CPSC 535
Assignment 2b: Inverse Filtering for Image Restoration

The goal of this assignment is to implement and test inverse filters to restore images that have been blurred by motion. A classic application of this technique is to read the license plate from a photograph of a car taken while the car is moving.

Section 3.3 of Szeliski covers inverse Wiener filters.

1 Motion Blurring

Whenever you take a picture with either a conventional film camera or an electronic one, there is always a finite image integration time, i.e., the time during which the shutter of the camera is open so that the imaging device can collect photons to measure image intensity. This is not a problem when everything you see is stationary, but when things move, the integration processes blurs the image.

The usual solution in photography is to use a faster film and shorter integration times. There is a limit to this approach ... as the integration time gets shorter less light reaches the imaging device and the image quality drops. Also, a good photographer will also *track* his subject as it moves, thus keeping the image of the subject sharp and blurring the background.

When faced with an image that has motion blur, and no option to retake the picture, we can *reverse* the motion blur digitally. Step one is to create a model that describes the blurring process.

To get a simplified model of motion blurring, suppose that we have a static scene, but that the camera is tracking from left to right, thus blurring the image. We divide the image integration time into small time intervals during which the scene is displaced by exactly one pixel. If in our contrived example the camera moves so that the image moves 10 pixels, then we imagine that the integration consists of averaging 10 sharp images, each displaced 10 pixel to the right from the last. We can describe the averaging as a convolution with the following filter kernel:

$$h = \frac{1}{10}\left[\begin{array}{cccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Greater blurring, i.e., more motion, results in a wider kernel while less blurring results in a narrow kernel. If the motion is in a different direction, the, we rotate H so that it matches the direction of the motion.

Thus we can model motion blurring as a convolution.

2 Deconvolution

If blurring is convolution, then unblurring is *deconvolution*, and if convolution equals multiplication in the Fourier domain, then deconvolution equals division in the Fourier domain. So in a nutshell, you can undo motion blurring, by dividing the Fourier transform if the image by the Fourier transform of the motion blur kernel, i.e.,

$$\begin{aligned} I(u,v) &= i(x,y) \\ H(u,v) &= Fh(x,y) \\ \hat{I}(u,v) &= \frac{I(u,v)}{H(u,c)} \\ \hat{i}(x,y) &= F^{-1}\hat{I}(u,v) \end{aligned}$$

Figure 1: Test image showing motion blur: an old license plate from California.

where i is the blurred image, h is the blurring kernel, $\hat{i}$ is the estimated de-blurred image, F and F^{-1} indicate Fourier and inverse Fourier transforms, and upper case denote Fourier domain functions.

In practice things are not so simple because if any values in $H(u,v)$ are zero then $\hat{I}$ is undefined. Two approaches to deal with this are

1. Band-limit the signal so that you use only the range of frequencies between zero and the lowest-frequency zero value, or
2. Wiener deconvolution.

Wiener's method is described in the text.

3 The Assignment

You should first take a known image (make one up if you want) and blur it with a known blurring kernel. Then try to deconvolve that image back to something that is not blurred using both deconvolution methods.

Next apply the deconvolution methods to the test image (see Figure 1) provided on the course web page.

4 Hand in

Hand in the following:

1. your code for the restoration of the sample image, and the two blurred images that go along with the assignment.
2. images to compare the original blurred image along with deblurred images produced using the two deconvolution methods above.
3. an explanation of any artifacts or unusual features that you discovered as you did the restoration.

You will be graded on the quality of your code, the deblurred images, and your written observations.