Computer Project 3: Image Enhancement

Name: Linsong Zhan

Due Date: April 20th at 11:59pm

1. Introduction

1.1. Purpose of the computer project

The goal of this computer assignment is to study the effects of various point and spatial operators for contract enhancement, noise removal, edge extraction and sharpening.

1.2. What will be accomplished or carried out?

Contrast Enhancement: In this part, I will try different number of bins and compare the results and effects on the histogram. Devise a "desired histogram" by applying specification algorithm. Then I will compare the histogram of the resultant image with those of the histogram equalization. Finally, I will try my designed algorithm on "Boat" image and see if the algorithm works.

Spatial Filtering: In this part, the input image will be a noisy version of "Lena" image by adding white Gaussian noise with SNR=5dB. I will use the MATLAB function "conv2" to apply various 2D low-pass filtering masks of different sizes and coefficients to remove the effects of the additive noise from the noisy image. Then I can compare the results of these filters in terms of SNR improvements and comment on their performance on frequency selectivity using function freqz2.

Median Filtering: In this part, the input image will be a noisy version of "Lena" image by adding "salt and pepper" noise with different levels of noise. I will apply the median filters of size 3x3, 5x5, 7x7 using the function "medfilt2" to the noise image. Then I will increase the level of noise to determine experimentally the maximum noise level that can be tolerated without producing considerable blurring artifacts.

Edge Sharpening: In this part, I will apply two different edge extraction masks by using function "edge" to sharpen the edges and improve the visual appearance of the "Boat" image. Then I can compare the results of these algorithms in terms of visual appearance.

2. Theory

2.1. Algorithm for histogram equalization and specification

The goal of histogram equalization is to devise a mapping that yields an image with a uniformly distributed histogram.

If we assume that image intensity is continuous represented by continuous r.v. X, then, the cumulative distribution function of the original image can be used as the mapping function to yield a uniformly distributed image.

The CDF of the digital image is

$$P_X(x) = \int_0^x P_X(\alpha)d\alpha = \sum_{k=0}^{K-1} \frac{N_k}{N} u(x - x_k)$$

Where $u(x)$ represents a unit step function.

If we use this CDF as the grey-scale mapping, for $x \in [x_i, x_{i+1}]$ the intensity levels of the mapped image are obtained using

$$y_i' = f(x) = P_X(x) = \sum_{k=0}^{i} \frac{N_k}{N}$$

There are two problems. First, the mapped values do not represent the actual intensity levels since it's normalized. Also, the PDF of the mapped image is a scaled version of the original PDF with the same number of bins and same amplitudes but with different spacing between the bins that depend on $P_X(x)$ mapping.

These problems can be solved by performing a uniform quantization mapping of the form

$$y_l = Int[\frac{(y_l' - y_{min}')}{1 - y_{min}'}(y_L - y_{L0}) + y_{L0} + 0.5]$$

Where y_{min}' is the smallest value of y_l'. y_{L0} and y_L are the smallest and largest intensity allowed in the mapped image y and $Int[x]$ represents integer part of quantity x.

2.2. Algorithm for Spatial Filtering

Effective method for removing additive Gaussian noise from noisy images. Uses a linear 2-D FIR filter where each pixel in an image is replaced by the weighted sum of the neighboring pixels within the mask i.e.

$$y(m,n) = \sum\sum_{k,l \in W} h(k,l)x(m-k,n-l)$$

Where $x(m,n)$: input image, $y(m,n)$: output image, $h(k,l)'s$: filter coefficients or the weights, and W: a suitable mask. Note that for low-pass filtering $\sum\sum_{k,l \in W} h(k,l) = 1$. A common choice is $h(k,l) = \frac{1}{N_w}, \forall (k,l) \in W$ i.e. "spatial averaging" where N_W represents the number of pixels in the mask, W.

