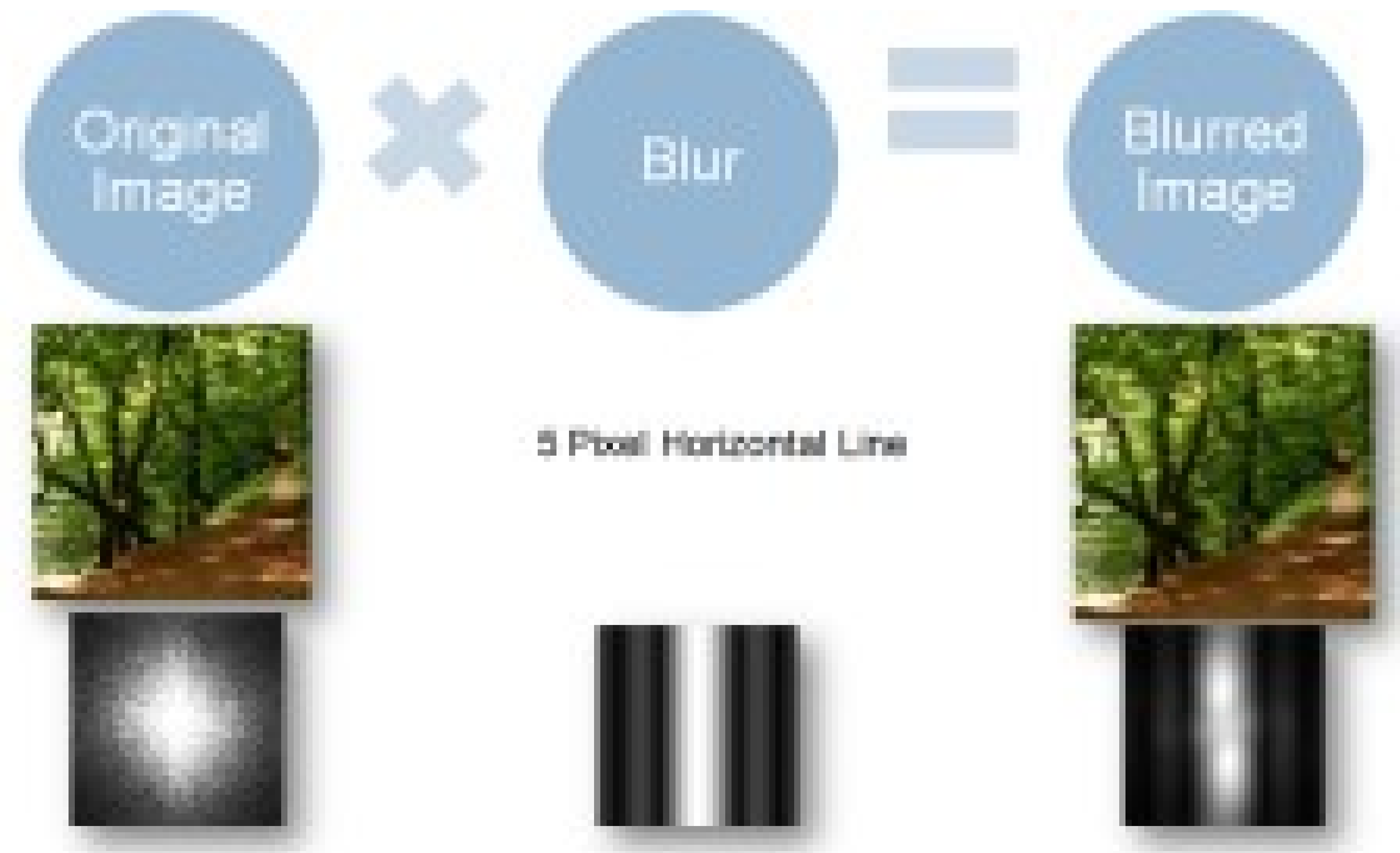
Implementation of Image Deblurring Techniques in Java

Peter Chapman

Abstract

A countless number of photographs are taken every day, and inevitably many of those suffer from some sort of blurring. A program that would be able to take a blurred image and restore the photograph back to its original, deblurred form would be invaluable. Anyone from law enforcement trying to read a blurred license plate on a getaway car to a family attempting to refine 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.

