Vanderbilt University
School of Engineering

EECE 253 IMAGE PROCESSING

LABORATORY ASSIGNMENT 6

Jack Minardi
28 Nov, 2011

Abstract

This paper reports the results of experiments done to explore filtering for the point of noise reduction and edge enhancement. The first experiment compared the wiener filter to simple blurring, and the wiener filter did much better. The blurring was done just to have a baseline. The second experiment was done to demonstrate how to remove periodic noise from an image. This was accomplished by zeroing out the noise peaks in the frequency domain, thus eliminating the noise from the image. The final experiment demonstrated how to remove halfltone noise. This noise presents itself as a dot with a ring of dots around it. To get rid of it three filters were created, a spatial domain filter, an unsharp mask, and a frequency domain filter. All three demonstrated good results.

Table of Contents

Introduction

Using Fourier transforms to manipulate images give us great insight into how frequency space works and what it represents. The goals of this laboratory are to learn about the Fourier transform of an impulse, how to perform convolutions on both the spatial and frequency domain, and how to filter images via convolution. It is found that convolution in the frequency domain is much faster for all but the smallest of weight matrices. Filtering in the frequency domain can also produce some interesting results, like high and low pas filters which can be used to sharpen or blur an image.

Description of Experiments

There were five experiments performed in this lab. Matlab was used for all the computation. All the scripts that were written can be found in the appendix and in the zip file this report came in. The results were somewhat subjective and discussions can be found in the report below.

Experiment 1.

Reduce the noise in the supplied image, Gilles Tran - Glasses - POV-Ray - Noise.jpg and compare the result to the original image, Gilles Tran - Glasses - POV-Ray.jpg.

(a) First blur the noisy image using at least three different blurring operators of your choice. For each result, compute the per pixel energy in the difference between the noise-reduced image and the original image and include them in your report. Select from the original image, three or more small windows1 that contain different types of features. Cut these out and save them for your report. cut out the same areas from the blurred images and save them. In your report display the original and blurred regions side-by-side so that the differences are apparent. Comment on those differences.

(b) Experiment with the adaptive Wiener filter from the Matlab image processing toolkit (wiener2) to find a better result than the simple blurring in the previous part. That is, generate images with wiener2 using different neighborhood sizes. Determine the quality of the result using the per pixel energy in the difference between the noise-reduced image and the original image and display the same windows as in part (a). I suggest that in this section you format your report so that one image region (window) is displayed on a single page. That is, for each window location create a table that displays the contents of the window in the original and each of the noise reduced versions so that they can be viewed together. You will then have one page per window each of which shows the effects of the different noise reduction techniques on a single region. An example is shown at the end of this document.

Experiment 2.

Design a frequency-domain mask to remove the periodic noise from the image supplied: TokyoHighwaysPdNoise.bmp. Compute the power spectrum of the image and locate the frequency components that are responsible for the noise. From those components determine the number of sinusoids that are corrupting the image. Derive the wavelengths in pixels and the orientations (with respect to the column axis) of the sinusoids. Create a frequency mask to eliminate the noise. Use the mask and display the resulting image In your report, list the row and column indices of the frequency components you zeroed, display the original image, the filter mask, and the image that results from filtering the original.

Experiment 3.

The image, Saoirse Ronan - Ember - 968x648 - HTD.bmp was given clearly visible halftone distortion (HTD). The distortion appears as small circles, each with a dot at its center. These are called rosettes and result from the printing process. For comparison, the original image is Saoirse Ronan - Ember - 968x648.jpg

(a) Design a spatial domain filter – a convolution mask, h1, – to minimize the halftone distortion while simultaneously minimizing the blurring of larger features. [Which type of filter do you know to be optimal in terms space-frequency localization?] Process the original HTD image, I, with that filter to produce a blurred image, J. I recommend that you compute the convolution in the frequency domain, using the program that you wrote for a previous lab assignment. Include the original HTD image, I and the result, J, in your report. Describe the filter and the parameters you used (e.g. spatial support dimensions, sigma, etc.) Explain your reasoning for the choice of parameters. Hint: I suggest using and annotating a small but greatly enlarged section from I to illustrate your parameter choices. Take the same section from J to illustrate the results. It would be useful to mark the same pixel in both images for comparison.

