CEN 590 Term Project

**Wavelet Transform of These Images**

Prepared by: Baftjar Tabaku

Epoka University, Tirane ALbania

2020

**Discrete Wavelet transform**

The DWT provides a sparse representation for many natural signals. In other words, the important features of many natural signals are captured by a subset of DWT coefficients that is typically much smaller than the original signal. This “compresses” the signal. With the DWT, you always end up with the same number of coefficients as the original signal, but many of the coefficients may be close to zero in value.

As a result, you can often throw away those coefficients and still maintain a high-quality signal approximation.

**‘https://www.mathworks.com’**

**Problem to be treated**

Providing the following images;

1. 10 iris images
2. 10 finger print images
3. 10 finger vein images

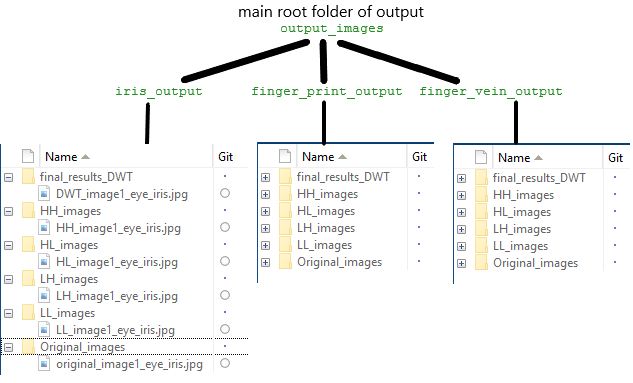
Then we need to calculate the Discrete Wavelet transform of these images. We will get 4 outputs. Determine the frequency components of these 4 outputs of each image (30 images in total).

Put next to each other:

1. the original image of an iris and its Fourier transform
2. the 1st output (of discrete wavelet transform) and Fourier transform
3. the 2nd output (of discrete wavelet transform) and its Fourier transform
4. the 3rd output (of discrete wavelet transform) and its Fourier transform
5. the 4th output (of discrete wavelet transform) the Fourier transform

**MATLAB Implementation**

Code will be separated into three files, one will be the main file where the execution will happen step by step the ‘main.m’ file, the other one the ‘DWT\_function.m’ file, implemented as a function and the ‘furier\_function.m’ that will calculate the furrier function.

Doing such work, we need to be precise, according to the paths and directories. Images are saved according to a root and folder, with specific names according to the following schema:

The folders will be created automatically if they do not exist, and according to root and sub-root everything will be precise.

**Project Code**

The ‘main.m’ file that will call and execute the respective functions functions.

clc, clear all, close all

fprintf('Cen Master, Epoka Univesity, Computer Engenieering\n===Discrete Wavelet Transform===\n');

% DWT Part

% NOTE: The script images should run one by one, uncomment, run , comment,

% go to next and so on.

% Creating the folders according to the root

% ======== Iris Part - 10 images ========

% creating root folder for the output

if ~exist('output\_images\iris\_output', 'dir')

mkdir('output\_images\iris\_output')

end

DWT\_function('output\_images\iris\_output','images\iris\','image1\_eye\_iris','.jpg');%1

% DWT\_function('output\_images\iris\_output','images\iris\','image2\_eye\_iris','.jpg');%2

% DWT\_function('output\_images\iris\_output','images\iris\','image3\_eye\_iris','.jpg');%3

% DWT\_function('output\_images\iris\_output','images\iris\','image4\_eye\_iris','.jpg');%4

% DWT\_function('output\_images\iris\_output','images\iris\','image5\_eye\_iris','.jpg');%5

% DWT\_function('output\_images\iris\_output','images\iris\','image6\_eye\_iris','.jpg');%6

% DWT\_function('output\_images\iris\_output','images\iris\','image7\_eye\_iris','.jpg');%7

% DWT\_function('output\_images\iris\_output','images\iris\','image8\_eye\_iris','.jpg');%8

% DWT\_function('output\_images\iris\_output','images\iris\','image9\_eye\_iris','.jpg');%9

% DWT\_function('output\_images\iris\_output','images\iris\','image10\_eye\_iris','.jpg');%10

% ======== Finger Print Part - 10 images ========

% creating root folder for the output

if ~exist('output\_images\finger\_print\_output', 'dir')

mkdir('output\_images\finger\_print\_output')

end

%

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image1\_finger\_print','.jpg');%1

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image2\_finger\_print','.jpg');%2

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image3\_finger\_print','.jpg');%3

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image4\_finger\_print','.jpg');%4

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image5\_finger\_print','.jpg');%5

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image6\_finger\_print','.jpg');%6

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image7\_finger\_print','.jpg');%7

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image8\_finger\_print','.jpg');%8

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image9\_finger\_print','.jpg');%9

% DWT\_function('output\_images\finger\_print\_output','images\finger\_print\','image10\_finger\_print','.jpg');%10

% ======== Finger Vain Part - 10 images ========

if ~exist('output\_images\finger\_vein\_output', 'dir')

mkdir('output\_images\finger\_vein\_output')

end

%

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image1\_finger\_vein','.jpg');%1

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image2\_finger\_vein','.jpg');%2

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image3\_finger\_vein','.jpg');%3

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image4\_finger\_vein','.jpg');%4

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image5\_finger\_vein','.jpg');%5

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image6\_finger\_vein','.jpg');%6

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image7\_finger\_vein','.jpg');%7

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image8\_finger\_vein','.jpg');%8

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image9\_finger\_vein','.jpg');%9

% DWT\_function('output\_images\finger\_vein\_output','images\finger\_vein\','image10\_finger\_vein','.jpg');%10

In the following code we run each image one by one, in a way not to mix the results and keeping track on what we are doing.

