Beykoz University

Department of "Computer Engineering"

"Image Processing - 7061MEEOS-CME0162"

Project II - Fall Semester

- Report -

Lecturer: ENVER AKBACAK

Leyla Abdullayeva - 1904010038

Due date: 16.12.2022

CONTENTS

Abstract

In this project, my main purpose is to write MATLAB programs and doing histogram equalization exercises,writing histogram equalization transforming functions, implementing high boost filtering by given functions and sumbasks, implementing Laplacian for image sharpening process by given function and given submask, developing a program for 3x3 median filtering process and applying to an image.

I have briefly shared the definitions of the operations performed below.

Histogram equalization is a method in image processing of contrast adjustment using the image histogram.

The high boost filter is simply the sharpening operator in image processing. High boost filter is a popular method that allows sharpening in high detail areas but little or no sharpening in flat or smooth areas.

The Laplacian filter detects sudden intensity transitions in the image and highlights the edges.

Median filtering is a nonlinear method used to remove noise from images. It is widely used as it is very effective at removing noise while preserving edges.
The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image).

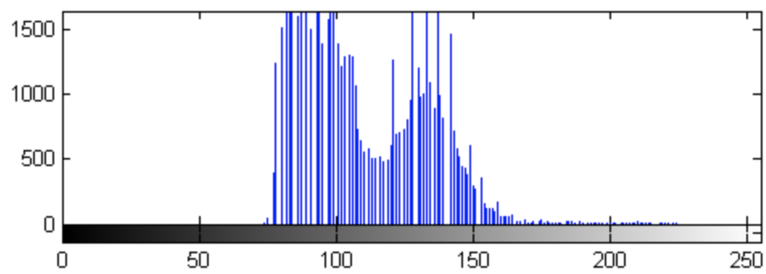
Technical Presentation

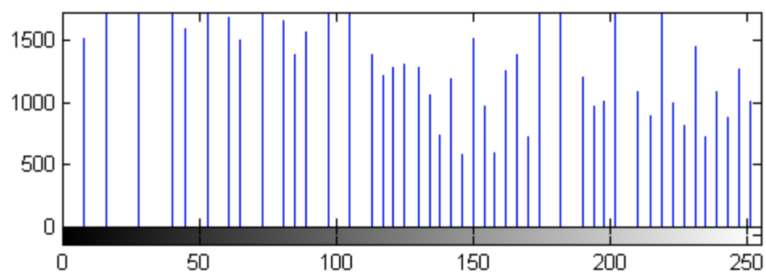
- Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. Histogram equalization, transformation function is calculated using cumulative frequency approach and this process is automatic.

Algorithm

The Histogram Equalization algorithm enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image is approximately flat.

```
I = imread('Phoenix.PNG');
J = histeq(I);
subplot(2,2,1);
imshow( I );
subplot(2,2,2);
imhist(I)
subplot(2,2,3);
imshow( J );
subplot(2,2,4);
imhist(J)
```