(b) Use unsharp masking to sharpen the small features in the image, J, that you blurred. To do unsharp masking, you must blur J (i.e., blur I a second time) to produce a more highly blurred image, B. The degree of blurring is a parameter that you must choose; it should be sufficient to blur the finest features in J without obliterating larger ones. Then you subtract B from J to create a difference image D. The unsharp masked image K is defined by $K = (1 + \alpha)D + B$ (1) where you must choose a good value for parameter $\alpha \geq 0$. Present the unsharp masked image in your report and justify your choice of the blurring parameter. [Compare it to the blurring parameter you used in the first part of the problem.] Choose a value for α that makes the result sharper in appearance than J but that does not significantly distort the result. By visual inspection determine what happens to the result, K, as α is increased from zero. I suggest using several values of α varying from, say 0.1 to 1.5. How does the appearance of K change as α gets larger? A comparison of the same region from K as you used in the first part could aid your description of the results. But, if that is too small to appreciate the results, compare larger regions of I, J, and K. 2

(c) Design and use a frequency-domain filter to remove the halftone noise. The noise is visible in the power spectrum as sets of approximately red, green, blue, and white, '+' features. Note that some of the lower frequency components of the halftone dots are barely visible at the edge of the central disk in the spectrum. Your mask should zero the '+'s to remove the distortion. I suggest using 'fat' dots – zero regions with a radius of 2 to 5. Be sure to blur the mask so that the notches in it have smooth, rather than, abrupt edges. How successful was blur-then-unsharp-mask approach to the removal of halftone distortion? Did the USM significantly improve the blurring the edges?. Compare those results to the frequency filter method. Which was superior? What are the similarities and differences in the appearances of the results? Recap of the USM part: You will load image Saoirse Ronan - Ember - 968x648 - HTD.bmp into a matrix, I. Then blur I by a filter, h1, sufficiently large to remove the half-tone dot distortion; call the result J. To perform unsharp masking, you will blur J by a second, larger filter h2 to create an image, B. Then create image D = J − B, the difference between the once-blurred image and the twice-blurred image. The sharpened image is computed using equation (1) with an appropriate choice of α. You will get better results if you scale image J to lie in the range [0, . . . , 255] before you blur it with the second filter (providing your filters each sum to 1). Then clip K at 0 and 255 rather than scaling it. [Note: Do not use the intrinsic filter fspecial('unsharp',0.5) to do this. It is a different operation than the one I want you to do.]

Results of Experiments

Experiment 1

The experiment dealt with removing noise by simple blurring. The following pages detail small sections of the image with different filter parameters. The blurred image cleans up the noise the best when it is set to 3x3 and gets worse as sigma grows. There is not much visual difference between the three wiener filter settings of 3x3, 5x5, and 7x7 but if you look at the energy difference per pixel you can see that the 3x3 is closest to the original image. The 3x3 is also the best out of every filter, and is the only one that is a better approximation of the original than the unaltered noise.

The per pixel energy difference was calculated with the following command:

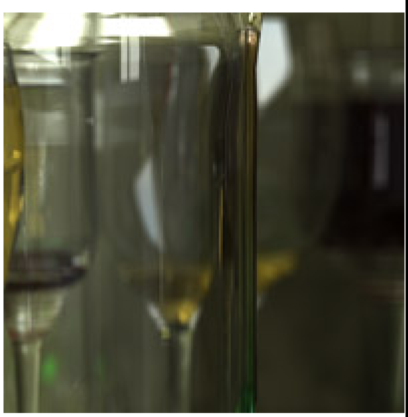
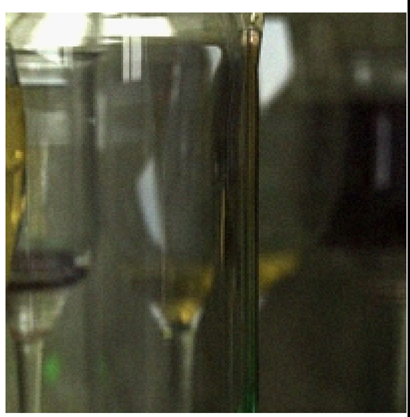
```
>> sum(abs(orig(:)-I(:)))/prod(size(I))
```

per pixel energy difference
noisy = 1.4897
3x3 blur = 4.5901
5x5 blur = 5.9860
7x7 blur = 6.4829
3x3 wiener = 1.3437
5x5 wiener = 1.7758
7x7 wiener = 2.2345

