Implementation of Image Deblurring Techniques in Java

Peter Chapman
Computer Systems Lab 2007-2008
Thomas Jefferson High School for Science and Technology
Alexandria, Virginia

April 2, 2008

Abstract

Families, friends, professionals, and enthusiasts take countless numbers of photographs every day, and inevitably, many images suffer from some sort of distortion, or "blurring." A program with the power to take a blurred image and create a much crisper and clearer "deblurred" version would be immensely valuable to many fields. Law enforcement agencies, for example, attempting to read the license plate from 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, Inverse Filtering

1 Introduction

Photographs are utilized in many different fields for a wide variety of purposes, and regardless of the subject area, a blurred image is often a useless one. A program with the ability reverse these damages would be extremely useful. Such functionality could be bundled into the software of consumer cameras with adjustments performed automatically after each shot, or perhaps, 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 is only built to handle images suffering simple types of motion blur.

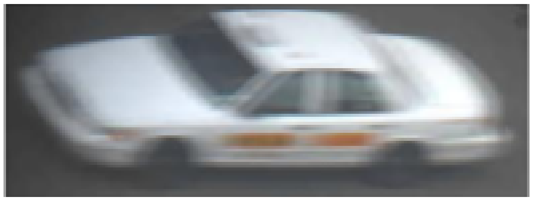

Figure 1: Photograph from a taxi suffering from motion blur (Raskar, Agrawal and Tumblin).

2 Background

Due to the value of a program that can deblur images, many have tried to create an all-purpose deblurring application, but few have found much success with a general approach to the problem. The tendency in the field is to focus on motion blur and narrow the scope 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 photograph, 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 impressive (Figures 1 and 2).

However, it is possible to have some success with general approaches 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 the specialized programs can.

Another paper, Image Deblurring with Blurred/Noisy Image Pairs, conquers blurred images by taking two separate photographs. The first image taken utilizes a very low exposure, resulting in a dark, noisy photograph with close to zero distortion. The second is a long exposure photo that gets the color and brightness in the image with lot of massive amount of blurring. The software developed by the team combines the two images 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 full motion video as the next step in their research.

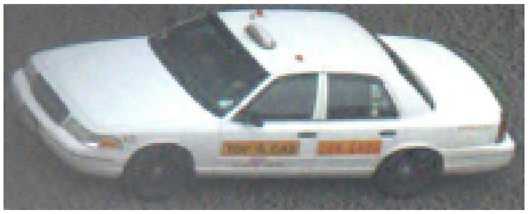

Figure 2: Deblurred photograph (Raskar, Agrawal and Tumblin).

In order to reverse the blur on an image, it is necessary to approach the task from a mathematical standpoint. If the process that blurs the image is considered to be a mathematical function, in order to restore the image the inverse of the function must be applied; however, to do so, it is necessary to understand exactly how the image was blurred: characteristics such as direction, type (motion, out of focus image, etc.), and magnitude are all crucial. The best way to approach such a complex task to is to convert the image into a different domain, since the way in which we normally view images, known as the spatial domain, reveals relatively little about the specifics of a blur without heavy analysis. If the image is converted into a series of sinusoidal functions through a mathematical technique known as a Fourier transformation, (Figure 3) it is possible to view the image in the frequency domain, where the axes are represented by trigonomic functions. (Gonzalez and Wintuz). Once in the frequency domain, it is possible to perform advanced analysis and mathematic operations on the image in a generalized fashion. It is understood that using the Fourier transformation of a normal image and the Fourier transformation of the blur (a five pixel horizontal line in the spatial domain corresponds to a five pixel blur) with a process broadly known 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).

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

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

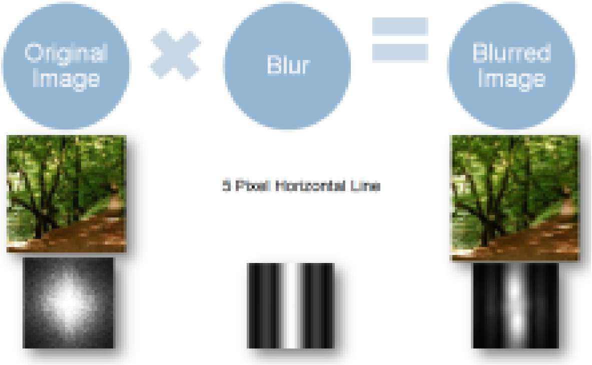

Figure 4: The blurring process with images taken from Cahill.

3 Rendering the Fourier Transformation

The first step in the process is to gain the ability to render the blurred image in the frequency domain. This is accomplished using a Fourier transformation. The general formula for a Fourier transformation involves the 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).