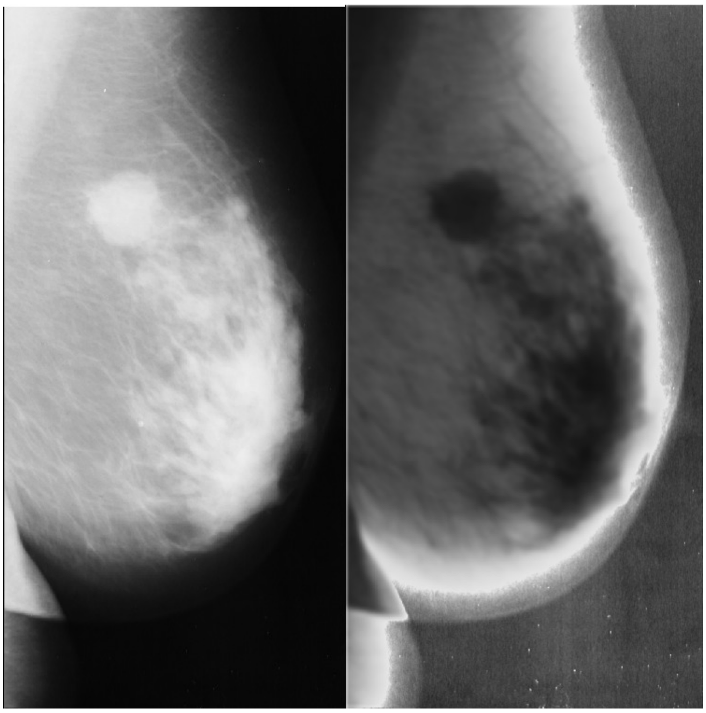




- In image processing, it is often desirable to emphasize high frequency components representing the image details without eliminating low frequency components. The high boost filter can be used to enhance the high frequency components. We can sharpen edges of a image through the amplification and obtain a more clear image. The high boost filter is simply the sharpening operator in image processing. High boost filter is a popular method that allows sharpening in high detail areas but little or no sharpening in flat or smooth areas. A high boost filter is used to retain some of the low-frequency components to and in the interpretation of a image. In high boost filtering the input image f(m,n) is multiplied by an amplification factor A before subtracting the low pass image are discuss as follows:

Example code:

```
High boost filter = A × f(m,n) - low pass filter
Adding and subtracting 1 with the amplification factor
High boost filter = (A-1) × f(m,n) + f(m,n) - low pass filter
But f(m,n) - low pass filter = high pass filter
High boost filter = (A-1) × f(m,n) + high pass filter
High boost filter = A × f(m,n) - low pass filter
Implementation of the technique for color images and grayscale images is outright.
h_size = input('Enter the Size of filter : ');
Sigma = input('Enter the value of Sigma : ');
A = input('Enter the value A : ');
High_Boost_Filt(h_size,Sigma,A);
```



- Image sharpening is an effect applied to digital images to give them a sharper appearance. Sharpening enhances the definition of edges in an image. The dull images are those which are poor at the edges. There is not much difference in background and edges. On the contrary, the sharpened image is that in which the edges are clearly distinguishable by the viewer. We know that intensity and contrast change at the edge. If this change is significant then the image is said to be sharp. The viewer can clearly see the background and foreground parts. A Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change. This determines if a change in adjacent pixel values is from an edge or continuous progression.

- Median filter is one of the well-known order-statistic filters due to its good performance for some specific noise types such as "Gaussian," "random," and "salt and pepper" noises. According to the median filter, the center pixel of a $M \times M$ neighborhood is replaced by the median value of the corresponding window. Note that noise pixels are considered to be very different from the median. Using this idea, a median filter can remove this type of noise problem. We use this filter to remove the noise pixels on the protein crystal images before binarization operation.

Discussion of Findings, Appendix and Results

Question 1.

Write a computer program for implementing the histogram equalization technique. Perform histogram equalization on image “Phoenix.PNG”. As a minimum, your report should include the original image, a plot of its histogram, a plot of the histogram-equalization transformation function, the enhanced image, and a plot of its histogram. Use this information to explain why the resulting image is enhanced.

```
clc
clear all
close all
a=imread('Phoenix.PNG');
b=rgb2gray(a);
subplot(2,2,1);
imshow(b);
title('Original Grayscale Image');
subplot(2,2,3);
imhist(b);
title('Histogram of Original Grayscale Image');
j=histeq(b);
subplot(2,2,2);
imshow(j);
title('Image after histogram equalization');%%Figure 1.Histogram Equalization
subplot(2,2,4);
imhist(j);
title('Histogram of image after histogram equalization');
```