		
Original Image	Noisy Image	

		
3x3 Blur	5x5 Blur	7x7 Blur
		
3x3 Wiener	5x5 Wiener	7x7 Wiener

Figure 1-8 Showing results of different filters (Window 1)

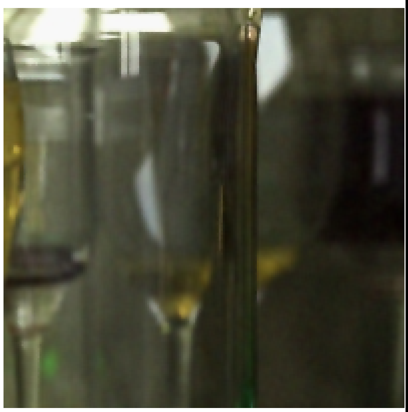
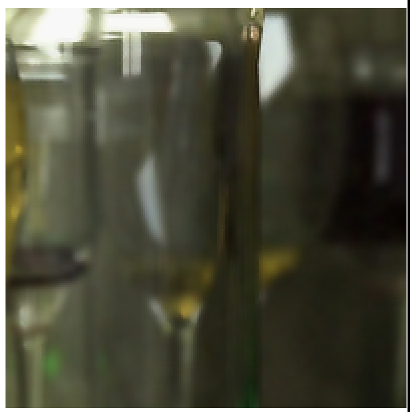
Original Image	Noisy Image	
		
3x3 Blur	5x5 Blur	7x7 Blur
		
3x3 Wiener	5x5 Wiener	7x7 Wiener

Figure 9-16 Showing results of different filters (Window 2)

		
Original Image	Noisy Image	
		
3x3 Blur	5x5 Blur	7x7 Blur
		
3x3 Wiener	5x5 Wiener	7x7 Wiener

Figure 17-24 Showing results of different filters (Window 3)

Experiment 2

This experiment involved zeroing out periodic noise. The supplied image had periodic noise added to it, and is shown below. The periodic noise shows up as the overlaid grid pattern.

Figure 25. Image with periodic noise added.

To remove the periodic noise the power spectrum was displayed to identify precisely what frequency the noise was so it could be zeroed out. The power spectrum is shown below.