The other will be the ‘DWT\_function.m’ this will be the part of the Discrete Wavelet transform that will take an image input and will make the respective calculations.

function [] = DWT\_function(output\_folder,image\_path\_location,image\_to\_proceed,image\_extension)

the first one will be the main folder of output, then the image path, then the image name and the extension at the end, so due to the fact that we had to work on multiple files and high volume we need to save them properly in a automatic mode.

function [] = DWT\_function(output\_folder,image\_path\_location,image\_to\_proceed,image\_extension)

% Join location strings

path\_location=strcat(image\_path\_location,image\_to\_proceed,image\_extension);

input\_image = imread(path\_location);

input\_image=rgb2gray(input\_image);% convert Gray

% Original Image

% figure(1)

% imshow(input\_image);

% title('Original Image');

% resizing the image

size\_of\_output\_result\_image=size(input\_image);

[LL,LH,HL,HH]=dwt2(input\_image,'db1');

% processingd Discrete Wavelet transform with levels

figure(1)

subplot(2,3,1);imshow(input\_image);title('Original Image');

subplot(2,3,2);imshow(LL);title('LL band of image');

subplot(2,3,3);imshow(LH);title('LH band of image');

subplot(2,3,5);imshow(HL);title('HL band of image');

subplot(2,3,6);imshow(HH);title('HH band of image');

output\_result\_transformed\_image = idwt2(LL,LH,HL,HH,'db1',size\_of\_output\_result\_image);

subplot(2,3,4);imshow(output\_result\_transformed\_image);

title('Transformed Image DWT');

% increase size of image to display united

set(figure(1),'units','points','position',[0,0,820,450])

%0-Saving the original image

original\_path=strcat(output\_folder,'\Original\_images\');

if ~exist(original\_path, 'dir')

mkdir(original\_path)

end

original\_path=strcat(original\_path,'original\_',image\_to\_proceed,image\_extension);

original\_image = figure(8);

set(original\_image,'Visible','off');

imshow(input\_image);

saveas(original\_image,original\_path);

% creating path for final result according to the root, but checking it

% first, ex 'output\_images\iris\_output'

% Creating final result folder -> \final\_results\_DWT\

transformed\_result\_image=strcat(output\_folder,'\final\_results\_DWT\');

if ~exist(transformed\_result\_image, 'dir')

mkdir(transformed\_result\_image)

end

%Adding file with the extension to the string

transformed\_result\_image=strcat(transformed\_result\_image,'DWT\_',image\_to\_proceed,image\_extension);

% Discrete Wavelet transform image

main\_output\_image = figure(2);

saveas(imshow(output\_result\_transformed\_image),transformed\_result\_image);

% imshow(output\_result\_transformed\_image)

% title('Transformed Image DWT');

set(main\_output\_image,'Visible','off');% close image after if needed to be closed

% Working with each output step by step

% storing outputs on separated variables

% Saving images LL (low-low), LH(low-high), HL(high-low), HH(high-high)

% Check if path exist, if not we create it

% 1 LL path

LL\_path=strcat(output\_folder,'\LL\_images\');

if ~exist(LL\_path, 'dir')

mkdir(LL\_path)

end

LL\_path=strcat(LL\_path,'LL\_',image\_to\_proceed,image\_extension);

image\_1\_LL = figure(4);

set(image\_1\_LL,'Visible','off');%image visible off

imshow(LL);

saveas(image\_1\_LL,LL\_path);

% 2 LH Path

LH\_path=strcat(output\_folder,'\LH\_images\');

if ~exist(LH\_path, 'dir')

mkdir(LH\_path)

end

LH\_path=strcat(LH\_path,'LH\_',image\_to\_proceed,image\_extension);

image\_2\_LH = figure(5);

set(image\_2\_LH,'Visible','off');

imshow(LH);

saveas(image\_2\_LH,LH\_path);

% 3 HL path creation

HL\_path=strcat(output\_folder,'\HL\_images\');

if ~exist(HL\_path, 'dir')

mkdir(HL\_path)

end

HL\_path=strcat(HL\_path,'HL\_',image\_to\_proceed,image\_extension);

image\_3\_HL = figure(6);

set(image\_3\_HL,'Visible','off');

imshow(HL);

saveas(image\_3\_HL,HL\_path);

% 4 HH image path creation

HH\_path=strcat(output\_folder,'\HH\_images\');

if ~exist(HH\_path, 'dir')

mkdir(HH\_path)

end

HH\_path=strcat(HH\_path,'HH\_',image\_to\_proceed,image\_extension);

image\_4\_HH = figure(7);

set(image\_4\_HH,'Visible','off');

imshow(HH);

saveas(image\_4\_HH,HH\_path);

% Fourier Transform part, each

% image output will be included into the function fourier transform

% 1-Low-Low image

furier\_function(LL\_path,2)

% 2-Low-High image

furier\_function(LH\_path,3);

% 3-High-Low image

furier\_function(HL\_path,4);

% 4-High-high image

furier\_function(LH\_path,5);

end

The upper code will take the image as input, join the folders path and locations under a single location, loading the image, calculating the 4-outptus and the final result image, then images will be plotted and saved.