2.3. Algorithm for Median Filtering

Median filtering is a nonlinear filtering process primarily used to remove impulsive or "salt & pepper" type noise. Similar to the spatial filtering, median filter operation involves sliding a window encompassing an odd number of pixels. The center pixel in the window is then replaced by the median of the pixels within the window, i.e.

$$y(m,n) = median[x(m-k,n-l)], \qquad k,l \in W$$

Finding the median value requires arranging pixel intensities in the window in increasing or decreasing order and picking the middle value. Generally, window size, N_W, is chosen to be an odd number to facilitate the selection of the median.

2.4. Algorithm for Edge Sharpening

Used to improve the visual appearance of images with less prominent edges. This is because an image with accentuated edges is subjectively more pleasing to human eyes than an exact reproduction. Edge extraction is also an important part of typical image analysis systems that rely on contours or other edge-based features to discriminate different objects.

The basic idea is to detect the edges of an image using a HPF or a 2-D gradient operator. To sharpen or emphasize the edges, a fraction of this gradient or high-pass filtered image is then added to the original image. The gradient or high-pass image can also be generated by subtracting an unsharp or blurred or low-pass filtered image from the original image. In general, the unsharp masking operation can be represented by,

$$y(m,n) = x(m,n) + \lambda g(m,n)$$

Where $g(m,n)$ is a suitable gradient computed for the mask centered at (m,n) and $\lambda > 0$ is a proportionality constant. The pixel at (m,n) is an edge pixel if $g(m,n)$ exceeds a pre-specified threshold. This threshold can typically be decided by examining the histogram of the image $g(m,n)$. The commonly used edge detection operators are the Roberts, Prewitt, Sobel and 2-D discrete Laplacian operators.

3. Results and Discussions

3.1.1. Histogram Equalization

Figure 1(a)&(b) show the “Pepsi” image and the output image after histogram equalization. The improvement in the visual appearance of these images over the original one is clearly noticeable. As discussed in the theory part, the histogram equalization increases the dynamic range of gray scale of the image, thus enhancing contrast of the image. The shortcoming of this method is that it’s hard to control the specific effects in the image. You always end up getting a globally equalized histogram.

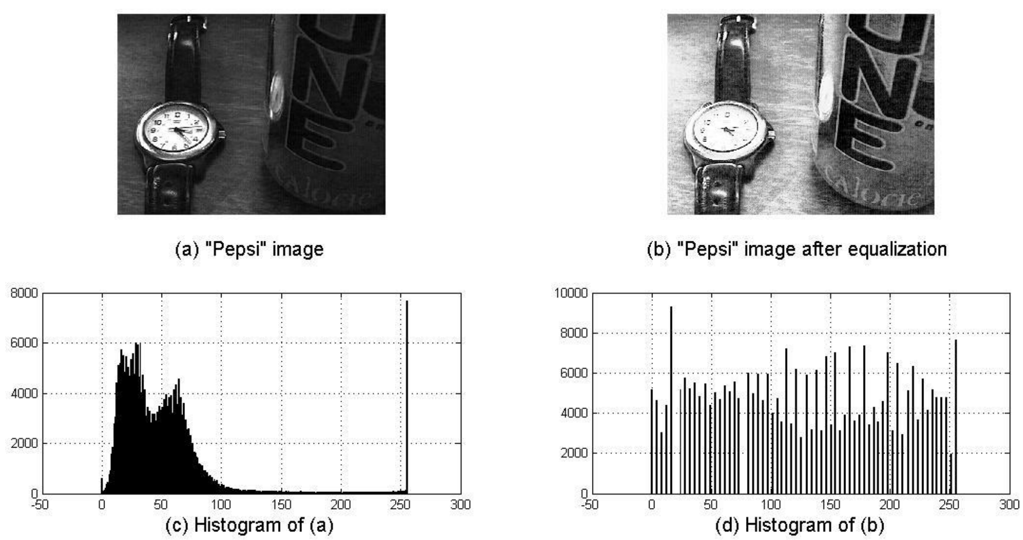


Figure 1: Histogram Equalization.

