EECE-5626 (IP&PR) : Homework-2

Due on October 8, 2021 by 11:59 pm via submission portal.

NAME: McKean, Tyler

Instructions:

1. You are required to complete this assignment using Live Editor.
2. Enter your MATLAB script in the spaces provided. If it contains a plot, the plot will be displayed after the script.
3. All your plots must be properly labeled and should have appropriate titles to get full credit.
4. Use the equation editor to typeset mathematical material such as variables, equations, etc.
5. After completeing this assignment, export this Live script to PDF and submit the PDF file through the provided submission portal.
6. You will have two attempts to submit your assignment. However, make every effort to submit the correct and completed PDF file the first time. If you use the second attempt, then that submission will be graded.
7. Your submission of problem solutions must be in the given order, i.e., P1, P2, P3, etc. Do not submit in a random order.
8. Please submit your homework before the due date/time. A late submission after midnight of the due date will result in loss of points at a rate of 10% per hour until 8 am the following day, at which time the solutions will be published.

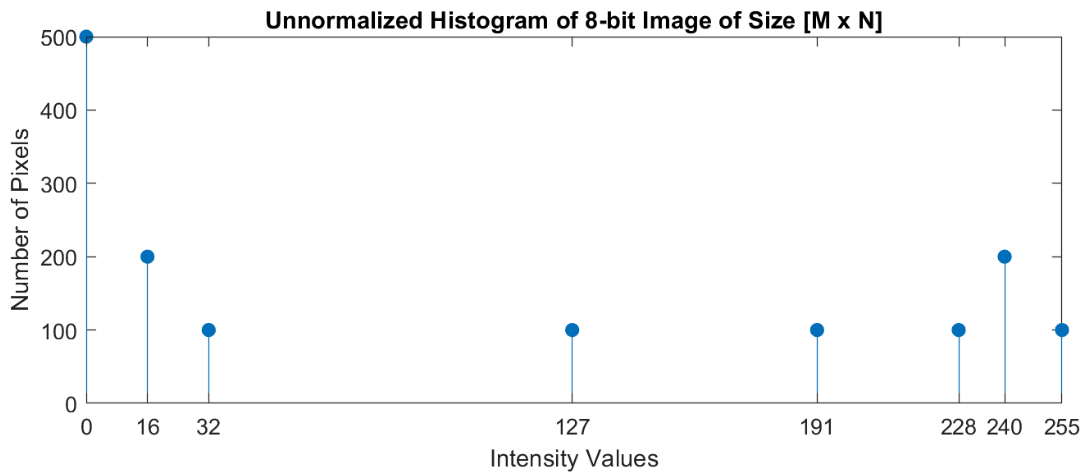
Reading Assignment: Chapters 3 from DIP4E and DIPUM3E.

Table of Contents

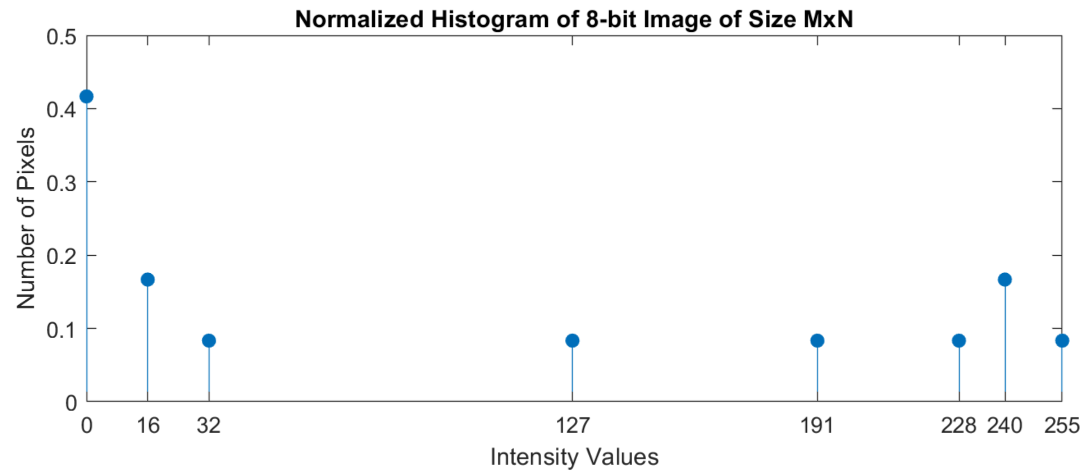
Problem-1: DIP4E Problem 3.7 (Page 240)

Solution:

```
clear all
close all

% Declare value for M,N
M = 30;
N = 40;
% Declare the intensity values that match the amount of pixels from the
% image of P3.7 in DIP4E
r0 = (N/2)*(M*(2/3)) + (1/2)*(N/2)*(M/3);
r16 = (M*(2/3))*(N/4);
r32 = (1/2)*(N/2)*(M/3);
r127 = (M/3)*(N/4);
r191 = (M/3)*(N/4);
r228 = (1/2)*(M*(2/3))*(N/4);
r240 = (M*(2/3))*(N/4);
r255 = (1/2)*(M*(2/3))*(N/4);
y = [r0,r16,r32,r127,r191,r228,r240,r255];
x = [0,16,32,127,191,228,240,255];
% Plot the Unnormalized Histogram of 8-bit Image
figure('Units','inches','Position',[0,0,8,3])
stem(x,y,"filled")
title('Unnormalized Histogram of 8-bit Image of Size [M x N]')
xlabel('Intensity Values')
ylabel('Number of Pixels')
xlim([0 255])
ylim([0 max(y)])
xticks([0 16 32 127 191 228 240 255])
```




```
% Plot the Normalized Histogram of 8-bit Image of Size [M x N]
yNorm = y/(M*N);
figure('Units','inches','Position',[0,0,8,3])
stem(x,yNorm,"filled")
title('Normalized Histogram of 8-bit Image of Size MxN')
xlabel('Intensity Values')
ylabel('Number of Pixels')
xlim([0 255])
ylim([0 0.5])
xticks([0 16 32 127 191 228 240 255])
```

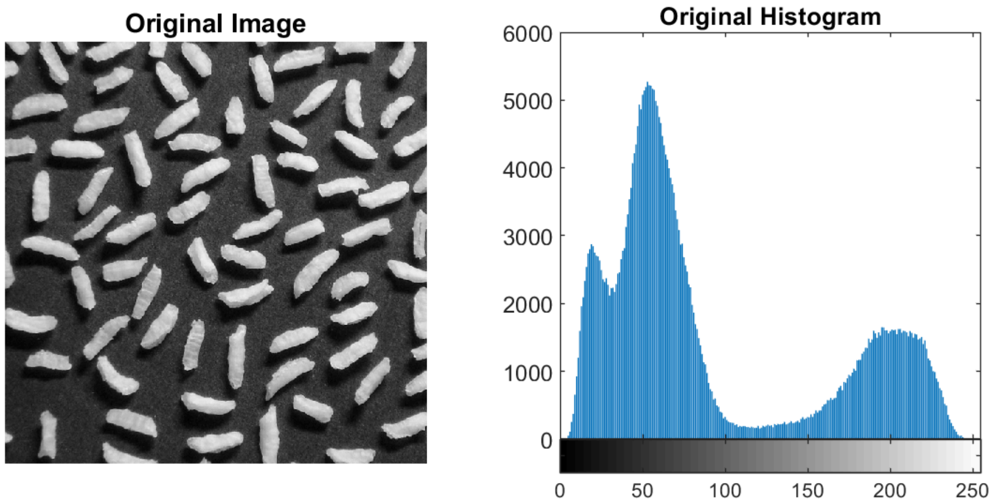


Problem-2: DIP4E Problem 3.9 (Page 240)

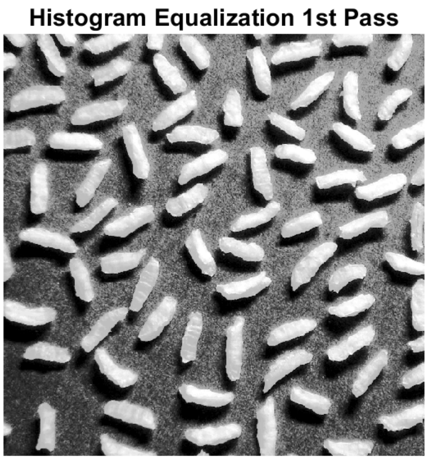
Solution:

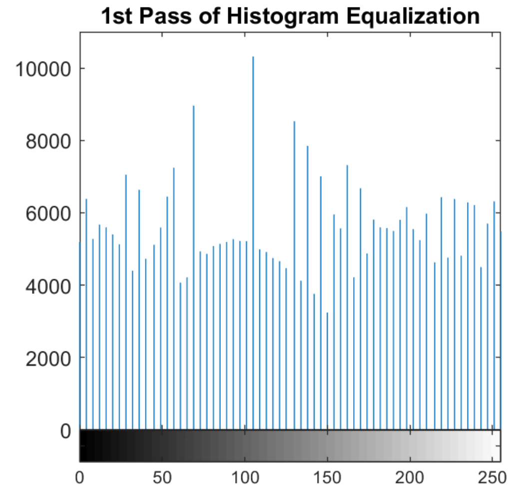
```
% Show that a second pass of Histogram Equalization (on the
% histogram-equalized image) will produce exactly the same result as the
% first pass

f = imread('rice.tif');
[N1,N2] = size(f); NoP = numel(f); % Number of Pixels
```

