SD 475 Image Processing

Fall 2011

Lab 2: Image Enhancement and Restoration
Friday, November 11th, 2011

Note: All lab reports will be submitted electronically to the TA's email address a42kumar@uwaterloo.ca. For help on how to use the Matlab functions mentioned throughout the lab, please type 'help' followed by the function (e.g., help imread)

1 Overview

The goal of this lab is to provide some hands-on experience with fundamental image enhancement and restoration concepts and techniques in both the spatial and frequency domain.

Image enhancement and restoration are important aspects of image processing, as many real-world applications depend on the underlying quality of an image. Examples include photo enhancement, object recognition and tracking in surveillance videos, and segmentation of important characteristics (e.g., tumors) in medical images. For this lab, we will study some fundamental image enhancement and restoration techniques such as noise reduction, image sharpening, and deblurring.

The following images will be used for testing purposes:

- lena.tif

- cameraman.tif

All of these images are included with Matlab and can be loaded using the $imread$ function.

NOTE: The proper way to calculate PSNR is the following (for images normalized to the range 0 to 1):
$$\text{psnr = 10*log10(1/mean2((f-g).\hat{} \ 2))}$$

2 Noise Generation

To test the effectiveness of an image processing algorithm, it is often necessary to evaluate its performance under various noise levels to allow for a systematic comparison with other techniques. It is generally not possible to capture real-world images at many noise levels. Furthermore, it is not possible to quantitatively

evaluate the performance of a method when applied to real-world noisy image scenarios since there is no reference image to compare against. As such, it is often necessary to generate synthetic noise at different noise levels based on known noise models to evaluate image processing algorithms.

For this study, we will apply additive zero=mean Gaussian (with variance of 0.01), salt and pepper (with noise density of 0.05), and multiplicative speckle noise (with variance of 0.04) to a synthetic toy image separately using the $imnoise$ function. The toy image consists of two grayscale bars and can be generated use the following code

f = [0.3*ones(200,100) 0.7*ones(200,100)];

Plot the noise contaminated images and the corresponding histograms for each of the noise models.

1. Describe each of the histograms in the context of the corresponding noise models. Why do they appear that way?

2. Are there visual differences between the noise contaminated images? What are they? Why?

3. In the speckle noise case, what is the underlying distribution used? Can you tell from the histogram? How?

4. In the speckle noise case, you will notice that the peaks of the histogram are no longer of the same height as they were in the original image. Also, the spread around each of the peaks is also different from each other. Why? Hint: Noise is multiplicative.

3   Noise Reduction in the Spatial Domain

Let us now study different noise reduction techniques based on spatial filtering, as well as the effect of filter parameters on image quality. Load the Lena image and convert it to a grayscale image using the $rgb2gray$ function. Furthermore, to get intensity of the image within the range of 0 to 1, use the $double$ function on the image and then divide it by 255. To evaluate the noise reduction performance of various noise reduction techniques, we will contaminate the Lena image with zero-mean Gaussian noise with a variance of 0.002. Plot the noisy image and the corresponding histogram and PSNR between the noisy image and the original noise-free image.

First, let us study the averaging filter method as well as the effect of window size on the noise reduction performance of a spatial filter. Create a 3×3 averaging filter kernel using the $fspecial$ function. Now apply the averaging filter to the noisy image using the $imfilter$ function. Plot the denoised image and the corresponding histogram. Also, compute the PSNR between the denoised image and the original noise-free image.

1. Compare the visual difference between the noisy image and the denoised image. How well did it work? Why? Did the PSNR decrease?

2. Compare the histograms of the noise-free, noisy, and denoised images. What happened? Why?

3. Based on visual quality of the denoised image, what are the benefits and drawbacks associated with the average filter?

Let us now create a 7×7 averaging filter kernel and apply it to the noisy image. Plot the denoised image and the corresponding histogram. Also, compute the PSNR between the denoised image and the original noise-free image.

1. Compare the visual difference between the denoised image from the 7×7 filtering kernel and the denoised image from the 3×3 filtering kernel. Are there any differences? Why? Did the PSNR decrease? Why?

2. Compare the histograms of the two denoised images. What are the differences? Why?

3. Based on visual quality of the denoised image, what are the benefits and drawbacks associated with using a larger window size?

Let us now create a 7×7 Gaussian filter kernel with a standard deviation of 1. Plot the denoised image and the corresponding histogram. Also, compute the PSNR between the denoised image and the original noise-free image.

1. Compare the visual difference between the denoised image from the Gaussian filtering kernel and the denoised images from the averaging filter kernels. Are there any differences? Why? Did the PSNR decrease? Why?

2. Compare the histograms of the denoised image using the Gaussian filtering kernel and the denoised images from the averaging filter kernels. What are the differences? Why?

3. Based on visual quality of the denoised image, what are the benefits and drawbacks associated with using a Gaussian kernel as opposed to an averaging kernel?

Let us now create a new noisy image by adding salt and pepper noise to the image. Now apply the 7×7 averaging filter and the Gaussian filter to the noisy image separately. Plot the noisy image, the denoised images using each method, and the corresponding histograms. Also, compute the PSNR between the denoised images and the original noise-free image.

1. How does the averaging filter and Gaussian filtering methods perform on the noisy image in terms of noise reduction? Explain in terms of visual quality as well as PSNR. Why do we get such results?

2. Compare the histograms of the denoised images with that of the noisy image. What characteristics are present in all of the histograms? Why?

Let us now apply the median filter on the noisy image. The $medfilt2$ function will come in handy for this. Plot the denoised image and the corresponding histogram. Also, compute the PSNR between the denoised image and the original noise-free image.

