E E 2 9 5 March 13, 2008

LAB 5 - Deconvolution Using Wiener Filtering

Lab Report Due: - March 19, 2008.

In this Lab we blur (convolve) an image with a deblurring filter and then deconvolve (deblur) the image using the Wiener filter. (Note that we have used the Wiener filter before in Lab. 4. We will cover inverse filtering (deconvolution) and Wiener filtering in class).

One of the fundamental problems in statistical signal processing is the deconvolution problem in additive noise. That is, we have

$$y = x * h + n$$

where x, y, n are the transmitted, received and addiitive noise signals respectively. The objective is to estimate from y the transmitted signal x when we may or may not know much about x or n.

There are many techniques for deconvolution and for deblurring. In this Lab., you will need only the adaptive Wiener filter deconvolution algorithm *deconvwnr*. This experiment is from the MATLAB for Image processing notes, pp.6-13 - 6-15. The original notes are in the Lab. and you should look at them.

Note the *PSF* and *OTF* terminology often used in non-signal processing literature. The MATLAB function J = deconvwnr(I,PSF) restores an image *I* that has been distorted by convolution with a filter and possibly also by additive noise. The deconvolution algorithm is optimal in a least-mean-square sense between the estimated and the true image. In the process of minimization of the *mse*, the solution requires, as we shall see later, the determination of the image and noise correlation matrices. In the absence of noise, the Wiener filter reduces to the ideal inverse filter.

In this Lab. we do experiments in Chapter 6, pages 6-13 to 6-15. However, all that you really need is the function *deblur_ex.m*.

Do the following:

(a) In MATLAB, do *file* → *open deblur_ex.m*.

(b) In the Editor, read the program and see that you are deblurring the image $N = 25$ pixels at an angle of 45°.

(c) In the Editor, do *Debug* → *run*.

(d) After a few seconds, cross hairs will appear. Within any region in the image, mark of 2 points, at an approximate angle of 45° counter-clockwise and about 25 pixels apart.

(e) Watch the top of the image and you see the pixel count over which the program is iterating. What you are seeing is the deblurring over the range $-3 \leq N \leq 3$ pixels. There is no attempt here to choose the optimal N that gives minimum distortion. (Optimization algorithms can be written to do that). This example is merely an illustration of the general process. As N approaches 25, you see an image closer to the original.

(f) After a few seconds, you are asked to put in the actual number of pixels used in the blurring. After that you see the reconstruction with this number.

(g) Keep repeating the experiment until you get a good sense of what is happening.

(h) Edit your program to add noise (B=imnoise(A,'gaussian')) to the image. Repeat the experiment.

REPORT

1. Write a clear, concise and meaningful description of this experiment as described in *deblur_ex.m*.

References: (1) MATLAB for Image Processing ©2007, The MathWorks, Inc. Image Restoration Techniques Ch.6. Note that we only have *one* copy in the Lab.