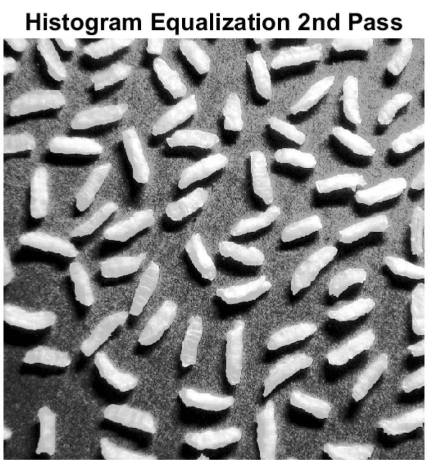
```
imghist = imhist(f); imghistmax = ceil(max(imghist(:))/1000)*1000;
figure('Units','inches','Position',[0,0,7,3]);
subplot(1,2,1); imshow(f); title('Original Image','fontsize',14);
subplot(1,2,2); imhist(f); set(gca,'ylim',[0,imghistmax]);
title('Original Histogram','fontsize',14);
```

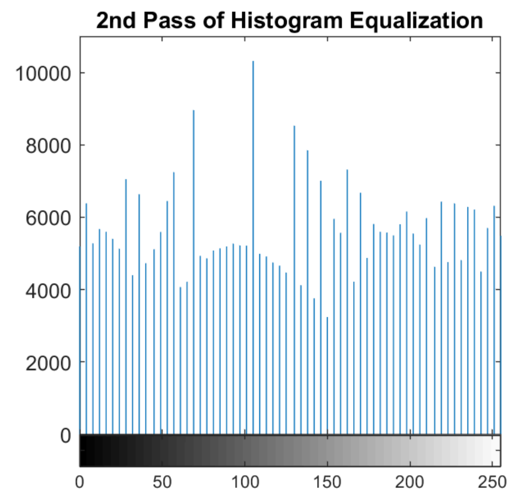



```
fHisteq = histeq(f);
figure('Units','inches','Position',[0,0,8,8]);
subplot(2,2,1); imshow(fHisteq); title('Histogram Equalization 1st Pass','fontsize',12);
imghist = imhist(fHisteq); imghistmax = ceil(max(imghist(:))/1000)*1000;
subplot(2,2,2); imhist(fHisteq); set(gca,'ylim',[0,imghistmax]);
title('1st Pass of Histogram Equalization','fontsize',12);
fHisteq2 = histeq(fHisteq);
subplot(2,2,3); imshow(fHisteq2); title('Histogram Equalization 2nd Pass','fontsize',12);
imghist = imhist(fHisteq2); imghistmax = ceil(max(imghist(:))/1000)*1000;
subplot(2,2,4); imhist(fHisteq2); set(gca,'ylim',[0,imghistmax]);
title('2nd Pass of Histogram Equalization','fontsize',12);
```










```
if(fHisteq - fHisteq2 == 0)
    disp('1st and 2nd Histogram Equalizations are Equivalent!')
else
    disp('Histograms are not equivalent!')
end
```

```
1st and 2nd Histogram Equalizations are Equivalent!
```

Here, by running the sample image through two passes of Histogram Equalization, you can see the Histogram and altered images look identical. When performing a conditional check by subtracting the intensity values of the first Histogram Equalization by the second, the result is equal to zero. Thus, the intensity values of a second Histogram Equalization is equivalent to a first pass of Histogram Equalization.

Problem-3: DIP4E Problem 3.13 (Page 240)

(a) Histogram Equalization

Solution:

The histogram equalization transformation $T(r)$ for the interval $[0, L-1]$ would be:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = \frac{2}{L-1}\int_0^r (w)dw = \frac{r^2}{L-1}$$

Thus, squaring the values of the input intensities and dividing by $(L-1)$ results in an image with intensities, s, that have a uniform PDF due to this histogram-equalization transformation.

(b) Histogram Specification

Solution:

The transformation function whose PDF is $p_z(z) = \frac{3z^2}{(L-1)^3}$ for $0 \leq z \leq L-1$ and $p_z(z) = 0$ for other values of z is as follows:

$$T(z) = (L-1)\int_0^z p_z(w)dw = \frac{3}{(L-1)^2}\int_0^z (w^2)dw = \frac{z^3}{(L-1)^2}$$

we then require $T(z) = s$, so we set $s = \frac{z^3}{(L-1)^2}$ and solve for z which results in:

$$z = T^{-1}(s) = \sqrt[3]{s(L-1)^2}$$

Thus the result will be an image whose intensities, z, will have the desired PDF in the interval $[0, L-1]$.

(c) Overall Transformation

Solution:

From part a.) $s = T(r) = \frac{r^2}{L-1}$ and from Part b.) $s = \frac{z^3}{(L-1)^2}$ then setting substituting s and solving for r we get:

$$\frac{r^2}{(L-1)} = \frac{z^3}{(L-1)^2} \Rightarrow z^3 = r^2(L-1) \Rightarrow z = \sqrt[3]{r^2(L-1)}$$

whose histogram-equalized intensities would equal to the input image intensities, r, times $(L-1)$, all cube rooted.

Problem-4: DIP4E Problem 3.36 (Page 243)

Note: I have modified the statement of this problem as follows:

An image is filtered with three Gaussian lowpass kernels of sizes 9×9, 13×13, and 25×25, and standard deviations 1.5, 2, and 4, respectively. A composite filter, w, is formed as the convolution of these three filters.

(a) Convolution of Gaussian filters

Is the resulting composite filter, w, Gaussian? Explain.

Solution:

Yes, the resulting composite filer, w, will be Gaussian, because of the Commutative and Associative properties of Convolution, you can convolve the three separate Gaussian lowpass kernels in any possible order to get the resulting composite filter, w. Along with this point, one of the fundamental properties of Gaussians are that the convolution of two Gaussians result in Gaussian functions as well.

(b) Standard Deviation of the Composite Filter

What is the standard deviation of the composite filter w?

Solution: The standard deviation σ of the composite filter w would be derived using the formula:

$\sigma_{f \star g} = \sqrt{\sigma_f^2 + \sigma_g^2}$ and inserting the three Gaussian lowpass kernel standard deviation values we get:

$\sigma_{f \star g \star h} = \sqrt{\sigma_f^2 + \sigma_g^2 + \sigma_h^2} = \sqrt{(1.5)^2 + (2)^2 + (4)^2} = \sqrt{22.25} = 4.71699$ or about 4.72

(c) Size of the Composite Filter

What should be the ***appropriate*** size of the composite filter w?

Solution: We must use a size equal to the closest odd integer to $[6\sigma] \times [6\sigma]$ so inserting the value for σ in the previous section we get:

$[(6)(4.72)] \times [(6)(4.72)] = [28.32] \times [28.32]$ or $[29] \times [29]$

Performing the ceiling operation to the closest odd-integer value would mean the appropriate size of the composite filter would be

$[29] \times [29]$

Problem-5: DIP4E Project 3.7 (Page 247)

Lowpass filtering

(a) Blurring of `testpattern1024` image.

Solution:

```
f = imread('testpattern1024.tif');
h = fspecial('gaussian',159,26.5);
g = imfilter(f,h);
figure, imshow(g), title('Blurred testpattern1024 Image'),pause(1)
```