In MATLAB, the histogram equalization is achieved by the function $histeq$, the function syntax is

$$J = histeq(I, nlev)$$

Where I is the input image, $nlev$ is the appointed discrete gray levels. If $nlev$ is much less than L, the histogram of J is flatter. The default value of $nlev$ in $histeq$ is 64. In general, we assign the value of L to $nlev$ (256 in general) to make the output image look better. Figure 2(c)&(d) below show perfectly the histograms after applying different gray levels to the original image. Loss of grey level resolution leads to some patchiness distinguishable in the mapped image more so in Figure 2(b) [1].

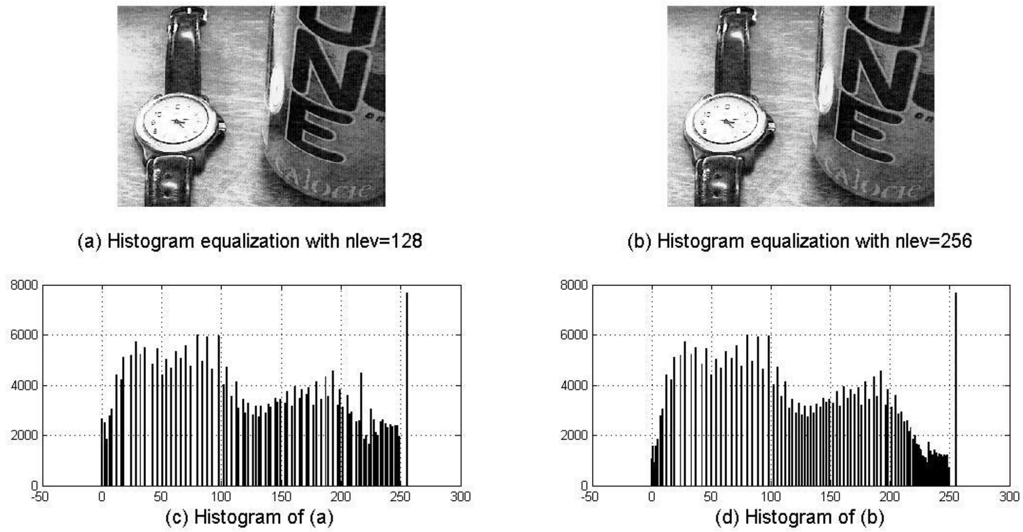


Figure 2: Histogram Equalization with different “nlevs”.

3.1.2. Histogram Specification

Figure 3(a) shows the histogram specified version of the “Pepsi” image in Figure 1(a) for a specified histogram in Figure 3(b) which is ½ cycle of a sinewave. The histogram of the resultant image in Figure 3(a) is given in Figure 3(c) which is similar in shape to the specified desired in Figure 3(b). Figure 3(d) again shows the histogram of the image after equalization for comparison [1]. The image in Figure 3(a) shows much better enhancement without introducing lots of undesirable artifacts compared with these in Figure 1&2 visually. Specifically, the hour hand and minute hand of the watch in Figure 3(a) is more clear than these in Figure 1&2.

