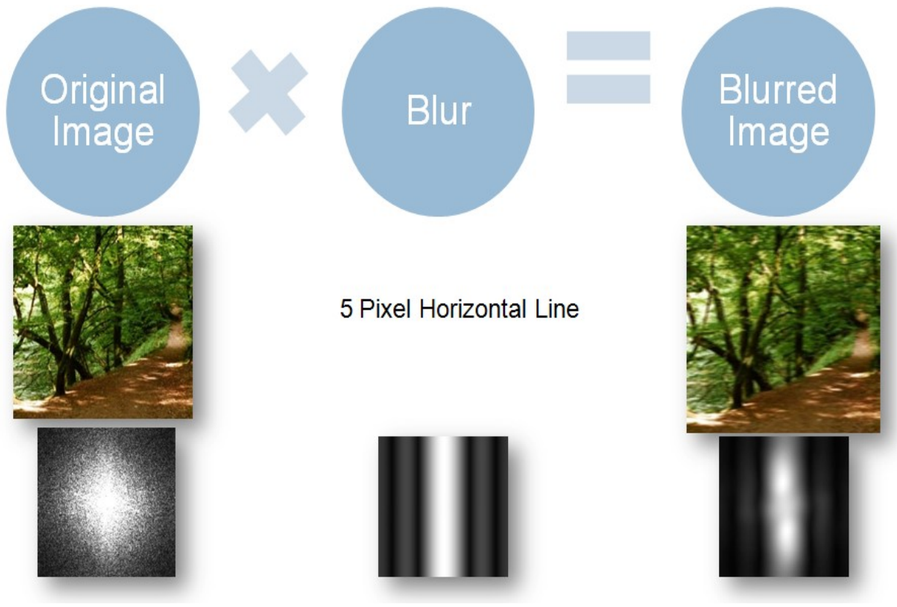
Implementation of Image Deblurring Techniques in Java

Peter Chapman

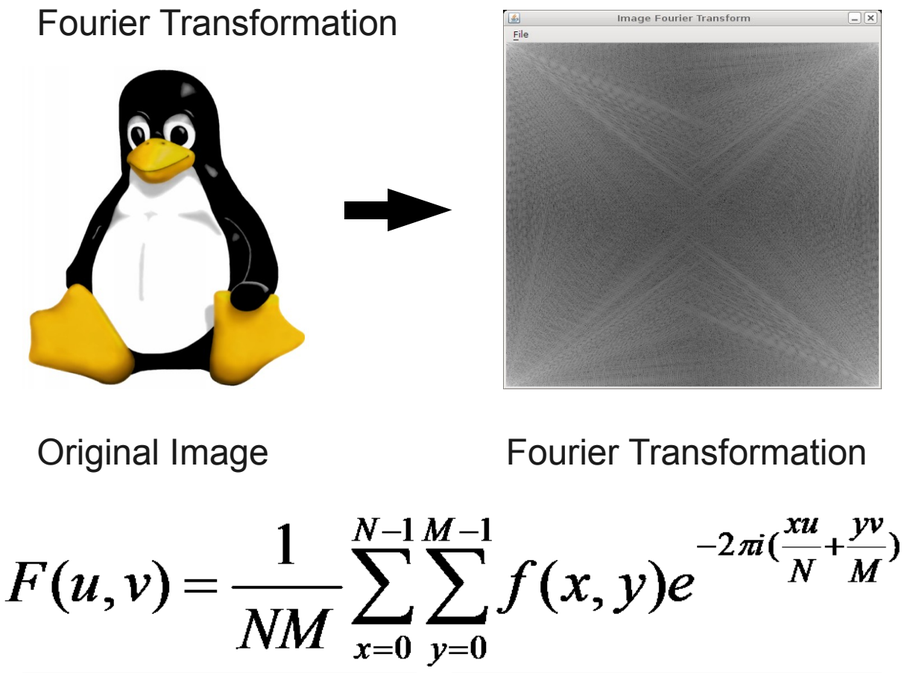
Abstract

Countless numbers of photographs are taken every day, and inevitably, many images suffer from some sort of "blurring." A program with the power to take a blurred image create a much crisper and clearer "deblurred" form would be immensely valuable. Law enforcement making out a blurred photo of the getaway car's license plate, or even 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.