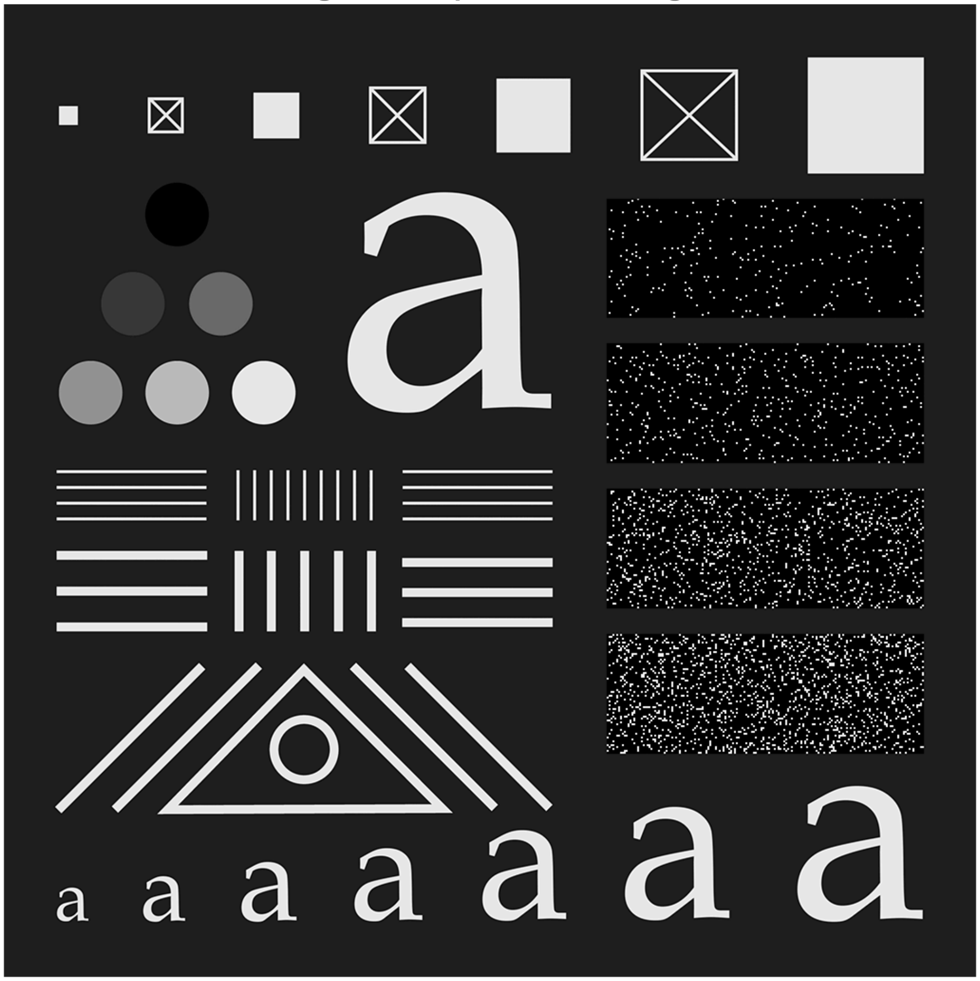
Blurred testpattern1024 Image

(b) Thresholding of `testpattern1024.tif` image.

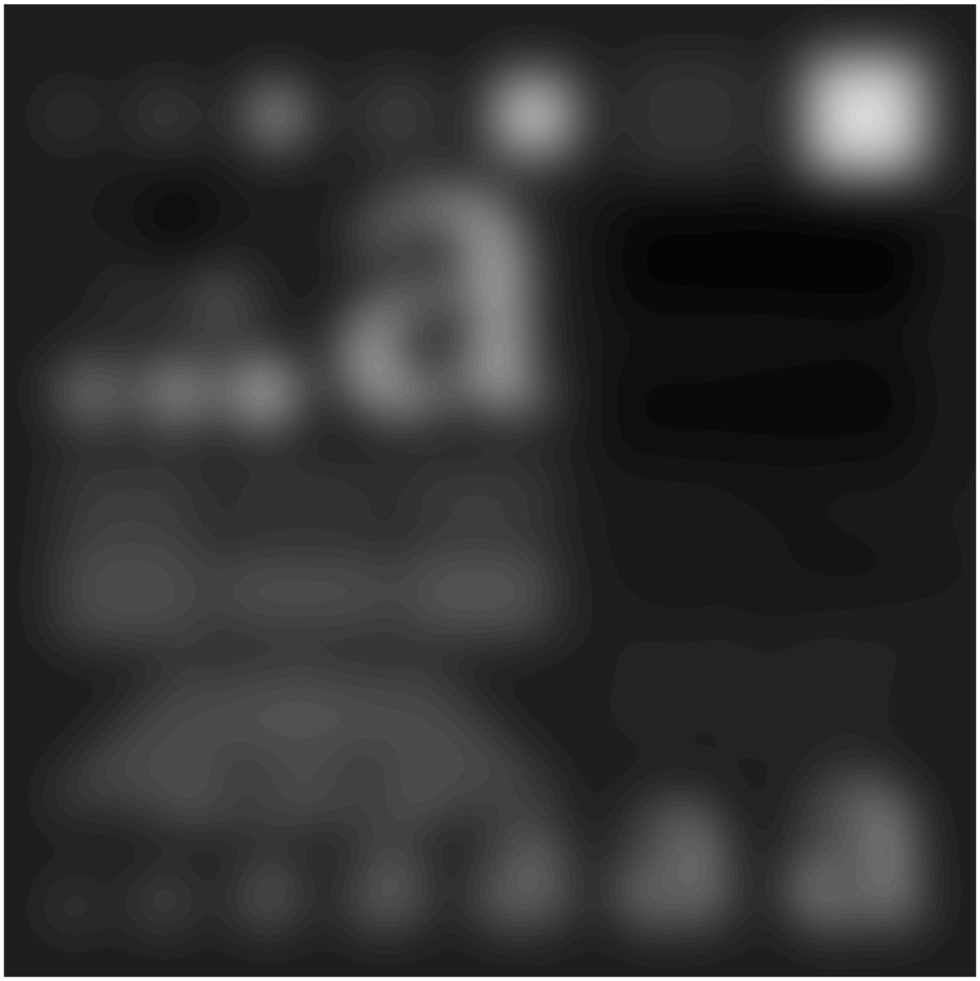
Solution:

```
f = intXform4e(f,'negative');
figure, imshow(f), title('Negative testpattern1024 Image')
```

Negative testpattern1024 Image

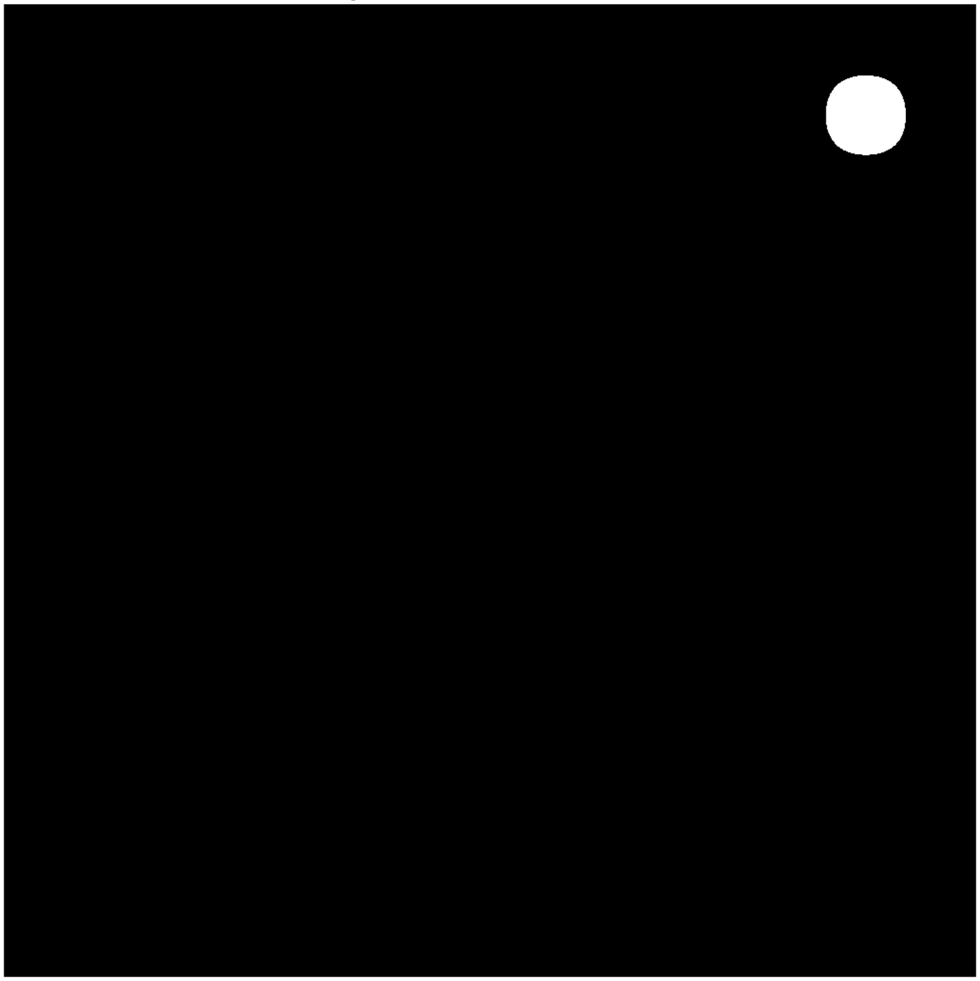


```
gaussFilter = gaussKernel4e(201,30);
g = twodConv4e(f,gaussFilter);
figure, imshow(g)
```



```
gMax = max(g(:));
gThreshold = g > round(gMax,1)*gMax;
figure, imshow(gThreshold), title('testpattern1024 Thresholded')
```