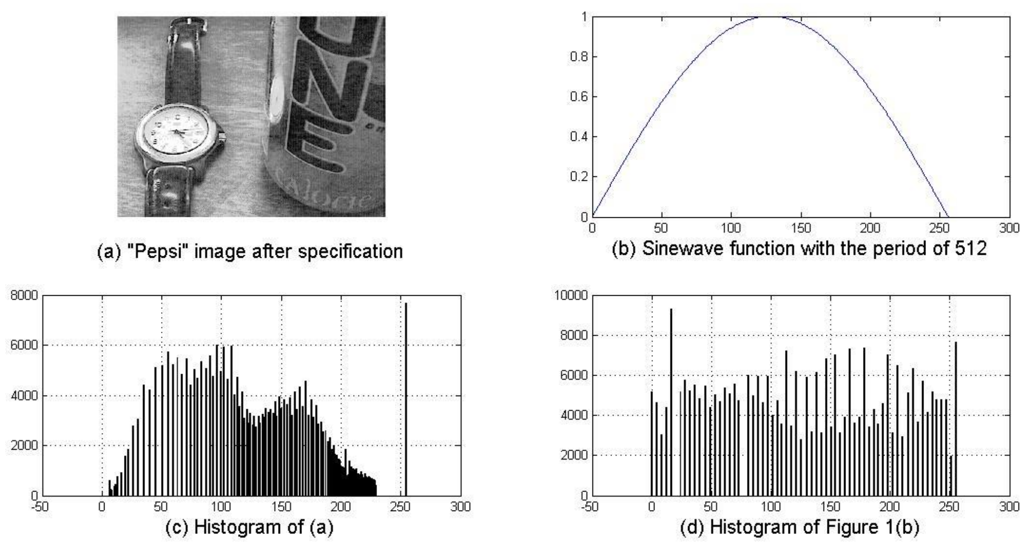
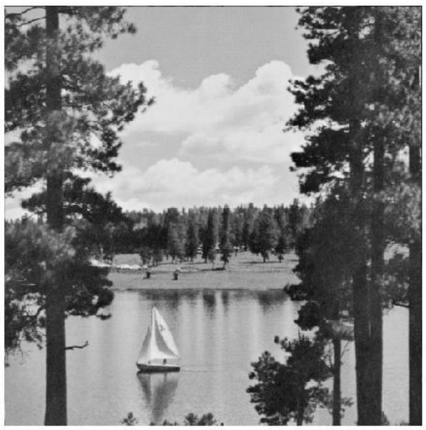
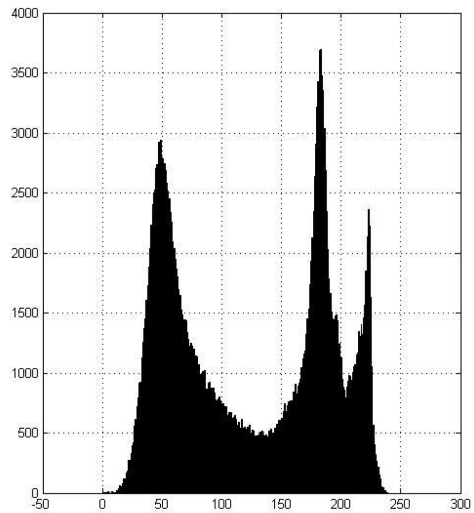


Figure 3: Histogram Specification with ½ cycle of a sinewave.

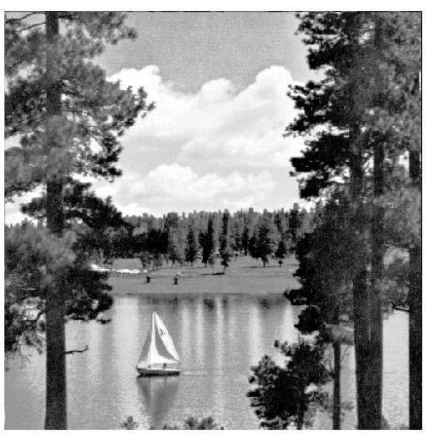
Figure 4(a)&(b) shows the "Boat" image and its histogram. Figure 4(c) shows the histogram equalization. Figure 4(d) shows the specified version of the "Boat" image in Figure 4(a) for a specified histogram in Figure 3(b) which is ½ cycle of a sinewave. Visual comparison of the images in Figure 4(c)&(d) with the one in Figure 4(a) is clearly noticeable. While the histogram specification in Figure 4(d) introduces more blurring artifacts than histogram equalization in Figure 4(c) does. I think this is due to the wide dynamic range of gray scale of the "Boat" image as shown in Figure 4(b).



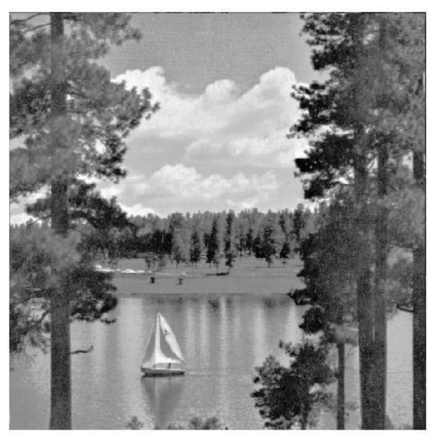
(a) "Boat" image



(b) Histogram of (a)



(c) "Boat" image after equalization



(d) "Boat" image after specification

Figure 4: "Boat" image and its filtered images

3.2. Spatial Filtering

The noisy version of “Lena” image with SNR=5dB will be used as the input to all filters in this part.