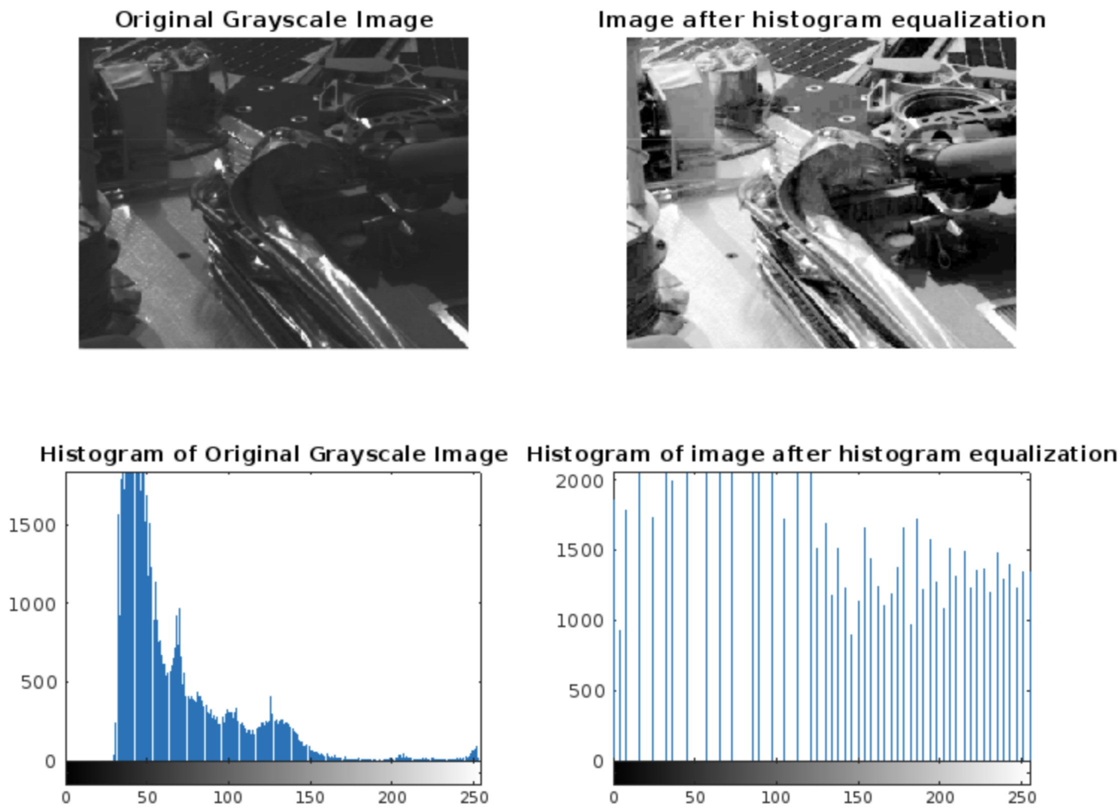


Figure 1: Histogram Equalization

Question 2.

Write a computer program to implement high-boost filtering given by

$$g(x,y) = f(x,y) + k * [f(x,y) - \bar{f}(x,y)]$$

where g(x,y) is the filtered image, k is a constant greater than 1, f(x,y) is the image to be filtered, and f̄(x,y) denote the blurred image obtained by averaging f(x,y). The averaging part of the process should be done by using the following mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Enhance image “softstone.PNG” by using the program you developed.

```
img = imread('softstone.PNG');
% create gaussian filter
h = fspecial('gaussian',5,2.5);
% blur the image
blurred_img = imfilter(img,h);
% subtract blurred image from original
diff_img = img - blurred_img;
% add difference to the original image
highboost_img = img + 3*diff_img;
subplot 221
imshow(img,[]);
title('Original Image')
subplot 222
imshow(blurred_img,[]);
title('Blurred Image')
subplot 223
imshow(diff_img,[]);
title('Difference Image')
subplot 224
imshow(highboost_img,[]);
title('HighBoosted Image')
```

Output:

Figure 2: High Boost filtering

Question 3.

The basic way in which we use the Laplacian for image sharpening is

$$g(x, y) = f(x, y) - [\nabla^{\wedge}2 \, f(x, y)]$$

where f(x,y) and g(x,y) are the input and sharpened images, respectively. Write a program to perform image sharpening. The Laplacian part of the process can be be done by using the following mask

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpen image "moon.tif" by using the program you developed.

```
clc;
close all;
a = im2double(imread('moon.tif')); %// Read in your image
lap = [-1 -1 -1; -1 8 -1; -1 -1 -1]; %// Change - Centre is now positive
resp = imfilter(a, lap, 'conv'); %// Change
%// Change - Normalize the response image
minR = min(resp(:));
maxR = max(resp(:));
resp = (resp - minR) / (maxR - minR);
%// Change - Adding to original image now
sharpened = a + resp;
%// Change - Normalize the sharpened result
minA = min(sharpened(:));
maxA = max(sharpened(:));
sharpened = (sharpened - minA) / (maxA - minA);
%// Change - Perform linear contrast enhancement
sharpened = imadjust(sharpened, [60/255 200/255], [0 1]);
figure;
subplot(1,3,1);imshow(a); title('Original image');
subplot(1,3,2);imshow(resp); title('Laplacian filtered image');
subplot(1,3,3);imshow(sharpened); title('Sharpened image');
```

Output:

Figure 3: Image Sharpening by Laplacian filter

Question 4.