testpattern1024 Thresholded

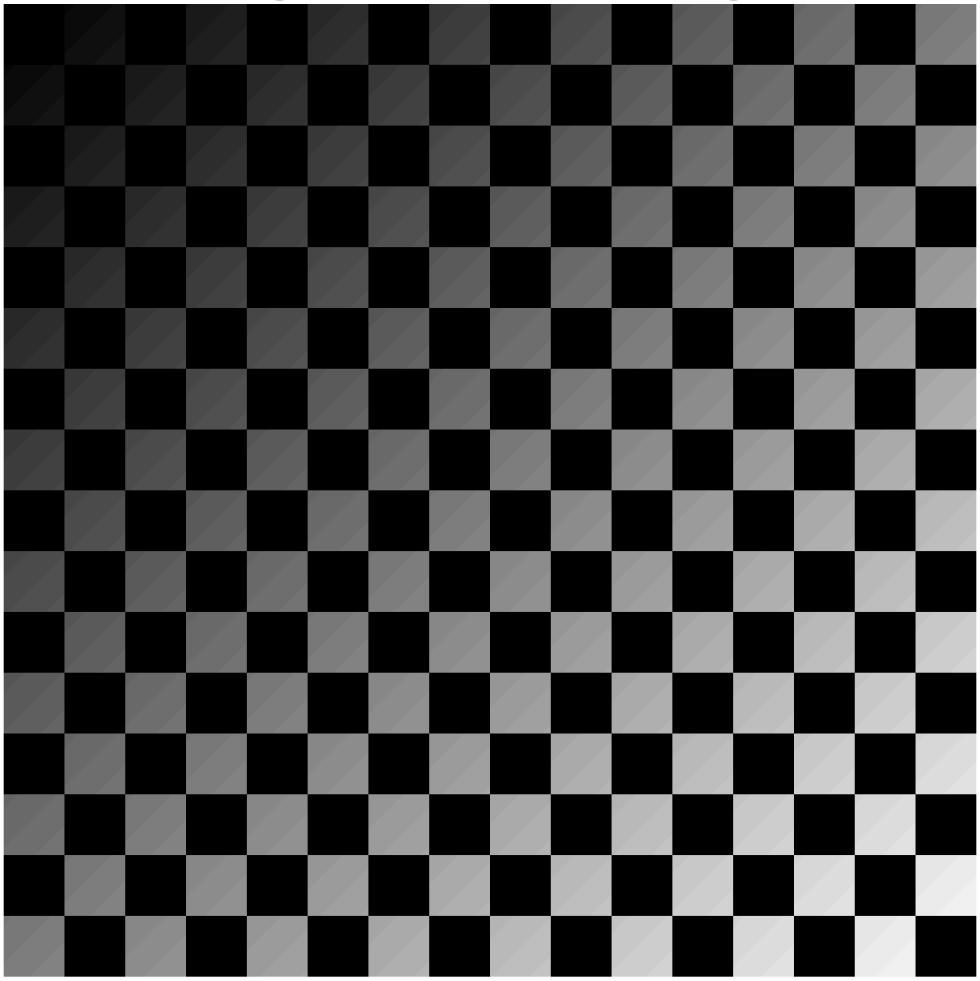


(c) Reproduce Example 3.18.

Solution:

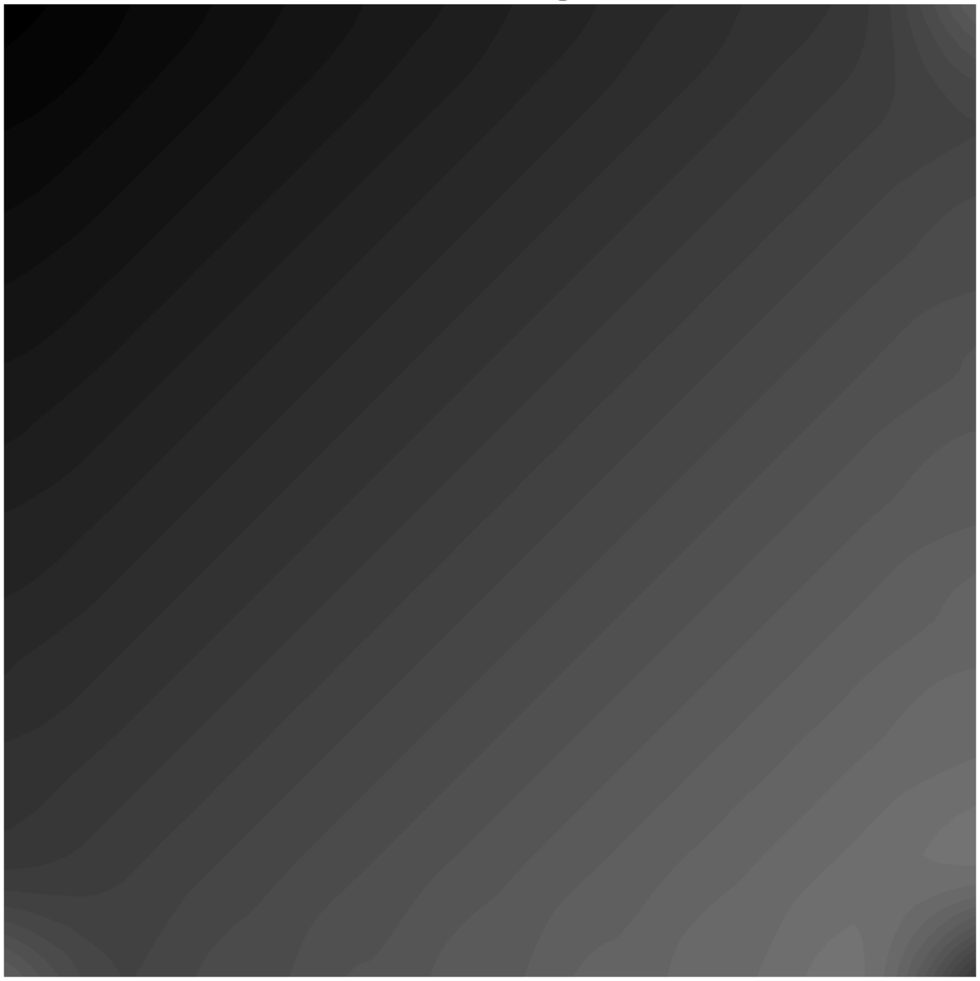
```
f = imread('checkerboard1024-shaded.tif');
f = intScaling4e(f);
figure, imshow(f), title('Integer Scaled Checkerboard1024 Image'),pause(1)
```

Integer Scaled Checkerboard1024 Image

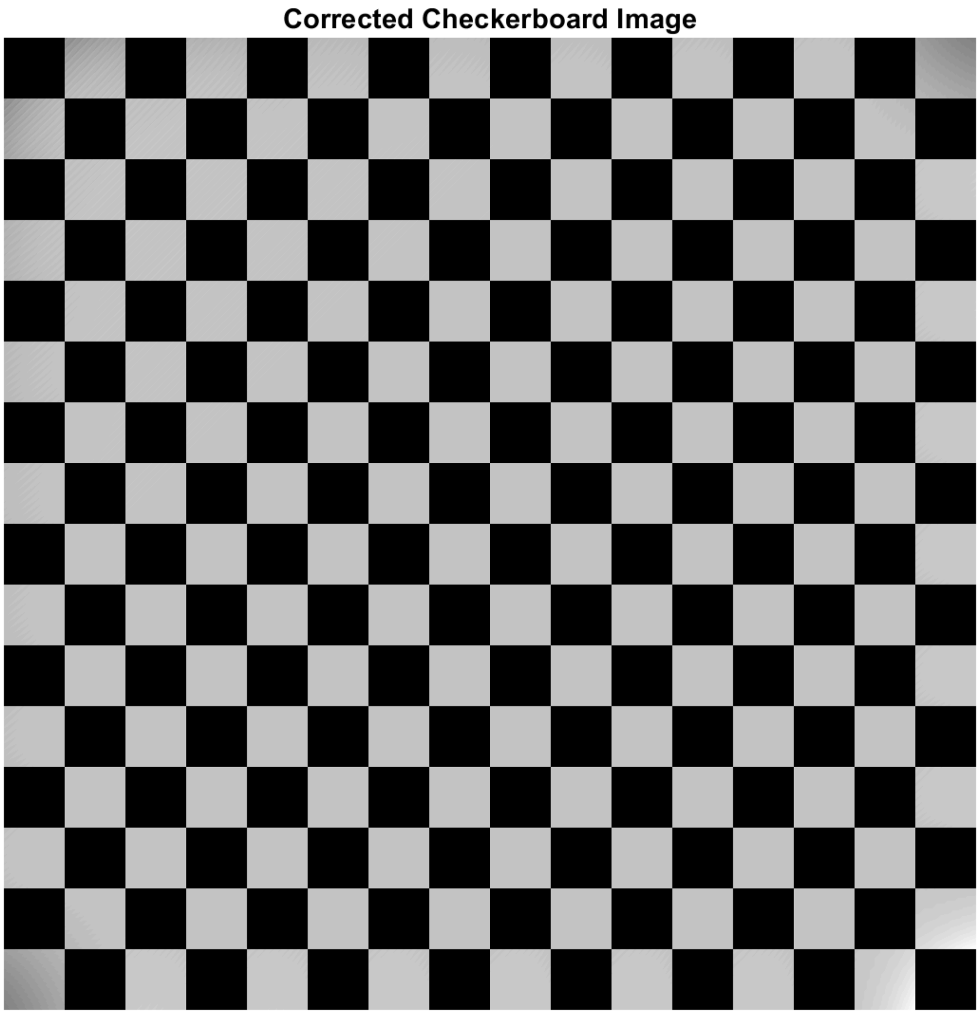


```
h = fspecial('gaussian',256,64);
g = twodConv4e(f,h);
g = intScaling4e(g);
figure, imshow(g), title('Estimate of Shading Pattern'),pause(1)
```