Original image

Noisy image with SNR=5dB

Figure 5(a)&(b) show two different 3×3 masks with different choices of coefficient. Figure 5(c)&(d) show averaging masks of size 4×4 and 6×6. The choices of the filter coefficients and window size present a trade-off between noise removal ability and the edge smearing artifacts caused due to loss of high frequency information [1].

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

(a) Mask 1

0	1/5	0
1/5	1/5	1/5
0	1/5	0

(b) Mask 2

1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16
1/16	1/16	1/16	1/16

(c) Mask 3

1/36	1/36	1/36	1/36	1/36	1/36
1/36	1/36	1/36	1/36	1/36	1/36
1/36	1/36	1/36	1/36	1/36	1/36
1/36	1/36	1/36	1/36	1/36	1/36
1/36	1/36	1/36	1/36	1/36	1/36
1/36	1/36	1/36	1/36	1/36	1/36

(d) Mask 4

Figure 5: Different Spatial Low-Pass Filtering Masks.

Figure 6(a)&(b) show filtered images ($SNR_1 = 20.2dB$ and $SNR_2 = 18.88dB$) using Masks 1 and 2 respectively. The noise removal ability (measured by SNR) of the first mask is better, the smearing artifacts are also more noticeable in Figure 6(b).

Figure 6: Noisy and Filtered Lena Images.

Figure 7(a)&(b) compare magnitude responses of these filters. The first mask has a smaller bandwidth, which means better noise removal ability and more blurring artifacts.

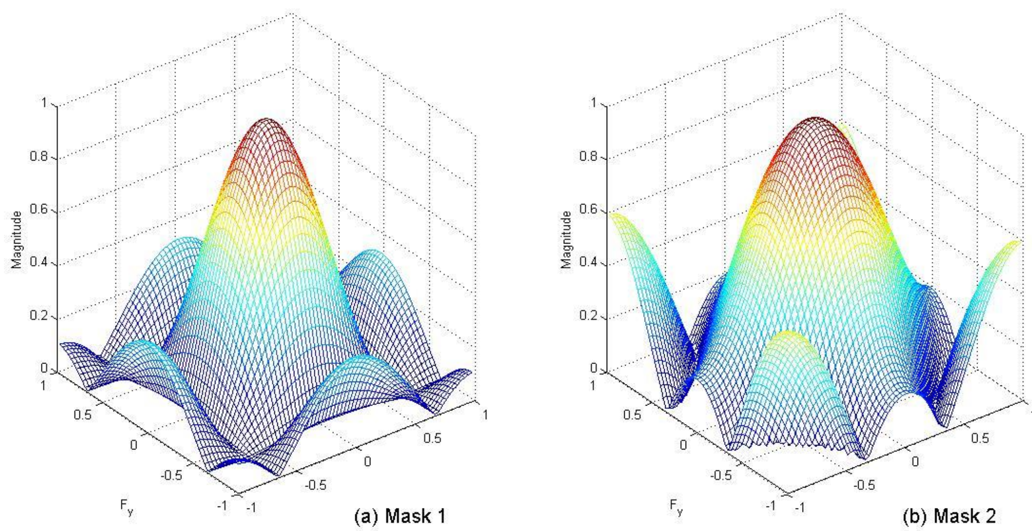


Figure 7: Magnitude Responses for Masks 1 and 2.

Figure 8(a)&(b) show the results of applying spatial averaging filters of size 4×4 and 6×6 respectively. The filtered image of 6×6 mask in Figure 8(b) shows more edge smearing artifacts than one in Figure 8(a) because of the reduced bandwidth of the spatial filter with the increasing of window size.