1. How does the denoised image produced using the median filter compare with the denoised images produced using averaging filter and Gaussian filtering methods? Explain in terms of visual quality as well as PSNR. Why do we get such results with median filter when compared to the other spatial filtering methods?

4   Sharpening in the Spatial Domain

Let us now briefly study sharpening techniques based on spatial filtering as well as the effect of sharpening filter parameters on image quality. Load the Cameraman image and get intensity of the image within the range of 0 to 1. One very useful and customizable technique for sharpening images is high-boost filter. Let us study it at its various stages. First, apply the 7×7 Gaussian filter on the cameraman image and subtract the Gaussian-filtered image from the original cameraman image. Plot both the Gaussian-filtered image and the subtracted image.

1. What does the subtracted image look like? What frequency components from the original image are preserved in the subtracted image? Why?

Now we add the subtracted image to the original image. Plot the resulting image.

1. What does the resulting image look like? How does it differ from the original image? Explain why it appears this way.

Now, instead of add the subtracted image to the original image, we multiply the subtracted image by 0.5 and then add it to the original image. Plot the resulting image.

1. Compare the results produced by adding the subtracted image to the original image and that produced by adding half of the subtracted image to the original image. How does it differ? Explain why it appears this way.

2. What does multiplying the subtracted image by a factor less than one accomplish? What about greater than one?

5   Noise Reduction in the Frequency Domain

Let us now study noise reduction techniques based on frequency domain filtering as well as the effect of filter parameters on image quality. Load the Lena image (adjust intensities to range of 0 to 1) and apply additive Gaussian noise with variance of 0.005 to the image. Plot the Log Fourier spectra of the original image and the noisy image by using the *log* function on the Fourier spectra.

1. Compare the two Fourier spectra. What are the differences? Where is these differences most visually prominent? Why?

Now let us study the ideal low-pass filter. To create an ideal low-pass filter with a cut-off radius r, first you need to create an image of a white circle with radius r. One approach to do use the $fspecial$ function in the following way:

h = fspecial('disk',r); h(h > 0)=1;

Now create a black image (representing an energy-less Fourier spectra) and center the circle onto the black image:

h_freq = zeros([height of image],[width of image]);
h_freq([[height of image]/2-r:[height of image]/2+r],[[width of image]/2-r:[width of image]/2+r])=h;

Create and plot the Fourier spectra of the resulting low-pass filter h_freq with a radius of 60. Now, apply the filter on the noisy image in the frequency domain and then perform an inverse Fourier transform. $ifft2$ and $ifftshift$ will come in handy. Plot the resulting denoised image and the corresponding PSNR.

1. Describe the appearance of the denoised image compared to the original and the noisy images. Why does it look this way? What does the ideal low-pass filter do?

2. There is a particular artifact present in the restored image. What is it and why does it happen?

Now create a low-pass filter with a cut-off radius of 20 and apply it to the noisy image in the frequency domain and then perform an inverse Fourier transform. Plot the resulting denoised image and the corresponding PSNR.

1. Compare the denoised image with the denoised image using a cut-off radius of 60. How does the image and the PSNR differ? Why?

2. What conclusions can you draw about the relationship between cut-off radius and resulting image after filtering? What is the trade-off in terms of noise reduction?

Now let's do the same thing with a Gaussian low-pass filter. Create a Gaussian low-pass filter kernel with a standard deviation of 60 and normalize it based on the highest value in the kernel. max and functions will come in handy. Apply the Gaussian low-pass filter to the noisy image in the frequency domain and then perform an inverse Fourier transform. Plot the resulting denoised image and the corresponding PSNR.

1. Compare the denoised image with the denoised images produced using the ideal low-pass filters. How does the image and the PSNR differ? Is it better or worse? Why? Does it have the same type of image artifacts?

6   Image Restoration in the Frequency Domain

Let us now study two frequency domain based image restoration methods: inverse filtering and Wiener filtering. Load the Camerman image (adjust intensities to range of 0 to 1). First, create a disk blur function of radius of 4 and apply it to the image in the frequency domain:

$$h = fspecial('disk',4);$$
$$f = im2double(imread('cameraman.tif'));$$
$$h\_freq = fft2(h,size(f,1),size(f,2));$$
$$f\_blur = real(ifft2(h\_freq.*fft2(f)));$$

Plot the blurred image and the corresponding PSNR. Now apply inverse filtering to it by dividing the image by the blurring function h_freq and plot the result.

1. Compare the restored image with the original image and the blurred image. How does the restored image and the PSNR differ from the blurred image? Is it better or worse? Why?

Now add zero-mean Gaussian noise with a variance of 0.002 to the blurred image. Apply inverse filtering to the blurred and noisy image. Plot the restored image and the PSNR.

1. Compare the restored image with the restored image from the previous step. How does the restored image and the PSNR differ from the previous restored image? Is it better or worse? Why?

2. Can you draw any conclusions about inverse filtering when applied to noise degraded images?

Let us study the Wiener filter for image restoration. Apply Wiener filtering on the blurred and noisy image used in the previous step. Use the $deconvwnr$ function and pass in the disk blur function as the point-spread function (PSF). Plot the restored image and the PSNR.

1. Compare the restored image with the restored image from the previous step. How does the restored image and the PSNR differ from the previous restored image? Is it better or worse? Why? Explain it in context with the concept behind Wiener filtering.

2. Can you draw any conclusions about Wiener filtering when applied to noise degraded images?

7   Report

Include in your report:

- A brief introduction.

- Printouts of pertinent graphs and images (properly labelled).

- Printouts of code

- Include responses to all questions.

- A brief summary of your results with conclusions.