Estimate of Shading Pattern



```
fCorrected = double(f)./g;
fCorrected = intScaling4e(fCorrected);
figure, imshow(fCorrected), title('Corrected Checkerboard Image'),pause(1)
```



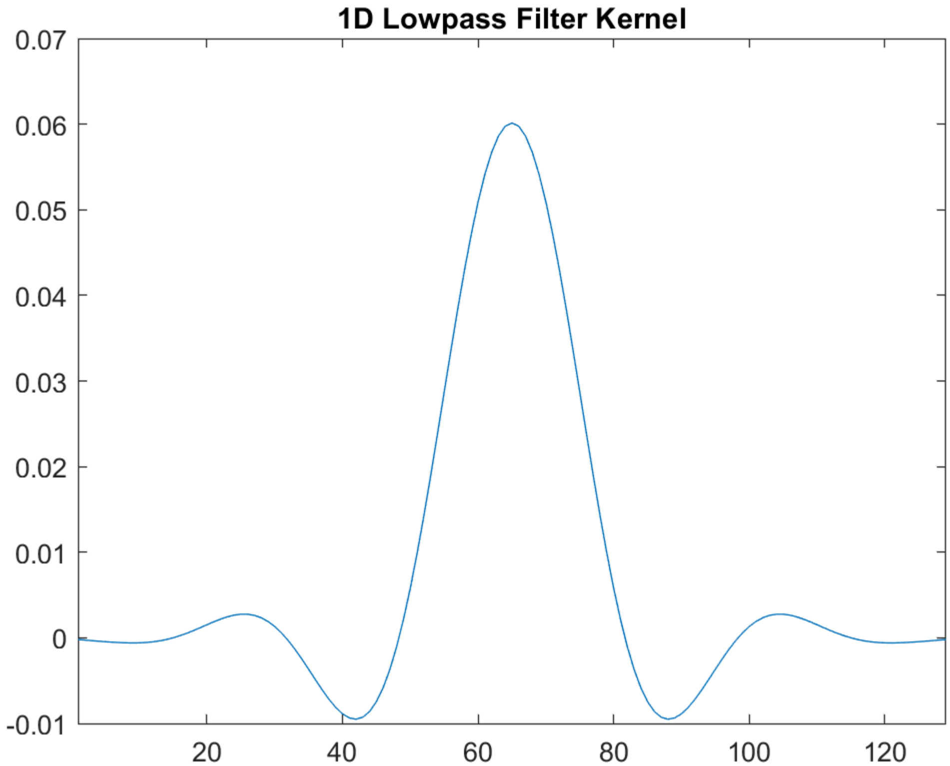
Problem-6: DIP4E Project 3.10 (Page 248)

Using kernels generated by filter design software.

(a) 1-D Lowpass Filter

Solution:

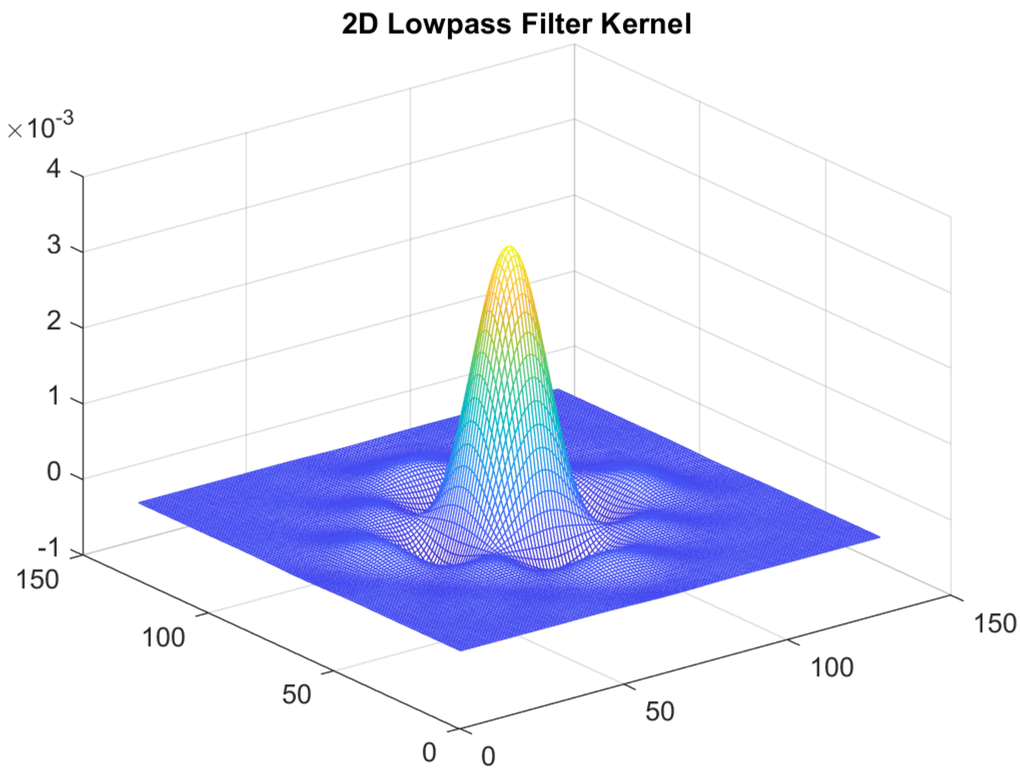
```
lowpassKernel = dlmread('lpkernel1D.txt');
figure,plot(lowpassKernel),title('1D Lowpass Filter Kernel')
xlim([1 129]),pause(1)
```



(b) 2-D Lowpass Filter

Solution:

```
lowpass2D = (lowpassKernel')*lowpassKernel;
figure,mesh(lowpass2D),title('2D Lowpass Filter Kernel'),pause(1)
```



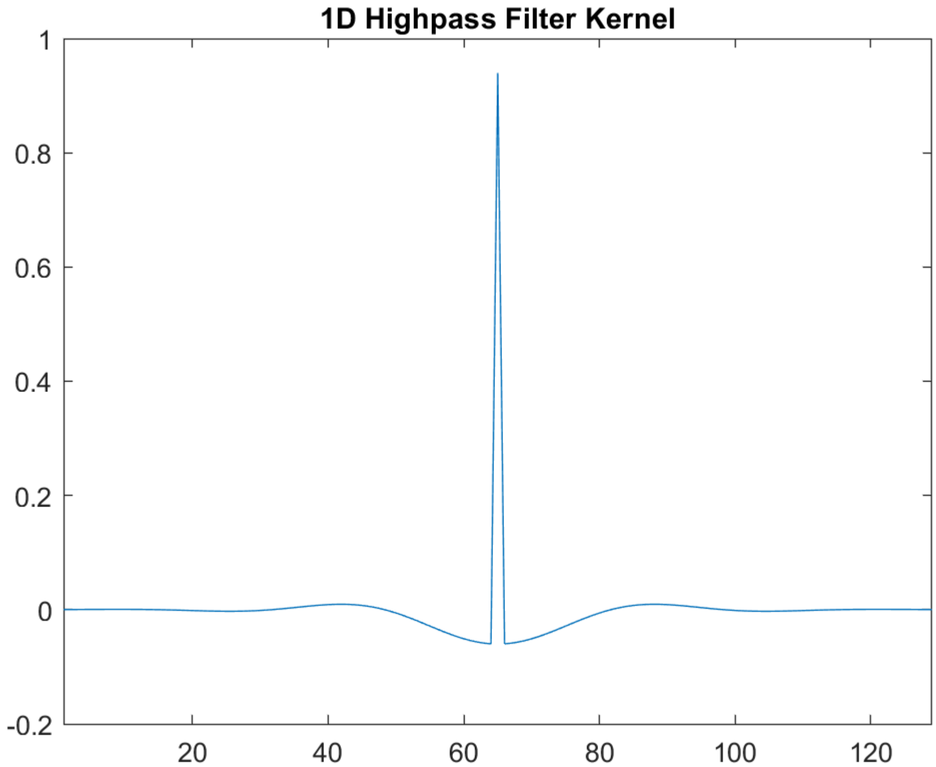

```
f = imread('testpattern1024.tif');
g = twodConv4e(f,lowpass2D);
figure,imshow(g),title('2D Lowpass Filter testpattern1024 Image'),pause(1)
```