Figure 8: Filtered Images.

3.3. Median Filtering

Figure 9(a)-(d) show the “Lena” image corrupted with salt & pepper noise density of 40% and median filtering results using 3×3, 5×5, 7×7 size windows. As can be observed, the median filter performs well. The 3×3 window shows some remaining noises due to this high density of noise while it causes least blurring artifacts among the other two. The smearing artifacts are more obvious in the hairy areas in Figure 9(d) compared with others.

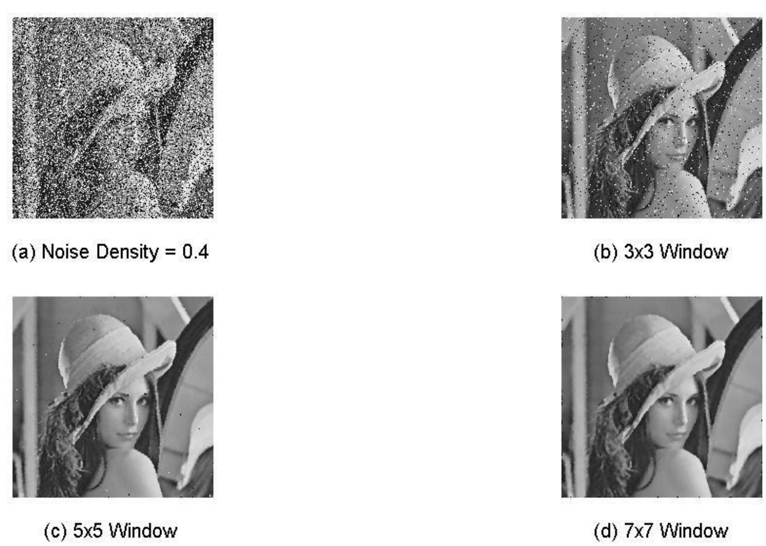


Figure 9: Noisy “Lena” and Median Filtered Versions with Different Sizes.

After changing the value of noise density d from 0.05 to 0.4, I find that the 3×3 window performs an acceptable image and causes least blurring artifacts when d is equal to 0.1 approximately. Figure 10 shows the maximum level of noise without producing noticeable blurring artifacts.

(a) Noise Density = 0.1　　(b) 3x3 Window

Figure 10: Noisy "Lena" and Best Median Filtered Version with d=0.1.

3.4. Edge Sharpening

Figure 11 shows the original "Boat" image which will be the input image in this part.

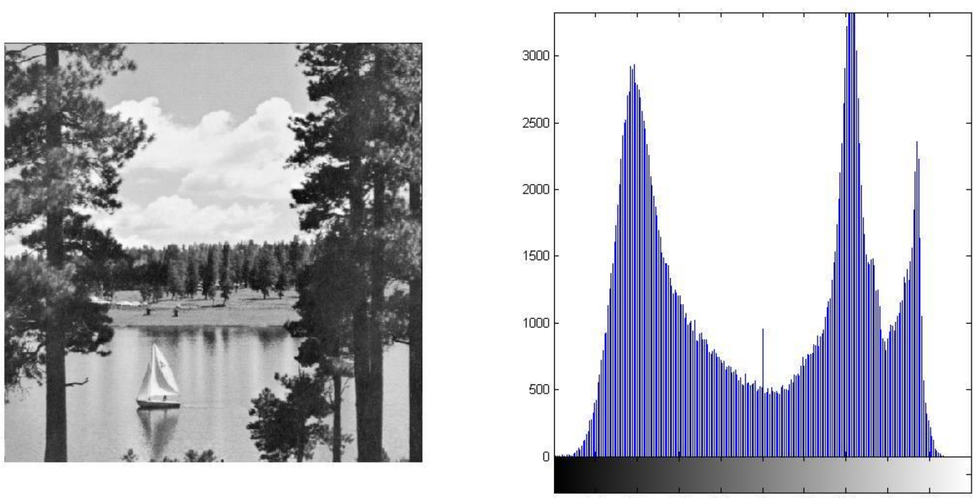
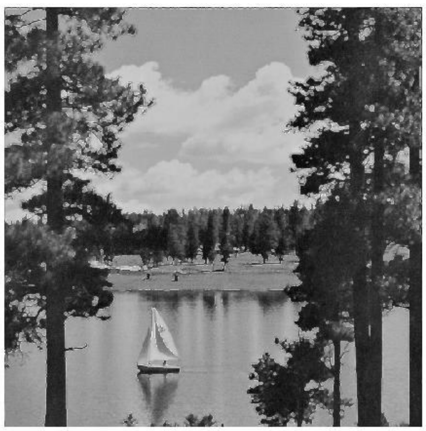
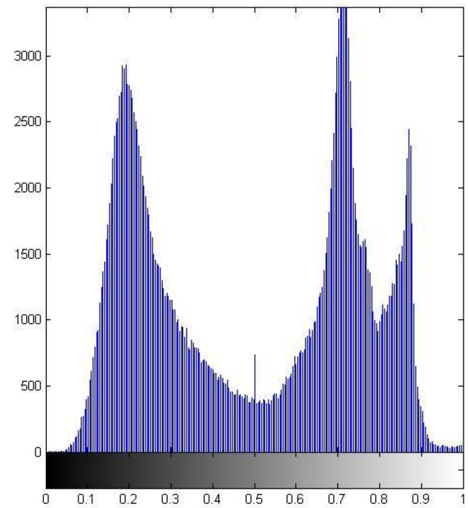