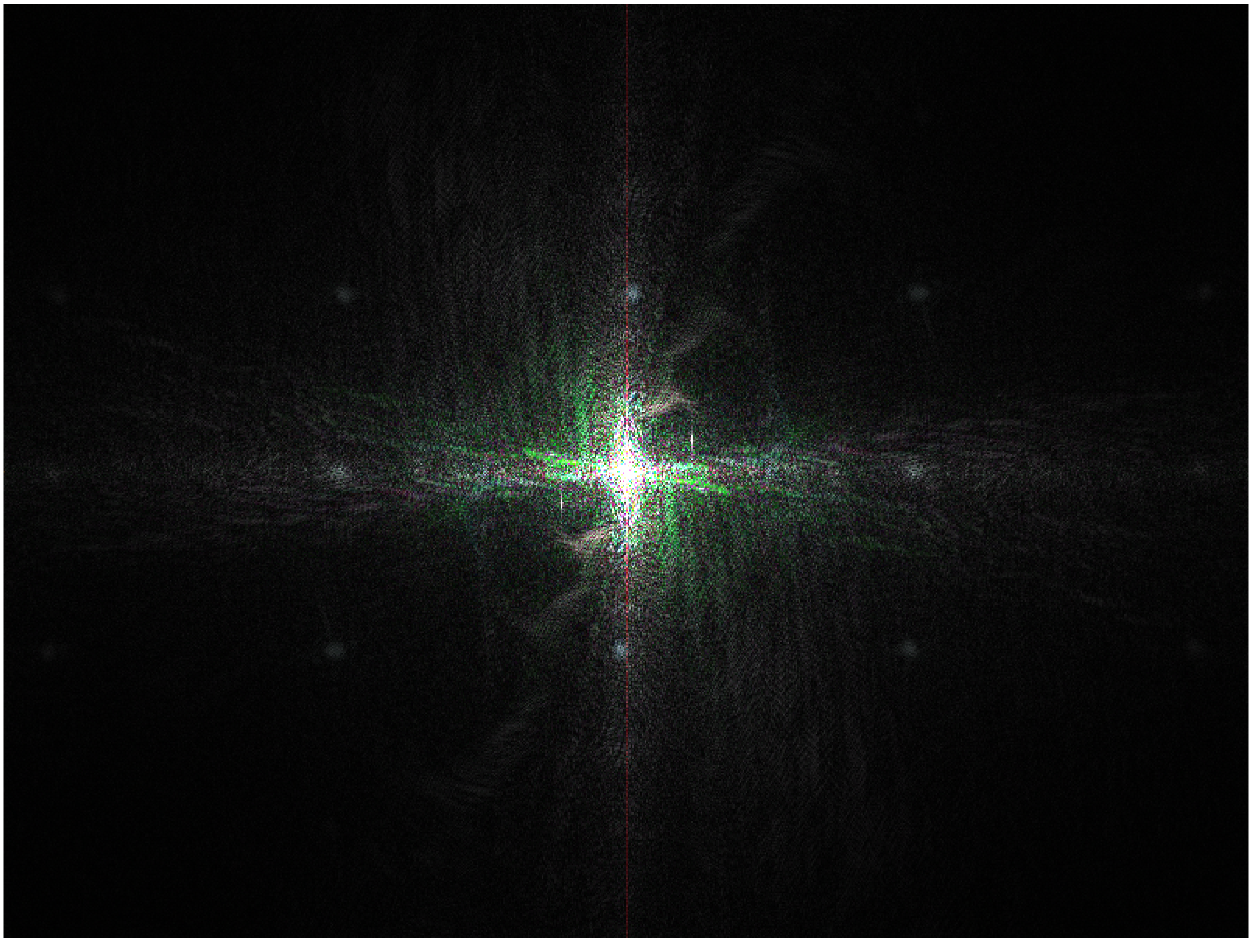


Figure 26. Power spectrum of image with periodic noise. Note the dots.

The dots in the figure above show the noise at the given frequencies. The exact frequency and direction can be calculated with the following formula.

lambda = sqrt((C/u)^2 + (R/v)^2)
omega = 1/lambda
theta = atan((v*C)/(u*R))

To zero out the frequencies reponsible for the noise the following mask was created.

Figure 27. Mask created to zero out the noise frequencies.

The x,y points I zeroed out:

220	187
404	187
589	189
581	416
397	416
213	416

Applying the above filter produced the resulting image:

Figure 28. Filtered image.

The noise is still visible, but has been somewhat improved.

Experiment 3

a) To filter the image in the spatial domain I used a convolution mask. As the mask I used a Gaussian that was 8x8 with a sigma of 2, which is known to be optimal in terms of space frequency localization. Below is an enlarged section of the original noisy image and the filtered image for comparison.