2D Lowpass Filter testpattern1024 Image

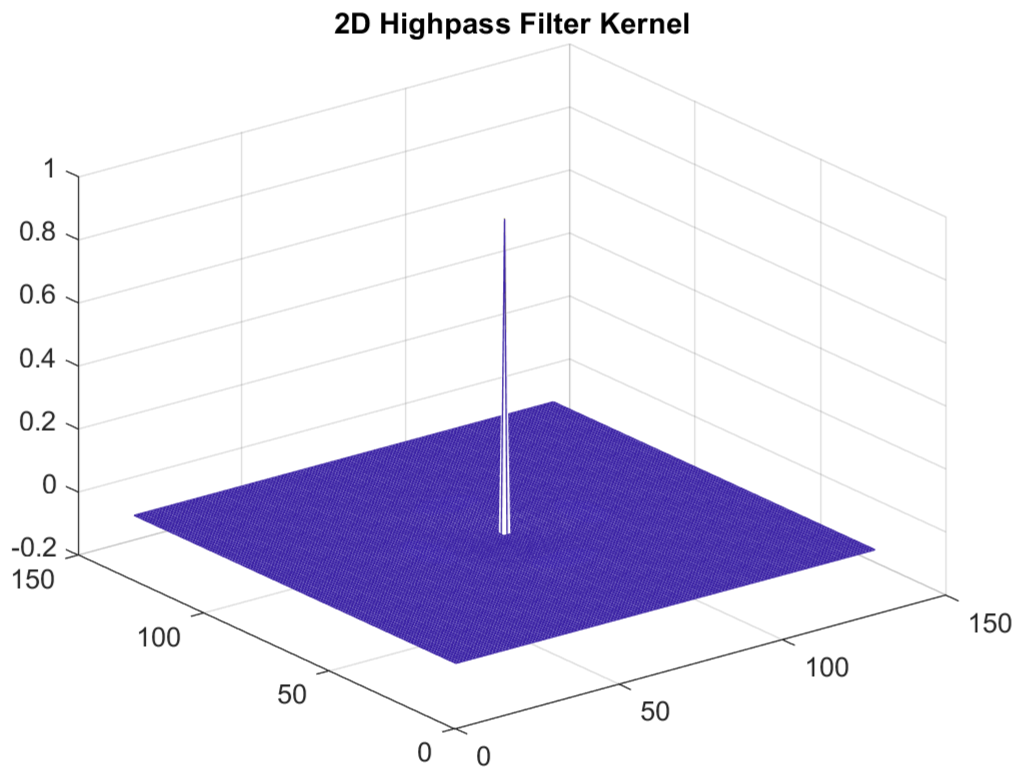
(c) Highpass filtering of `testpattern1024.tif` image.

Solution:

```
Impulse = zeros(1,129);
Impulse(65) = 1;
highpass = Impulse - lowpassKernel;
figure,plot(highpass),title('1D Highpass Filter Kernel')
xlim([1 129]),pause(1)
```

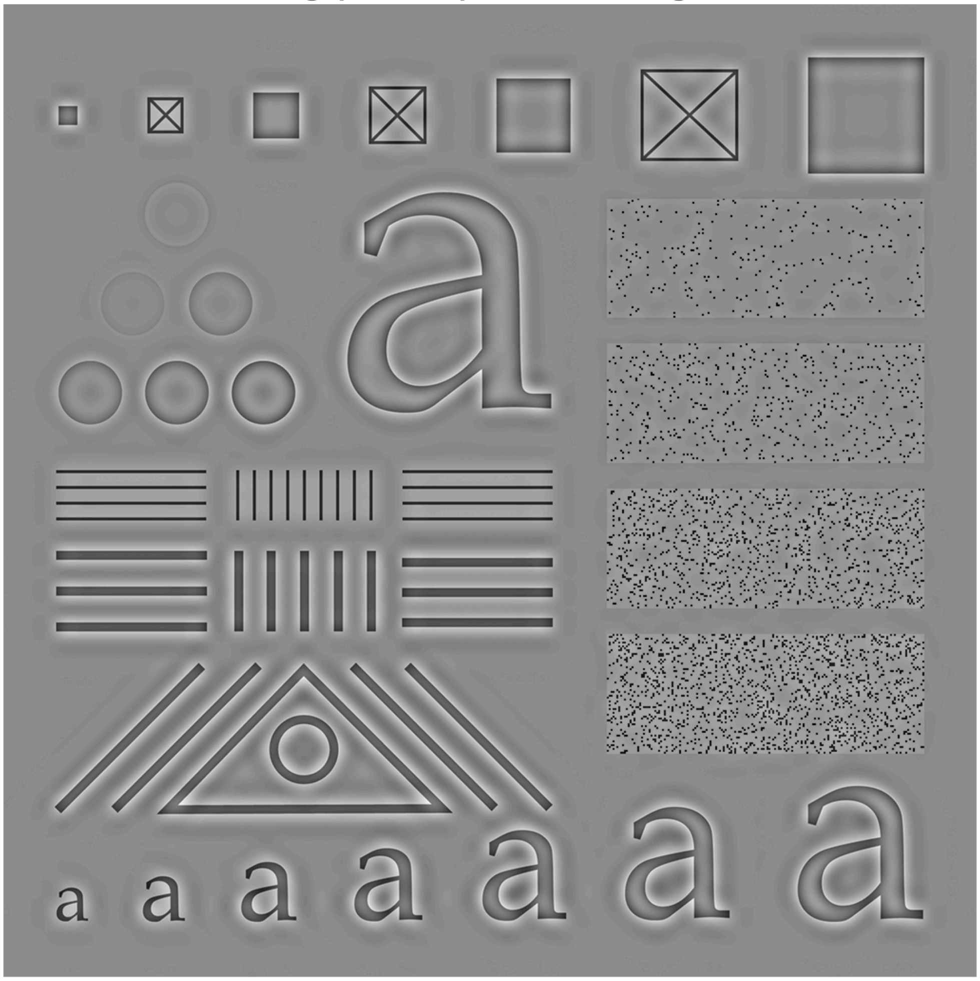



```
Impulse2D = zeros(129,129); Impulse2D(65,65) = 1;
highpass2D = Impulse2D - lowpass2D;
figure, mesh(highpass2D),title('2D Highpass Filter Kernel'),pause(1)
```




```
gHP = twodConv4e(f,highpass2D);
gHP = intScaling4e(gHP);
figure, imshow(gHP), title('Highpass testpattern1024 Image'),pause(1)
```

Highpass testpattern1024 Image

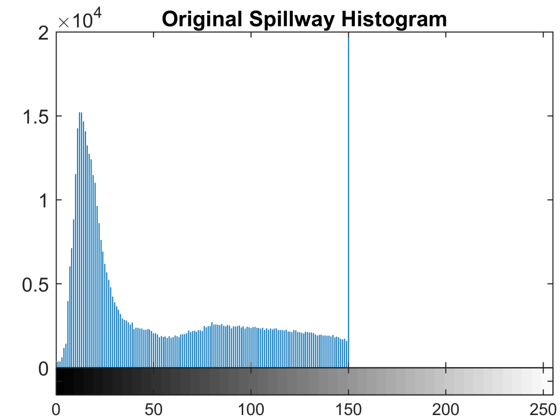


Problem-7: DIPUM3E 3.1 (Page 191)

(a) Log.

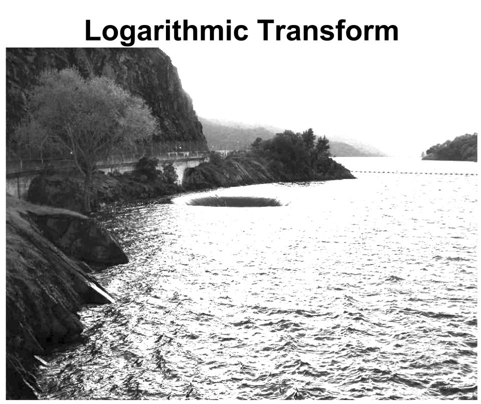
Solution:

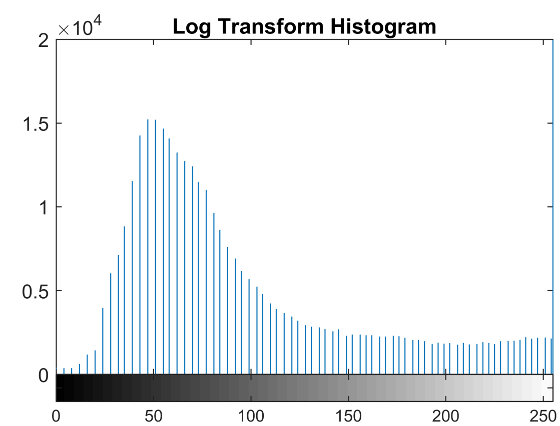
```
f = imread('spillway.tif');
[N1,N2] = size(f); NoP = numel(f); % Number of Pixels
imghist = imhist(f); imghistmax = ceil(max(imghist(:)));
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(f); title('Original Spillway Image','fontsize',14);
subplot(1,2,2); imhist(f); set(gca,'ylim',[0,20000]);
title('Original Spillway Histogram','fontsize',14);pause(1)
```



