SD 575 Image Processing

Fall 2018

Lab 2: Image Enhancement
Due Monday October 15, 2018 at 11:59 PM (Midnight)

Note: All lab reports will be submitted on learn in the corresponding group dropbox.

For help on how to use the Matlab functions mentioned throughout the lab, please type 'help' followed by the function (e.g., `help imread`). Function information can also be found through Google.

1 Overview

The goal of this lab is to study noise reduction and provide some hands-on experience with fundamental image enhancement and restoration concepts in the spatial 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 and image sharpening.

The following images will be used for testing purposes:

- mandrill.png
- cameraman.tif

These images can be downloaded from the course website, 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):

```
psnr = 10*log10(1/mean2((f-g).^ 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 various 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 mandrill 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 (alternatively use `im2double`). To evaluate the noise reduction performance of various noise reduction techniques, we will contaminate the mandrill 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. Plot this filter using `imagesc` and `colormap(gray)`. 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 filter. 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. 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 adding the subtracted image to the original image, 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 Report

Include in your report:

- A brief introduction.
- Pertinent graphs and images (properly labelled).
- Include code (can be included in appendix).
- Include responses to all questions.
- A brief summary of your results with conclusions.