Image Processing I

EECE 253 Fall 2011

Assignment 6: Filtering for Noise Reduction and Edge Enhancement

Due at 12:00 midnight on Monday 28 November 2011

The goals of this lab are to learn: how to perform image noise reduction, both (1) random and (2) periodic, and (3) how to remove half-tone dot distortion in the frequency domain. Furthermore, grad students will write and test a color median filter program.

In this lab assignment (and all the others) you will be writing your own image processing functions. Except as specified below, you may not use the functions from the `Matlab` image processing toolkit by themselves or within your functions to solve the problems in the labs. You may compare the results of your own functions against those of the IP toolkit for your own information but those results may not be used in your lab reports.

Help can be gotten on any `matlab` command (or function) by typing `help command_name`, *e.g.* to get help on `imwrite()`, type `help imwrite` at the command prompt. Another way is to press the <F1> key, select the **index** tab in the window, and type the command name into the search box.

In completing this assignment (and the others) if specific images are not supplied, you may use any images you like providing that they are of the type specified by the problem description (*e.g.* 24-bit truecolor). Please do not use images that are obscene or gruesome. In general, an image is OK if you could show it to your grandmother without upsetting her or embarrassing yourself. A wide variety of images are available under creative commons license on the web site `www.flickr.com`. If you are not the photographer, be sure to credit the person who did take the picture.

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 windows[1] 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

[1] The window size should be in the range 128×128 to 256×256.

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.

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.

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, h_1, – 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

   $$K = (1 + \alpha)D + B \tag{1}$$

   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.

(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.

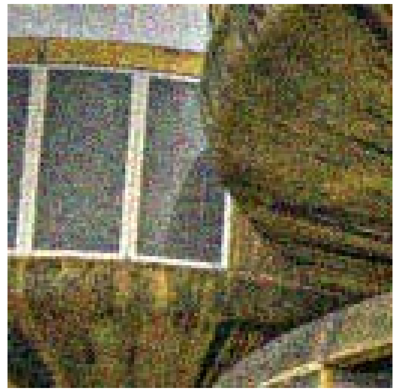
How successful was blur-then-unsharp-mask approach to the removal of halftone distortion? Did the USM significantly improve the blurring the edges?. Compare those results to the frequency filter method. Which was superior? What are the similarities and differences in the appearances of the results?

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, h_1, 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 h_2 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 $[0, \ldots, 255]$ 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.

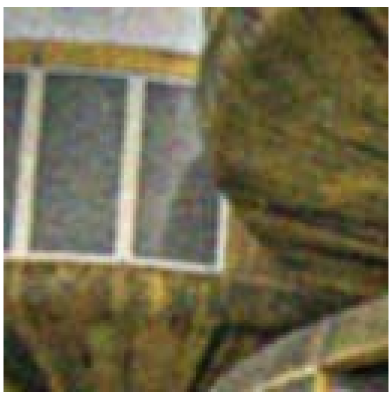
[Note: **Do not** use the intrinsic filter `fspecial('unsharp',0.5)` to do this. It is a different operation than the one I want you to do.]

4. **Extra problem for graduate students.** Write a `Matlab` program to perform color median filtering. Test it on image, `Gilles Tran - Glasses - POV-Ray - Noise.jpg`. Display the result in your report. Compute the per pixel energy of the difference between your result and the original image, `Gilles Tran - Glasses - POV-Ray.jpg`. How does this value compare to those from Problem 1, above.

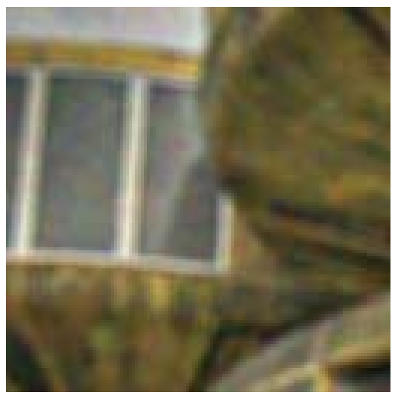
(a) Original Image, $\varepsilon = 0$



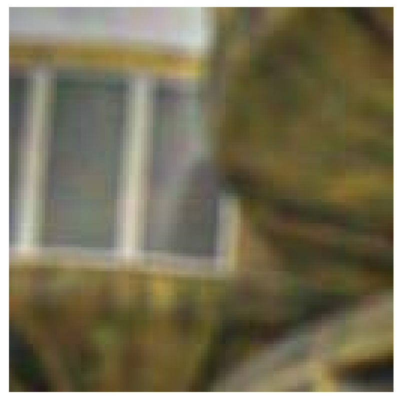
(b) Noisy Image, $\varepsilon = 83.9322$



(c) 3×3 blur, $\varepsilon = 43.1729$



(d) 5×5 blur, $\varepsilon = 41.9308$



(e) 7×7 blur, $\varepsilon = 46.2843$

(f) 3×3 LMS, $\varepsilon = 11.6897$

(g) 5×5 LMS, $\varepsilon = 22.9560$

(h) 7×7 LMS, $\varepsilon = 30.0118$

Figure 1: Comparison of a single window from the original, the noisy, and the noise reduced images. ε is the per pixel energy in the difference between the given image and the original. (The energy measurements are from the entire images, not just this window.) Original image by CypherOne on `www.flickr.com`.

Rules for assignments

1. All work must be yours and yours alone. Collaboration on the assignments is forbidden, with the following exceptions:

   (a) You may obtain help on any aspect of the homework from either Prof. Peters or the TAs for this course.

   (b) You may obtain technical help on the *programming language* you are using from anyone you wish.

   (c) You may get help in obtaining the *input* images for the assignments from anyone you wish.

   (d) You may get help in the formatting or storing or transmission of your reports from anyone you wish.

   (a) Explain the tasks you performed in detail.

   (b) Include in the report the original image you used and those resultant images that I specified above.

   (c) Include all computer code that you wrote and used, clearly documented, in an appendix.

   (d) Write your results in a clear laboratory report format using MS Word, WordPerfect, LaTeX, or any other word processor with which you can embed images in text. Name the file which contains your report:

   `YourName_eece253_F2005_Assignment_5.doc`

   or `.pdf` or .whatever depending on the file type. Please upload your report file to blackboard.