Above is the original Spillway image with it's Histogram of the intensity values.

```
g1 = intensityTransformations(f,'log',4);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(g1); title('Logarithmic Transform','fontsize',14);
subplot(1,2,2); imhist(g1); set(gca,'ylim',[0,20000]);
title('Log Transform Histogram','fontsize',14);pause(1)
```



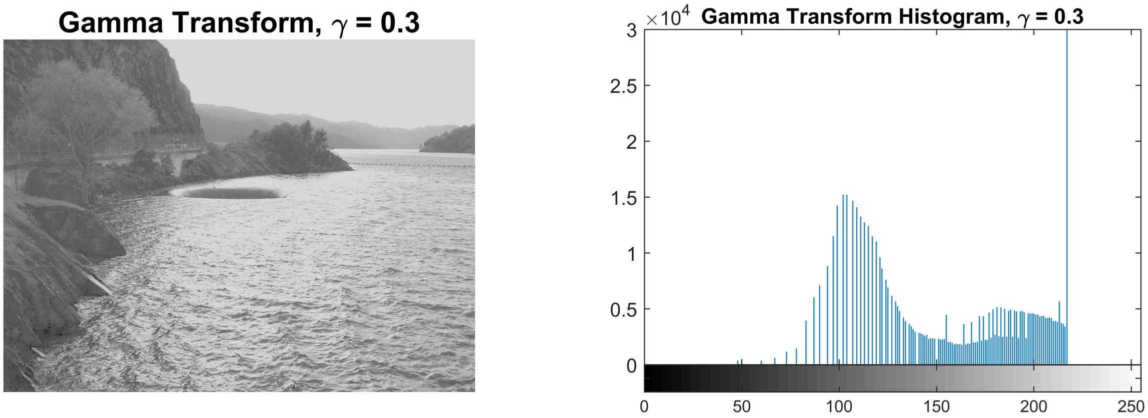


For the logarithmic transformation, the resulting Histogram has a more dynamic range that compresses the values to the full 8-bit intensity range. Compared to the original image, the log transform expanded the intensity values to reach the upper range, whereas before, the highest intensity value was around 150. This operation performed a similar operation to a histogram equalization by spreading the values to cover the full range, but did not redistribute or change the shape of the original histogram. This allows you to now be able to see the coastal road in the image and brightens the water more.

(b) Gamma.

Solution:

```
g2 = intensityTransformations(f,'gamma',0.3);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(g2); title('Gamma Transform, \gamma = 0.3','fontsize',14);
subplot(1,2,2); imhist(g2); set(gca,'ylim',[0,30000]);
title('Gamma Transform Histogram, \gamma = 0.3','fontsize',14);pause(1)
```

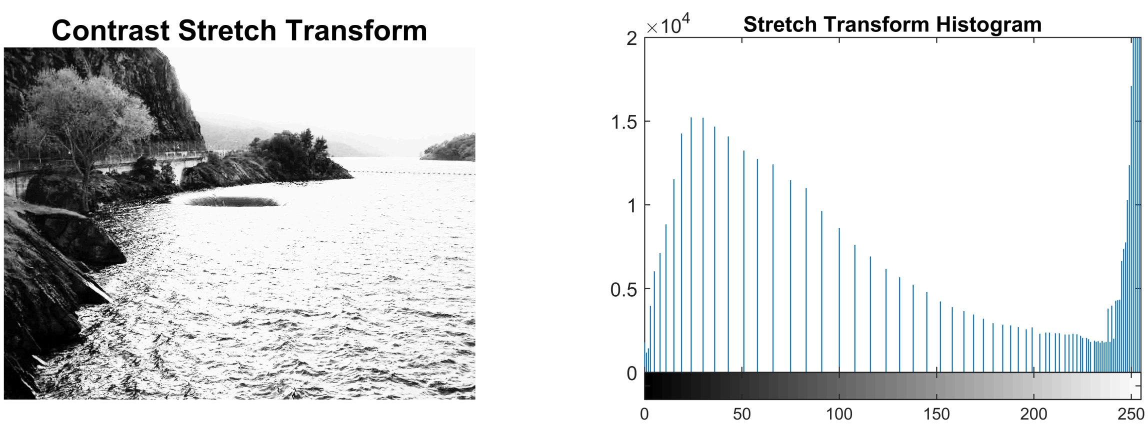


When the Gamma value is less than 1, the intensity transformation has a similar characterisitc to the logarithmic transform, but has a more variable response. The result of using these parameters shifted the original intensity values to have a higher contrast without affecting the original distribution shape of the intensities. This results in the original image having a more grayish tone as it removes all of the black intensities, but did not stretch the values to have brighter white intensitiy values. This effect allows us to see the coastal road that was hidden in the darker intensity values of the original image.

(c) Stretch

Solution:

```
g3 = intensityTransformations(f,'stretch', 0.1, 3);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(g3); title('Contrast Stretch Transform','fontsize',14);
subplot(1,2,2); imhist(g3); set(gca,'ylim',[0,20000]);
title('Stretch Transform Histogram','fontsize',14);pause(1)
```

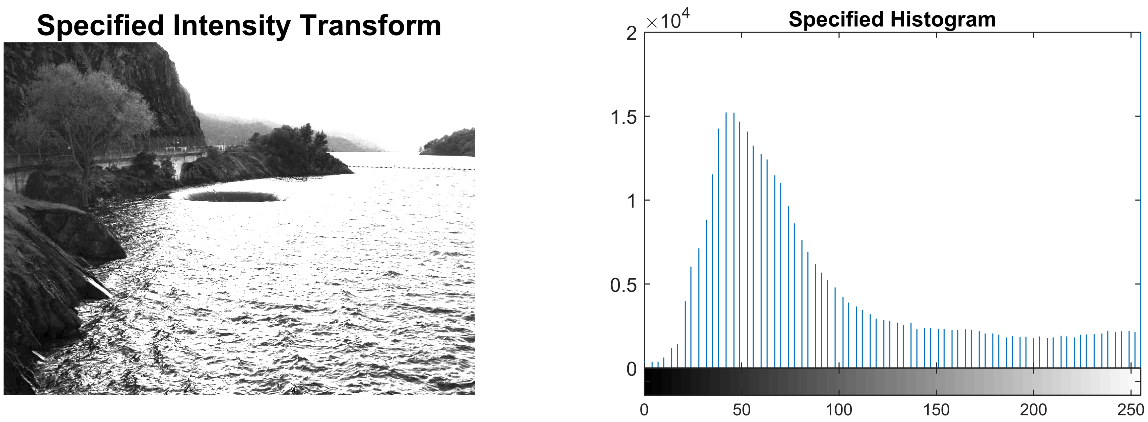


For the Contrast-Stretching Transformation, the resulting image contains a higher contrast than the original. By setting the M value of the intensityTransformations function to 0.1, the darker valued intensities of the original images were stretched to higher values, which gave the resulting image a more concentrated intensity distribution near the upper limit of the 8-bit image. Here, even the coastal road has higher white intensities and parts of the water are so saturated with higher values, they blur into white intensities.

(d) Your specified transformation function.

Solution:

```
r = 3.5.*linspace(0,1,numel(f));
g4 = intensityTransformations(f,'specified',r);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(g4); title('Specified Intensity Transform','fontsize',14);
subplot(1,2,2); imhist(g4); set(gca,'ylim',[0,20000]);
title('Specified Histogram','fontsize',14);pause(1)
```



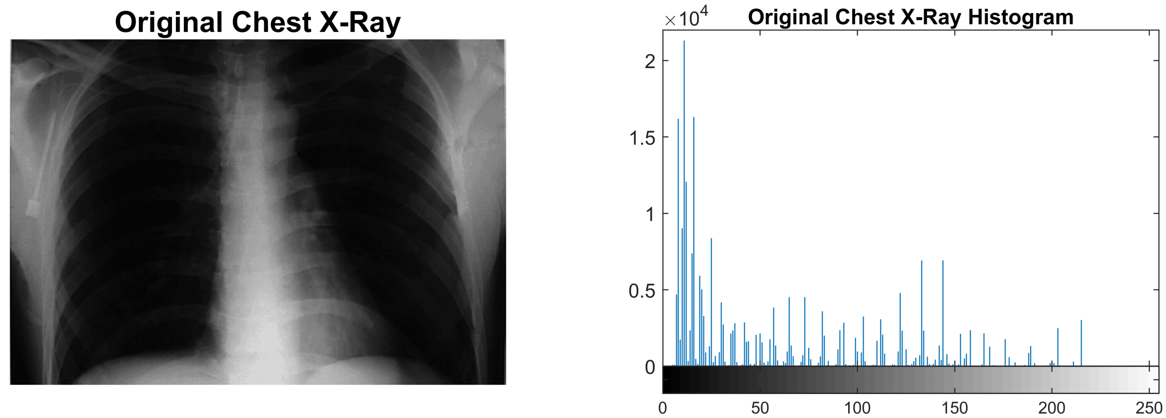
For the Specified Intensity Transformation, I chose to use a linear spaced vector that is 3.5 times the original images intensities. This results in a higher contrast than the original and allows us to view the coastal road.

