Ryan Chau, Philip Kwan

EE114

Computer Assignment 6

Abstract: In this assignment, we will be looking at 2-D DFTs of images by performing transforms on a sample image with and without modifications. By doing this, we get an idea of how the coefficient array produced by the 2-D DFT is distributed (where higher frequencies reside near the center and middle edges and lower frequencies reside around the corners, with some gradient between the high and low). This means that by shifting the array in a circular fashion, the array looks more like a circle of lower frequencies that increase radially.

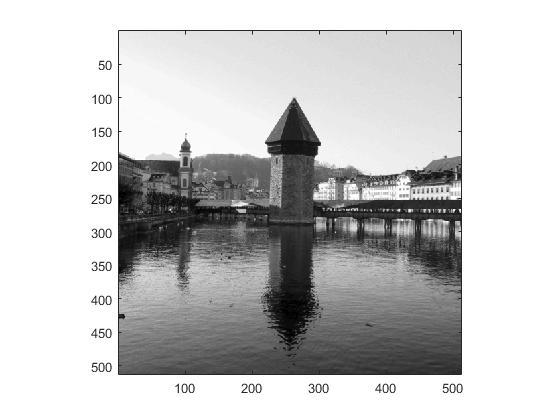
The tasks we have at hand are to display an image and its DFT (which log scaling is applied to display), to modify the coefficients of the DFT while maintaining an unchanged window around the center (which should preserve the higher frequencies) and display the modified image, and to repeat the modifications and displays with smaller windows.

**1. Displaying the image and its FT:**

To display the image:

image = imread('ca6\_image.tif')

imagesc(image); colormap(gray);



Perform DFT:

a\_ft = fft2(image);

To find max:

max = 0;

for i = 1:1:512

for j = 1:1:512

if a\_ft(i,j) > max

max = a\_ft(i,j);

end

end

end

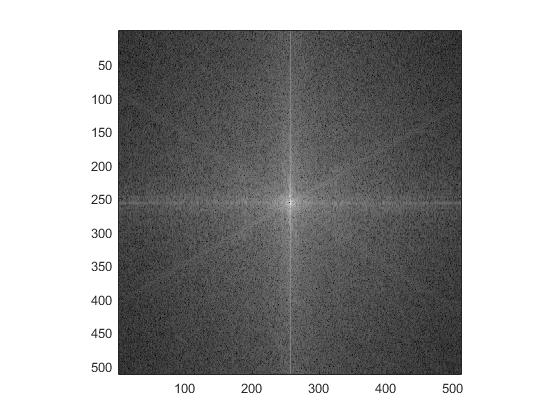
To convert to a log scale:

a\_ft = 255/6\*(log10(abs(a\_ft)/max+0.000001)+6);

Display coefficient array:

imagesc(fftshift(a\_ft)); colormap(gray);

axis square;



**2. Modify the coefficients and observe the effects**

a\_ft = fft2(image);

To modify the coefficients:

a\_ft(129 : 512-128, : ) = 0.0001;

a\_ft( :, 129 : 512-128) = 0.0001;

Take the inverse to get the reconstructed image:

a\_ift = real(ifft2(a\_ft));

for i = 1:1:512

for j = 1:1:512

a\_ift(i,j) = abs(a\_ift(i,j));

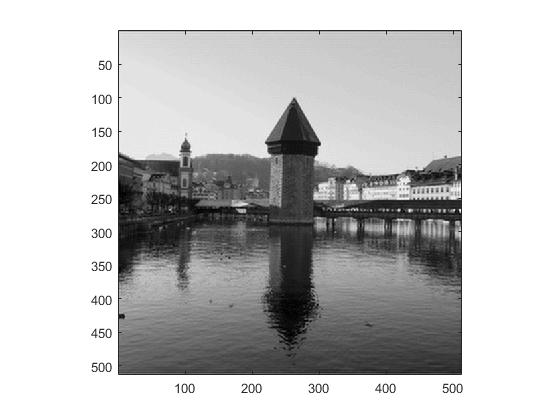
end

end

Display the image:

imagesc(a\_ift); colormap(gray);

axis square;



PNSR:

image2 = im2double(image);

sum = 0;

for i = 1:1:512

for j = 1:1:512

sum = sum + (image2(i,j)-a\_ift(i,j))^2;

end

end

sum =

7.2861e+09

PNSR = 10\*log10(255^2/(sum/(512\*512)));

The PNSR is 3.6912

**3. More processing**

For [-64, +63]:

a\_ft = fft2(image);

a\_ft(65 : 512-64, : ) = 0.0001;

a\_ft( :, 65 : 512-64) = 0.0001;

a\_ift = ifft2(a\_ft);

for i = 1:1:512

for j = 1:1:512

a\_ift(i,j) = abs(a\_ift(i,j));

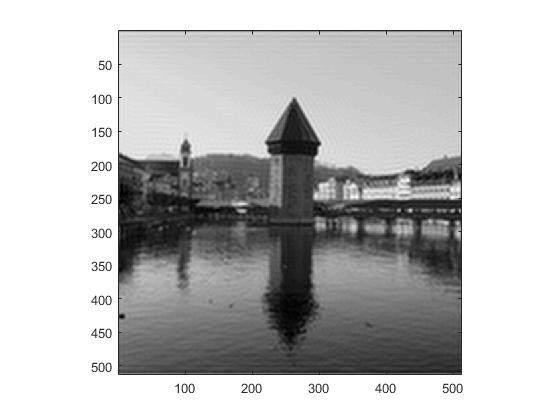
end

end

Display the image:

imagesc(a\_ift); colormap(gray);

axis square;



PNSR:

image2 = im2double(image);

sum = 0;

for i = 1:1:512

for j = 1:1:512

sum = sum + (image2(i,j)-a\_ift(i,j))^2;

end

end

sum =

2.0440e+08

PNSR = 10\*log10(255^2/(sum/(512\*512)));

The PNSR is 3.7052.

For [-32, +31]:

a\_ft = fft2(image);

a\_ft(33 : 512-32, : ) = 0.0001;

a\_ft( :, 33 : 512-32) = 0.0001;

a\_ift = ifft2(a\_ft);

for i = 1:1:512

for j = 1:1:512

a\_ift(i,j) = abs(a\_ift(i,j));

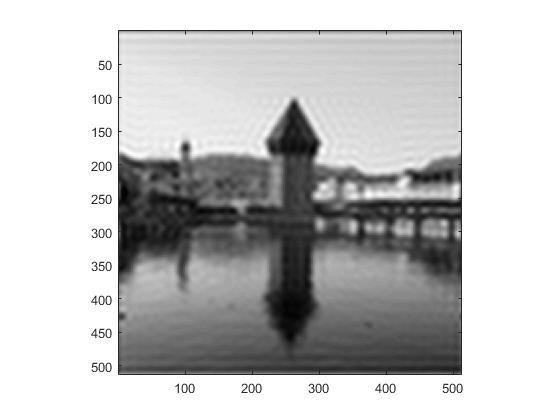
end

end

Display the reconstructed image:

imagesc(a\_ift); colormap(gray);

axis square;



PNSR:

image2 = im2double(image);

sum = 0;

for i = 1:1:512

for j = 1:1:512

sum = sum + (image2(i,j)-a\_ift(i,j))^2;

end

end

sum =

7.2339e+09

PNSR = 10\*log10(255^2/(sum/(512\*512)));

The PNSR is 3.7225.