Image Deblurring
TJHSST Senior Research Project Proposal
Computer Systems Lab 2009-2010

Vincent DeVito

October 30, 2009

Abstract

In the world of photography and machine vision, blurry images can spell disaster. They can ruin an otherwise perfect photo or make it impossible for a computer to recognize the image or certain components of it for processing. The best way to counter this without taking another, clearer picture is to utilize deconvolution techniques to remove as much blur as possible. That is the design of this project. My plan is to first design a program that takes an image, blurs it using a known blur kernel, then deblurs it to reproduce the original image. After that I will attempt to create a program to determine the blur kernel of a naturally blurred image. I will use Python and a package called Python Imaging Library, and my success will be measured simply by how much the output (deblurred) image matches the input (original) image.

Keywords: deblurring, noise reduction, deconvolution, image processing

1 Introduction and Background

The goal of this project is to create a program that can take an image input that has been blurred (first artificially, and later hopefully by poor image capture) and to employ image deblurring techniques to restore the image and create a sharp, more recognizable output image with as few blur artifacts as possible.

1.1 Previous Research

So far I have found a paper regarding image deblurring and noise suppression called "Image Deblurring with Blurred/Noisy Image Pairs" by Lu Yuan, et al.[2] that I plan to utilize in helping me understand the techniques and algorithms that go into reducing the noise of and deblurring an image. In their research they used a blurry image with proper intensity and poor sharpness and paired it with an identical picture with good sharpness but poor intensity and riddled with noise to create a sharp, correct intensity output with few or no artifacts left in the output image.

Another paper[1] I have read discusses an algorithm that the group of researchers discovered that allows for a mostly accurate estimation of the blur kernel, or the path/direction of the blur from the original image. Their algorithm takes four inputs: the blurry image, a section of the image that has a good sample of blurring (in case the image is not uniformly blurred), if the blur is estimated to be more horizontal or more vertical, and the estimated size of the blue kernel. Given these inputs, their algorithm can sufficiently estimate the blur kernel such that, after deconvolution, the image, which was captured using poor technique with a standard, off-the-shelf camera, is satisfactorily deblurred with few artifacts. What artifacts that are left can generally be removed by an experienced photo editor.

1.2 Other Research

Through my own work I am accruing a detailed understanding of basic and intermediate image processing techniques and algorithms from various online worksheets and lessons at http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm. I plan to use these techniques to help me code and understand the more complex concepts behind image deblurring and the intermediate steps involved.

2 Procedure and Methodology

I plan to use the programming language Python to write the code for this project. I have decided to use Python because of its simplicity and adaptability. I have also found, online, a package called the Python Imaging Library (PIL) that will allow me to read in and process almost any image format using Python with relative ease. This will enable me to use more compressed image formats, such as .jpg and .gif, instead of the larger uncompressed formats such as .ppm and .pgm as well as letting me process them much more efficiently.

First I will design my deconvolution algorithm that, when given an image and its known blur kernel, can deblur the input image. This will likely be based on previously discovered, more efficient algorithms. At that point I can attempt to add a noise reduction filter at the end to remove any excess noise in the image and further sharpen and clarify the image. If I have time left after completing these tasks I will begin to design a program that can obtain an estimate of the blur kernel from first an artificially blurred photo, then a naturally blurred one. This next step is a large one and currently under a lot of research since blind deconvolution (or deconvolution without a known blur kernel) is quite difficult.

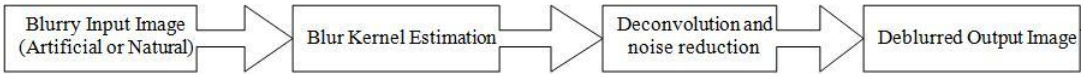

Figure 1. A graphic representation of my program design.

My project's success will be measured by its ability to take an image (which will most likely be artificially blurred) and return it to its original, sharp quality. I will test my projects ability by running a series of tests. These tests will entail attempting to deblur images of different content, type, and contrast and will be artificially blurred to different levels. This will test the adaptability of my program to repair images regardless of content or level of blur distortion (although there will obviously be an upper limit to the amount of blur that can be plausibly be removed).

Figure 2. This is an example of a blurry image input, with a particularly blurry section highlighted.[1]

Figure 3. This is the same section from figure 2, but with the blur drastically reduced, due to deconvolution algorithms.[1]

3 Expected Results

From this project I expect to be able to obtain a completely, or almost completely, deblurred image from an image that has a moderate level of blur distortion. I plan to be able to accomplish this within a few months after completing my image processing introductory work, and after I have learned a sufficient amount about image deblurring such that I can begin to program. I hope that my program can perform these processes with a relatively high level of success, though I think that perfect deblurring of the image will probably be impossible, at least not without a higher level of understanding of complex concepts that I do not think I will be able to touch on within my limited time frame.

For this project, visual representations will be very easy to acquire since the inputs and outputs themselves are images. To present the effectiveness or outcome of my project I can just compare the input and output images and show the changes, or lack thereof, that my program created.

3.1 Scope

The scope of this project is rather narrow but important. It pertains only to blurry images, but this is a rather large problem in the worlds of image processing, photography, and machine vision. In photography, blurry images are undesirable because they lack sharpness or clarity and in machine vision, blurriness can make an image indecipherable by the computer or render certain processes ineffective (such as edge detection).

References

[1] Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., & Freeman, W. T. (2006, July). Removing camera shake from a single photograph. *ACM Transactions on Graphics, 25*(3), 787-794. doi:10.1145/1141911.1141956

[2] Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (2003). *Image Processing Operator Worksheets* [Image processing worksheets]. Retrieved from http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm

[3] Yuan, L., Sun, J., Quan, L., & Shum, H.-Y. (2007, July). Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics, 26*(3). doi:10.1145/1276377.1276379