Develop a program to perform 3×3 median filtering. Apply median filtering to the image "noisyImage.PNG", and enhance it.

```
%% Median Filter
%READ IMAGE
sad1=imread('noisyImage.PNG');
figure,imshow(sad1)
title('IMAGE WITH NOISE');
median_filt=zeros(size(sad1)+2);
re_sad1=zeros(size(sad1));
       for x=1:size(sad1,1)
           for y=1:size(sad1,2)
               median_filt(x+1,y+1)=sad1(x,y);
           end
       end

%LET THE WINDOW BE AN ARRAY
%STORE THE 3-by-3 NEIGHBOUR VALUES IN THE ARRAY
%SORT AND FIND THE MIDDLE ELEMENT
for i= 1:size(median_filt,1)-2
   for j=1:size(median_filt,2)-2
       window=zeros(9,1);
       inc=1;
       for x=1:3
           for y=1:3
               window(inc)=median_filt(i+x-1,j+y-1);
               inc=inc+1;
           end
       end

       med=sort(window);
       re_sad1(i,j)=med(5);

   end
end
re_sad1=uint8(re_sad1);
title('IMAGE AFTER MEDIAN FILTERING');
figure,imshow(re_sad1)

Output:
```

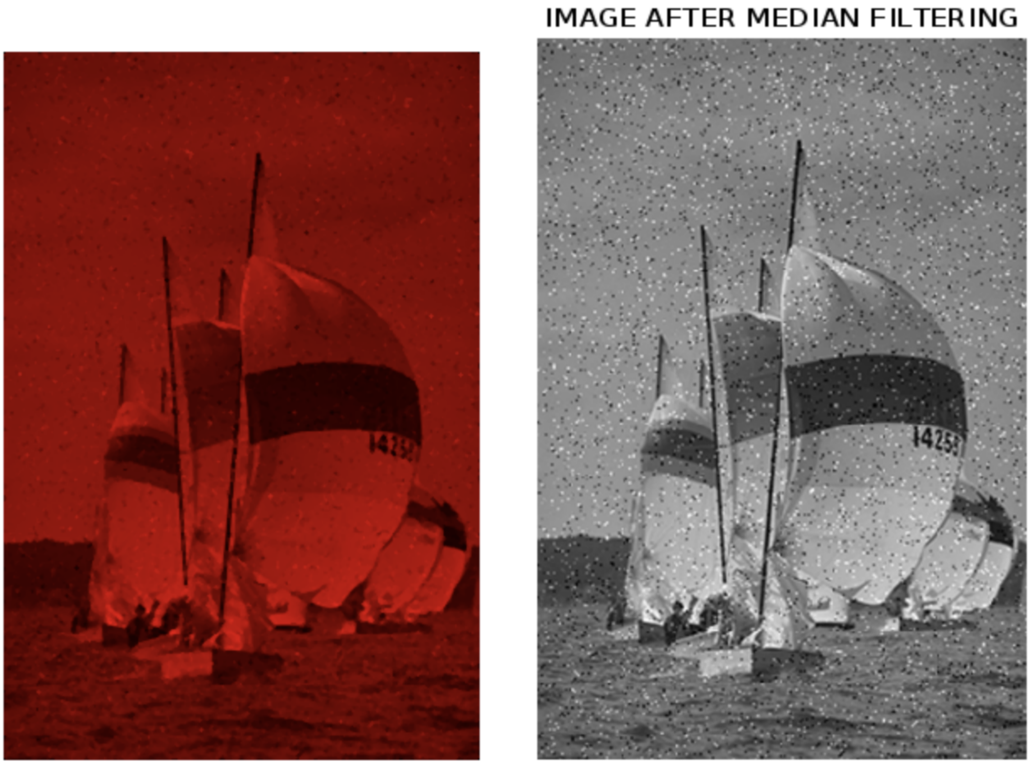


Figure 4: 3x3 Median Filtering application

Table of Figures

References

https://www.mathworks.com/

https://www.geeksforgeeks.org/noise-removal-using-median-filter-in-c/