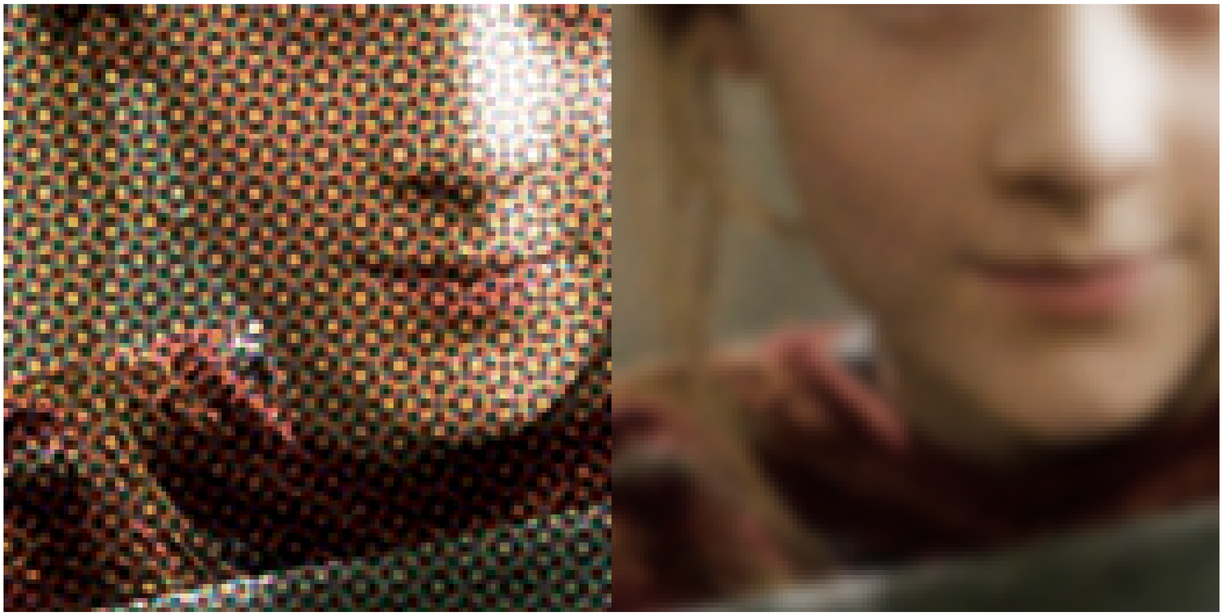


Figure 29. Enlarged regions of the original HTD image and the blurred image.

As you can see the slight blurring greatly improved the quality of the image. The enlarged section on the right that has been blurred appears much clearer.

b) The section deals with unsharp masking. This is a method that attemtps to bring back in some of the high frequency content that was lost in the blur. To acheive this I blurred the image again by using an 8x8 Gaussian with a sigma of 3 as the convolution mask. Then I created a difference image by subtracting this re-blurred image from the original blurred image. The final unsharp mask image, K, is created with the follow equation:

```
>> K = (1 + a)D + B
```

I experimented with different values for a and decided the best was .5. The results of the unsharp mask operation are shown below.

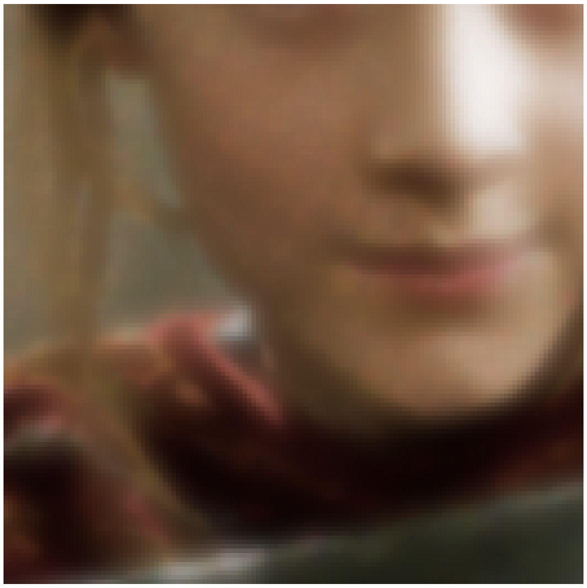


Figure 30. Results of the unsharp mask filter.

As the parameter a gets larger, the edges start to glow as they are being added back in too strongly.

c) For this part I used the function removeNoise(). It was not able to totally remove all the noise, but a good amount of it was removed. The results of this function are displayed below.

Figure 31. Results of the frequency domain filtering.

While this did not totally remove all the HTD, it could be used in combination with spatial domain blurring. If some of the noise is first removed with this more targeted approach, the spatial domain blurring does not have to be as strong.

Conclusion

Each of these experiments is an exercise in removing noise. None of them is perfect, and they all have their pros and cons. While blurring is the simplest, it is somewhat heavy handed and sometimes knocks out features we want to preserve. Is the noise is of a periodic nature we can move to the frequency domain to specifically target the frequencies of the noise and not touch other aspects of the image. Using both of these techniques, as well as unsharp masking, one should be able to remove a large variety of different noise additions.