Background

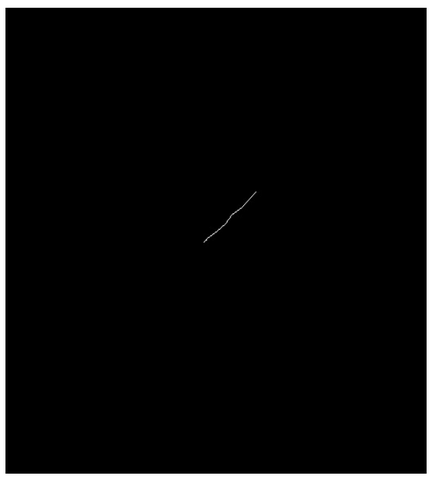
In order to reverse the blur on an image, it is necessary to approach the task mathematically. If the process that blurs the image is considered a mathematical function, it must be reversed in order to restore the image; however, to do so, it is necessary to understand how the image was blurred, characteristics such as direction, type (motion, out of focus image, etc.), and magnitude. The best way to approach such a complex task to 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 it is possible to view the image in the frequency domain. Once in the frequency domain, it is now possible to perform more advanced analysis on the image and perform mathematical operations in a more generalized way. It is understood that using the Fourier transformation of an unblurred 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 produces the Fourier transformation of the blurred image. Thus, by performing the inverse, a deconvolution on the image, the unblurred 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 unblur the image.



Procedure

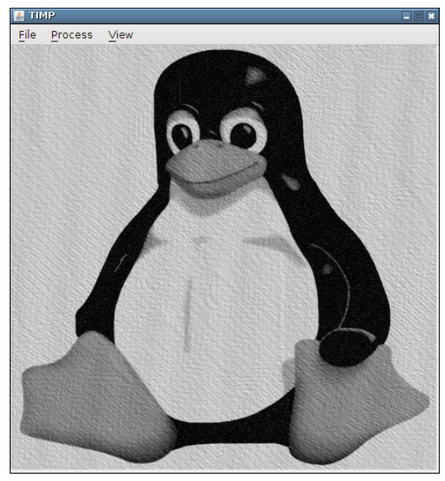
Rendering Fourier Transformations
The first step is to render the blurred image in the frequency domain using the Fourier transformation. The general formula for a Fourier transformation requires a continuous function. Since an image can seldom be represented as a continuous function, it is necessary to treat the image as a set of values in a limited domain. Using the formula for the discrete Fourier transformation 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 sums 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. The speed of the FFT is derived from the symmetric nature of the Fourier transformation, allowing much fewer calculations to be made on the data thus decreasing the run-time.

The inverse of the FFT, a step necessary for returning the deblurred image back to spatial domain, 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 an FFT; and finally calculating the conjugate once again. The result of the entire process results in a significance amount of noise for which must be compensated.



Blur Filter

Blurred Image



Deblurred Image

Blurring and Deblurring

Although time consuming, properly rendering the Fourier transformations work at the heart of the blurring and deblurring process, regardless of the method used. As a starting point, I decided to utilize the simplest image deblurring method known as inverse filtering. As discussed earlier, blurring, at its most basic level, is the multiplication of the Fourier representation of the image and the Fourier representation of some other filter. By dividing, while ensuring that one does not divide by zero. The result, shown above is fairly accurate, with the exception of a series of wave-like lines filling the image. This method must have a deblur filter that is nearly exactly the same as the filter used to blur the image. More advanced methods of deblurring calculate, or more accurately guess, the original blur filter; my application is not as advanced, but works well enough to experiment with the effect of certain filters on certain images. The next step will be to attempt to preform more advanced techniques of image deblurring.