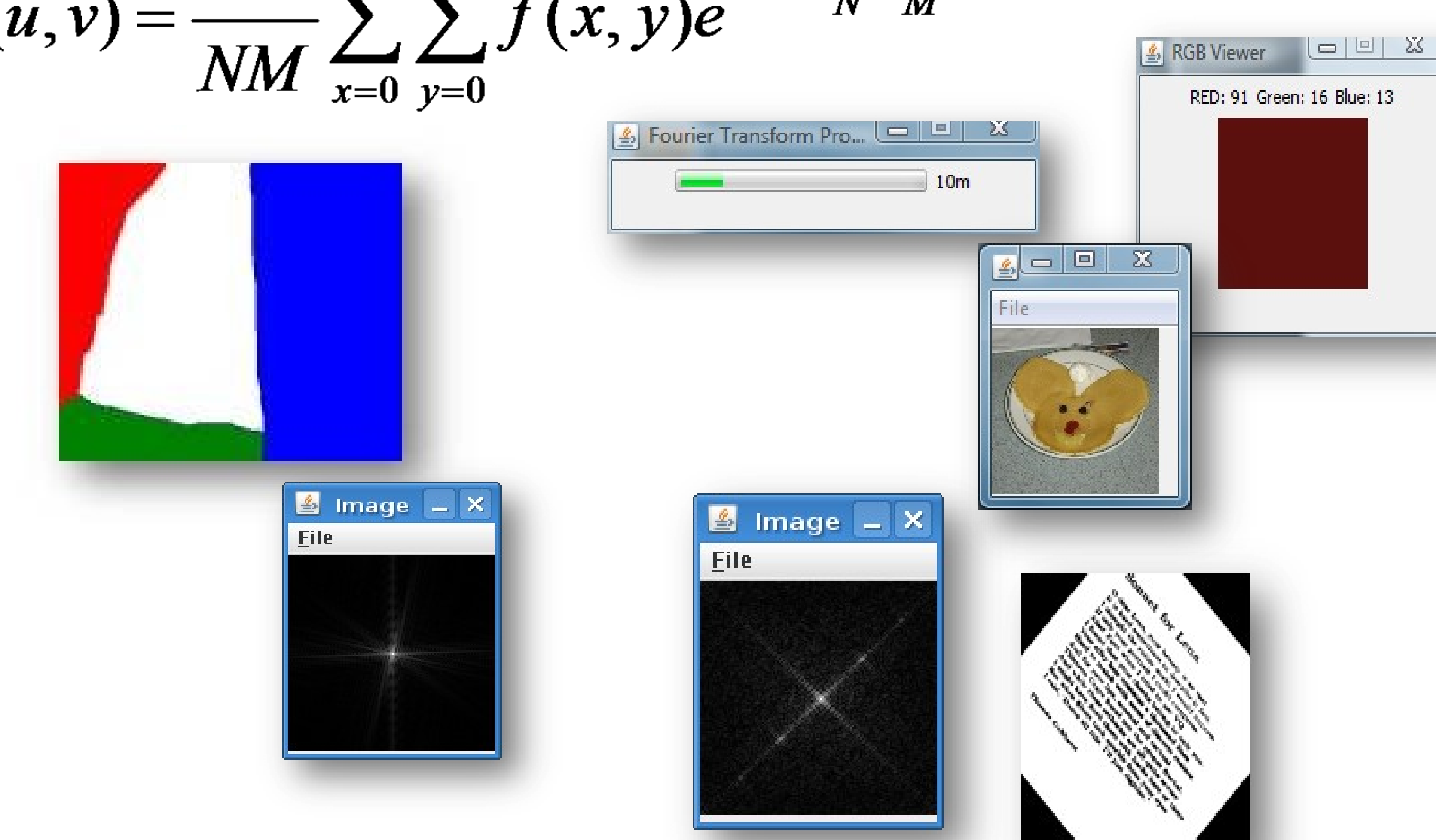


Background

In order to reverse the blurring process it is necessary to consider the subject area in a mathematical context. Therefore, if the blurring process is considered a mathematical function, it must be reversed; to do so, it is necessary to understand how the image was blurred: direction, type (motion, out of focus image, etc.), magnitude. In order to accomplish this, it is necessary to visualize the image in a different domain. Normally we only view images in the spatial domain, but if the image is converted into a series of sin functions through a mathematical technique known as a Fourier transformation it is possible to view the image in the frequency domain . Once in the frequency domain, it is possible to analyze the image in a more advanced and perform operations in a more generalized way. It is understood that the Fourier transformation of the original image multiplied by the Fourier transformation of the blur (a five pixel horizontal line corresponds to a five pixel blur) produces the Fourier transformation of the blurred image (Figure 4). Thus, the blurred image divided by the blur factor should produce the deblurred image . 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 blur the image. This entire process is known as deconvolution .

Fourier Transformation

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



Procedure

Rendering Fourier Transformations

The first step in the deconvolution process is to render the blurred image in the frequency domain using the Fourier transformation. The general formula for a Fourier transformation requires 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 render the Fourier transformation of the image. A 2D discrete Fourier transformation requires every single point to perform a calculation on every other point, resulting in an extremely slow O(N^3). 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(NlogN) time. Since the sum in the discrete Fourier transformation can be separated, a two dimensional Fourier transformation can be a rendered quickly by applying an FFT to the rows and then to the columns.

Conclusions

Rendering Fourier Transformations

The first step in the deconvolution process is to render the blurred image in the frequency domain using the Fourier transformation. The general formula for a Fourier transformation requires 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 render the Fourier transformation of the image. A 2D discrete Fourier transformation requires every single point to perform a calculation on every other point, resulting in an extremely slow O(N^3). 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(NlogN) time. Since the sum in the discrete Fourier transformation can be separated, a two dimensional Fourier transformation can be a rendered quickly by applying an FFT to the rows and then to the columns.