To: Prof. Ross Snider

From: Jeff Meirhofer

Regarding: Lab #8 – Filtering and Edge Detection of Images

Date: March 12, 2009

**Summary:**

The purpose of this lab is explore the concept of filtering digital images. These images can be considered a type of digital signal, and therefore can be treated as such. We will experiment with implementing highpass and lowpass filters on some images, and then explore the problem of edge detection. For edge detection, we will need to use linear and nonlinear filters.

**Implementing Filters on Digital Images**

**3.1 a)** First, we will load an image that has been provided titled “echart.mat”. We will look at the signal contained in the 33rd row of this image. We can run this signal through a 7-point averager and examine the output. The original signal and filtered signal are shown below.



Figure 1. Input and output of a 7-point averager.

We can see that the filter makes the waveform “smoother”. We know that these are the characteristics of a lowpass filter.

**3.1 b)** Now we will look at the 99th column of the image, and filter it with a first-difference. The same comparison as before is made on the next page.



Figure 2. Input and output of a first-difference filter

We can see that this system makes the output rougher than the input. We know that this type of filter is a highpass.

**3.2 a)** In order to filter the image completely, we must use the “conv” function for each row and each column. This can be very time-consuming. Luckily, MATLAB has a “conv2” function that will speed this process up. We will use it to filter all of the rows at once. When the coefficients for bh are in a row vector, conv2 filters the rows. When the coefficients for bh are in a column vector, conv2 filters the columns of the image. Now we will implement the 7-point averager with the conv2 function and compare the result to the original image.



Figure 3. Input and output of 7-point average.

We can see that the output image is blurred in the x-direction. When we extract row 33 from this image and compare it to the output from part (a), we can see that the two are almost exactly the same.



Figure 4. Comparison of row 33.

**3.2 b)** Now we will filter the output image again, but this time in the y-direction. When we compare all three images (Figure 5), we can see the last output is an image smoothed in both directions. This makes the image looked blurred. It is similar to what we did to the speech signal when we ran it through a lowpass filter.



Figure 5. Original, x-direction filtered, and both directions filtered signal

**3.2 c)** If we repeat this experiment with a 21-point averager, we expect the output to be even more blurred that the 7-point. In the next figure, you can see that that clearly is the case.



Figure 6. Outputs from a 21-point averager.

**3.3)** The next part of the lab involves filtering the “lenna” image with five different filters, and observing the results. The filters’ coefficients are as follows:

i) [1 1 1]/3

ii) [1 1 1 1 1 1 1]/7

iii) [1 -1] (on this filter, we will filter only the rows)

iv) [-1 3 -1] (on this filter, we will filter only the rows)

v) [-1 1 1 1 -1]

The results of the filtering is shown below.



Figure 7. Five different filters.

The first image is the original. The filters that act in a lowpass manner are a2 and a1, while a3, a4, and a5 appear to be highpass filters. Lowpass filters tend to make the image more blurry.

**3.4 a)** Now we will try to do some sharpening of an image. First, we load the baboon (baboon.mat). Now, we will create a filter that has Gaussian shaped coefficients. A stem plot of the impulse response is shown, along with the magnitude of the frequency response. We can see that it is a lowpass filter.



Figure 8. Response of lowpass filter.

**3.4 b)** Next, we create a filter whose coefficients are a Gaussian multiplied by a cosine. A plot of its coefficients and its frequency response is shown on the next page. We can see that it is a bandpass filter because its response is attenuated for high frequencies and low frequencies.



Figure 9. Response of bandpass filter.

**3.4 c)** When the test image is filtered with the lowpass filter, the resulting image is very blurred. Because the resulting has been lowpass-filtered and is very blurry, we can conclude that y[m,n] contains mostly low frequency content.



Figure 10. Original image. Figure 11. Lowpass-filtered image.

**3.4 d)** Now we will filter the image with the bandpass filter. The results are shown.



Figure 12. Bandpass-filtered image.

We can see that the image does not contain high-frequency or low-frequency content. It only contains frequencies in between high and low.

**3.4 e)** Now we will sum the lowpass-filtered image with the bandpass-filtered image. We will use the equation to produce the output. This uses alpha to control how much of the image will come from each filter. We found that the best value of alpha is about 0.95. The resulting image looks sharper than the blurry image because some high-frequency content has been added to it from the bandpass filter.



Figure 13. Sharpened image.

**3.5)** We will try to do some de-blurring using the method of synthetic highs.

**3.5 a)** First we must construct a highpass filter with coefficients 1, -2, and 1. The frequency response is shown below.



Figure 14. Frequency response of highpass filter.

**3.5 b)** Now we must compute the synthetic high image. This is done by passing the blurred image through the highpass filter.

**3.5 c)** The frequency content of the synthetic high image (w[m,n]) consists of the highest frequencies in the blurred image. The content is at lower frequency than the content from the bandpass filter.



Figure 15. Frequency response of cascaded system.

**3.5 d)** Finally, we will sharpen the image with the same summing process as before. We will sum the results of the blurred image and the highpass filter.

**3.5 e)** Below you can see a comparison of the blurry image and the sharpened image. We found that the best value of alpha was about



**Appendix**

**3**