Problem-8 DIPUM3E 3.5 (Page 193)

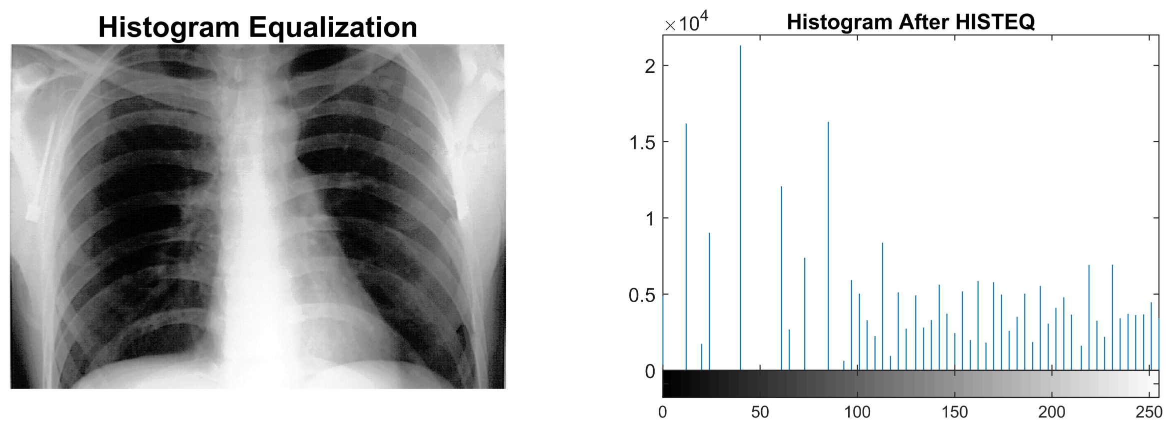
(a) Histogram Equalization

Solution:

```
f = imread('chestXray-dark.tif');
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(f); title('Original Chest X-Ray','fontsize',14);
subplot(1,2,2); imhist(f); set(gca,'ylim',[0,22000]);
title('Original Chest X-Ray Histogram','fontsize',14);pause(1)
```



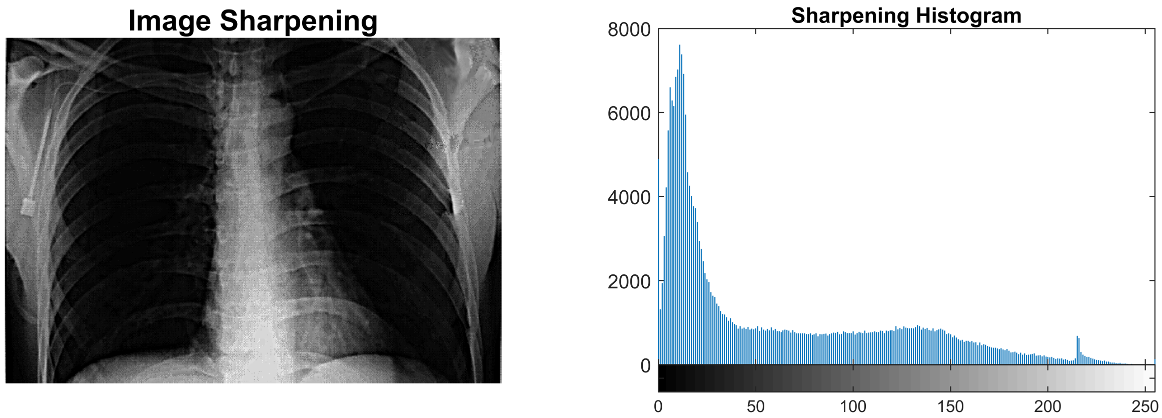

```
feq = histeq(f);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(feq); title('Histogram Equalization','fontsize',14);
subplot(1,2,2); imhist(feq); set(gca,'ylim',[0,22000]);
title('Histogram After HISTEQ','fontsize',14);pause(1)
```



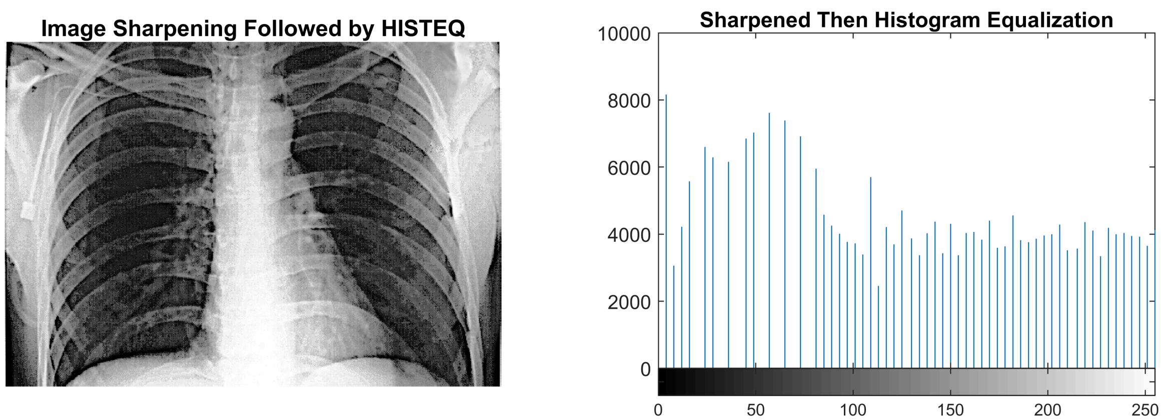
(b) Sharpening followed by Enhancement

Solution:

```
gSharp = imsharpen(f, 'radius', 5, 'amount', 2.5);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(gSharp); title('Image Sharpening','fontsize',14);
subplot(1,2,2); imhist(gSharp); set(gca,'ylim',[0,8000]);
title('Sharpening Histogram', 'fontsize',14); pause(1);
```




```
geq1 = histeq(gSharp);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(geq1); title('Image Sharpening Followed by HISTEQ');
subplot(1,2,2); imhist(geq1); set(gca,'ylim',[0,10000]);
title('Sharpened Then Histogram Equalization','fontsize',14);
```

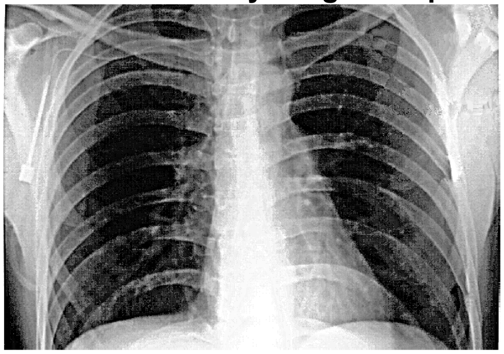


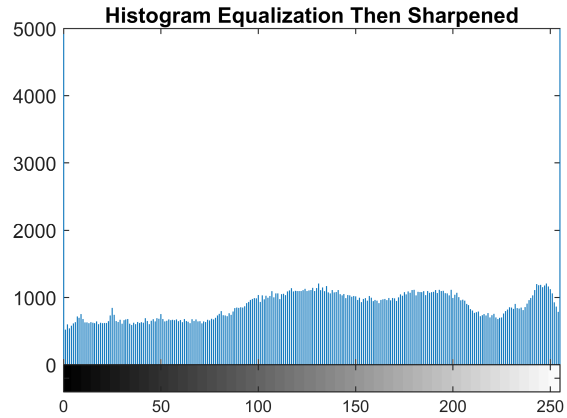
(c) Enhancement followed by Sharpening

Solution:

```
geq2 = histeq(f);
gSharp2 = imsharpen(geq2, 'radius', 5, 'amount', 2.5);
figure('Units','inches','Position',[0,0,10,3]);
subplot(1,2,1); imshow(gSharp2); title('HISTEQ Followed by Image Sharpening','fontsize',14);
subplot(1,2,2); imhist(gSharp2); set(gca,'ylim',[0,5000]);
title('Histogram Equalization Then Sharpened')
```

HISTEQ Followed by Image Sharpening





Analyzing and comparing the results of (b) and (c), sharpening the image followed by a histogram equalization, the distribution of the intensities are more spread out and contain higher magnitudes of pixel counts, whereas, the process of histogram equalization followed by sharpening the image results in a intensity distribution that is more compressed and contained lower magnitudes of pixel counts.