Figure 11: Original "Boat" image.

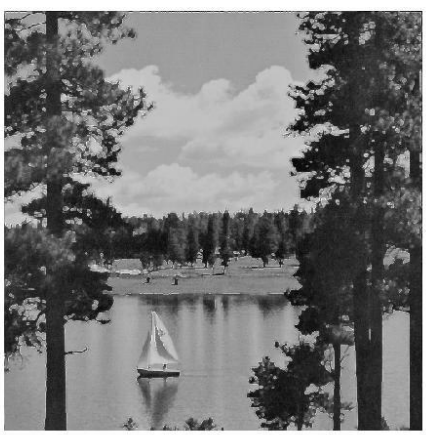
Figure 12 (a)-(d) show edge sharpened "Boat" images using four methods: Sobel, Prewitt, Laplacian and Canny. The "lamda" λ for each of them is assigned to different values to avoid considerable edge saturations.

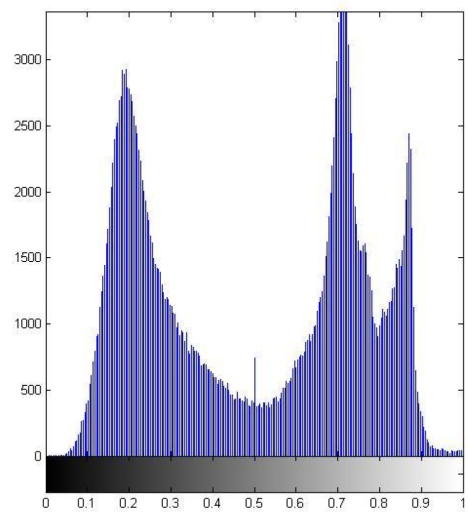
As can be observed, the histogram of edge sharpened image using Sobel method shows subtle difference from the one using Prewitt method. Figure 12(a)&(b) show some levels of edge sharpening especially near the area of strong edges like tree branches, while they are less likely to detect the weak edges like the shadow of trees in the lake.





