. New York: McGraw-Hill Book Company, 1986.
- [2] Cahill, M. D. How Automatic Deconvolution Works. 2003. 1 November 2007 [<http://www.hamangia.freemove.co.uk/how/index.html>];.
- [3] Gonzalez, Rafael C. and Paul Wintuz. Digital Image Processing. Reading, Massachusetts: Addison-Wesley Publishing Company, 1987.
- [4] Jones, Douglas L. Decimation-in-time (DIT) Radix-2 FFT. 2006. 21 January 2008 [<http://cnx.org/content/m12016/latest/>];.

- [5] Raskar, Ramesh, Amit Agrawal and Jack Tumblin.
Coded Exposure Photography: Motion Deblurring using Fluttered Shutter.
Cambridge, MA: Mitsubishi Electric Research Labs, 2006.
- [6] Yuan, L., Sun, J., Quan, L., and Shum, H.
Image deblurring with blurred/noisy image pairs. San Diego, CA:
ACM SIGGRAPH, 2007.