Due to the size of the data set in an image, 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 (NlogN) 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 and Sedgewick)

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

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; the function is easily performed by essentially taking the conjugate of the image in the frequency domain, realizing that the data 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, in an advanced program, is compensated. (Sedgewick and Wayne)

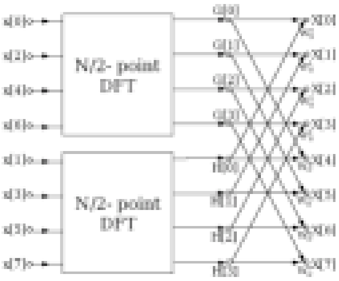

Figure 6: Chart illustrating the increased efficiency provided by the fast Fourier transformation, taken from Jones.

4 Blurring and Deblurring the Image with Inverse Filtering

The ability to take the forward and inverse FFT allow the application to take the image from the spatial domain, preform the appropriate manipulation in the frequency domain, and return the image back to the spatial domain. To preform a blur, both the original image, and the "blur filter" are put in the frequency domain. Then, the corresponding pictures in each image are multiplied by one another, resulting in the blurred image in the frequency domain. The inverse fast Fourier transformation is then preformed resulting in a visibly blurred image from the original. (Image Restoration)

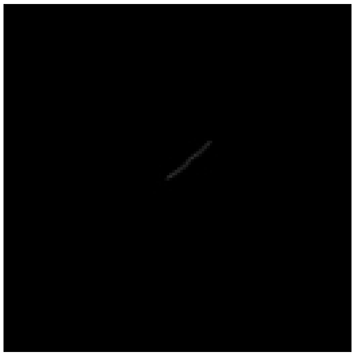

Figure 7: A blur filter created for this paper. This horizontal line should blur the image horizontally.

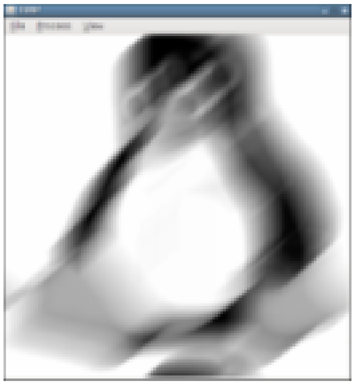

Figure 8: The Linux penguin, Tux, blurred using the above filter.

Now, understanding that a blurred image is the result of the above process, to deblur the image we must multiply by the inverse of the values in the blur filter. However, it is important to not that values close to zero in the filter will result in astronomically high values in the output, so a minimum value must be set. For the tests in this paper that value is 0.2. (Image Restoration)

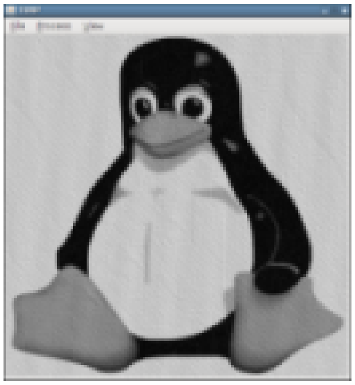

Figure 9: Deblurred photo using results of research.

Due to the scope and the time frame under which this project was created, this research does not attempt to automatically find the filter used to blur the image, a process generally known as automatic deconvolution. I have only attempted blind deconvolution, and for my tests, I am giving the application the same blur filter that I used to blur the original image. Tests of mine have indicated that the filter used to deblur the image must be extremely close to the filter used to blur it. A filter off more than a few pixels will simply destroy the image, showing that having a human input what he or she believes to filter to be simply through a series of educated guesses is not a feasible solution to the problem; blind deconvolution must be used in a practical application.

5 Conclusion

The research here shows that blind deconvolution is simply not an appropriate way of deblurring real-world images. Advanced types of automatic deconvolution must be used, all of which requiring complex high-level math that was simply out of the scope of this project. However, it should be noted that the application that came as a result of this research can be a very useful tool for teaching students the effects and proprieties of Fourier transformations, or as an introduction to the dilemma of image deblurring.

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.freeserve.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] Rice University. Image Restoration. April 2008 ¡http://www.owlnet.rice.edu/ elec539/Projects99/BACH/proj2/intro.html/¿;.

[7] Sedgewick, Robert and Wayne, Kevin. Introduction to Programming in Java: An Interdisciplinary Approach. 2007. April 2008 ¡http://www.cs.princeton.edu/introcs/97data/index.php/¿;.

[8] Yuan, L., Sun, J., Quan, L., and Shum, H. Image deblurring with blurred/noisy image pairs. San Diego, CA: ACM SIGGRAPH, 2007.