(a) Sobel with lamda = 75

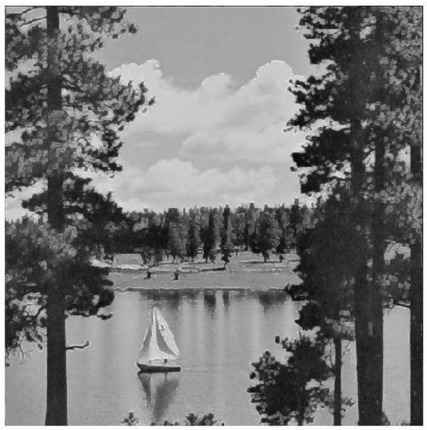


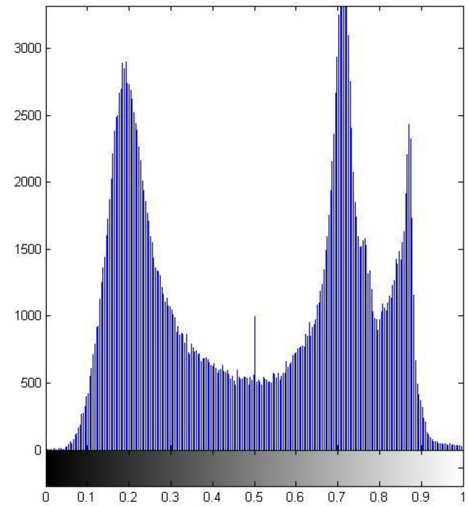


(b) Prewitt with lamda = 75

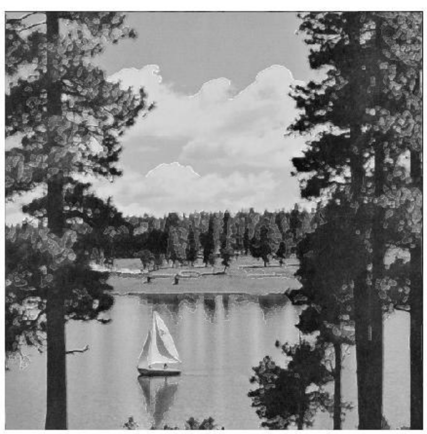
Figure 12: Edge Sharpened "Boat" images using four methods.

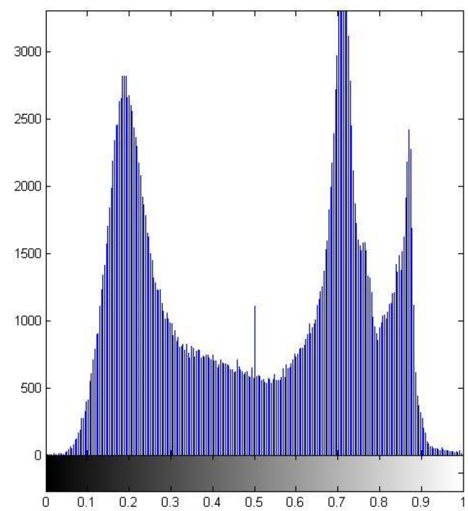
On the other hand, by using two thresholds to detect strong and weak edges, the Canny method is less likely than the other methods to be fooled by the noise, and more likely to detect true weak edge [2]. Figure 12(c)&(d) show more blurring artifacts and less details than these in Figure (a)&(b). This is due to a trade-off between extra weak edge sharpening ability and blurring artifacts.





(c) Laplacian with lamda = 50





(d) Canny with lamda = 40

Figure 12: Edge Sharpened “Boat” images using four methods.

4. Conclusion

In this computer assignment, I study at least five methods to enhance the quality of different images and perform experiments for each of them. The histogram equalization works well on these images with wide dynamic range of gray scales while it is bad at handling specific effect on the image. This shortcoming can be fixed well by using specified histogram specification on the image.

The spatial Low-Pass filtering is an effective method for removing additive Gaussian noise from noisy images. The choice of the filter coefficients and window size present a trade-off between noise removal ability and edge smearing artifacts.

The median filtering is a nonlinear filtering process mainly focused on removing impulsive or "salt & pepper" type noise. Similar with the spatial filter, the choice of larger size window will lead to less noise samples but more blurring artifacts.

The Sobel operator for edge sharpening can perform better than the other methods in terms of edge details remained in the image, but it is not good at weak edge sharpening. The Canny method can find weak edges well by using two separate thresholds.

5. Reference

[1]. Lecture 19-20, ECE513 Digital Image Processing, CSU, SP17.
[2]. https://www.mathworks.com/help/images/ref/edge.html