% Fourier Transform part, each

% image output will be included into the function fourier transform

% 1-Low-Low image

furier\_function(LL\_path,2)

% 2-Low-High image

furier\_function(LH\_path,3);

% 3-High-Low image

furier\_function(HL\_path,4);

% 4-High-high image

furier\_function(LH\_path,5);

Each of 4 outputs will be proceed and pass thought a Fourier transform function, according to the outputs that will be taken as input by this function named

furier\_function(image\_path,2)

The number 2 will be the image name ex: “figure 2”, the code will be shown below;

function [] = furier\_function(path\_image\_location, fig\_num)

% First we will be loading image

imdata =imread(path\_image\_location);

figure(fig\_num)

subplot(2,3,1);imshow(imdata); title('Original Image');

imdata = rgb2gray(imdata);

subplot(2,3,2); imshow(imdata); title('Gray Image');

%Get Fourier Transform of an image

F = fft2(imdata);

% Fourier transform of an image

S = abs(F);

subplot(2,3,3);imshow(S,[]);title('Fourier transform of an image');

%get the centered spectrum

Fsh = fftshift(F);

subplot(2,3,4);imshow(abs(Fsh),[]);title('Centered fourier transform of Image')

%apply log transform

S2 = log(1+abs(Fsh));

subplot(2,3,5);imshow(S2,[]);title('log transformed Image')

%reconstruct the Image

F = ifftshift(Fsh);

f = ifft2(F);

subplot(2,3,6) ;imshow(f,[]),title('reconstructed Image')

% increase image size

set(figure(fig\_num),'units','points','position',[0,0,820,450])

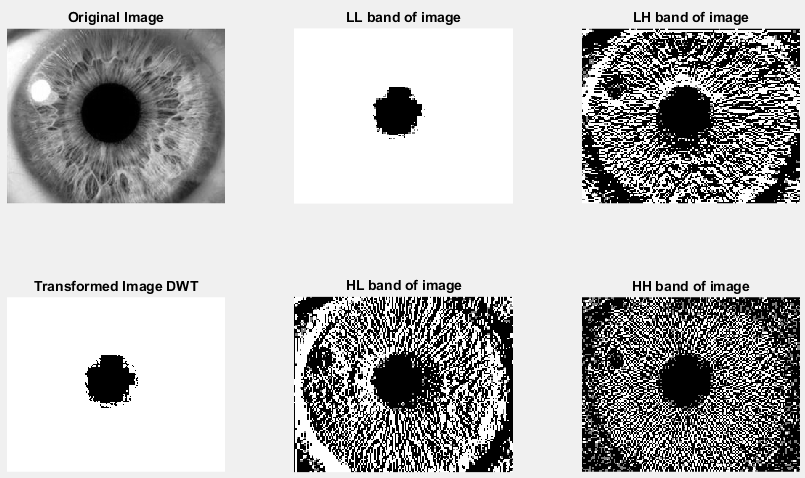
end

Image will be taken, and proceed, converted in gray color, passing thought Fourier functions and plotted according to the respective calculations.

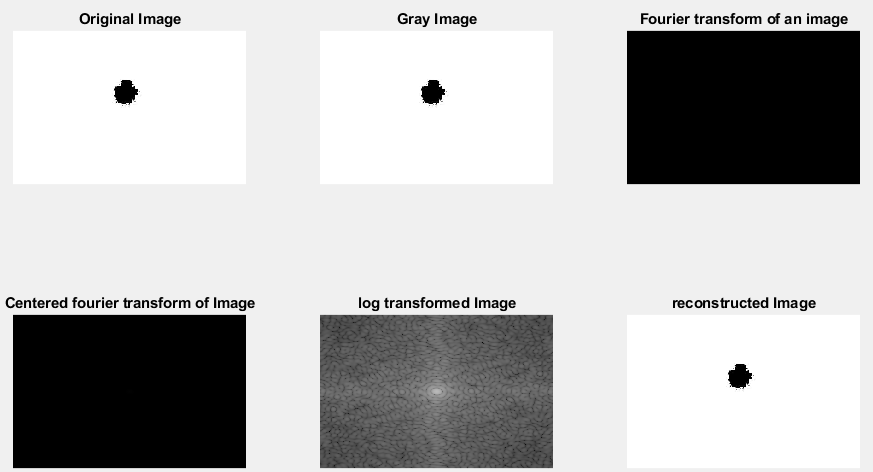
**Proceeding with the images**

Iris part, the 10 images analysis with the respective Fourier transform part for each output,

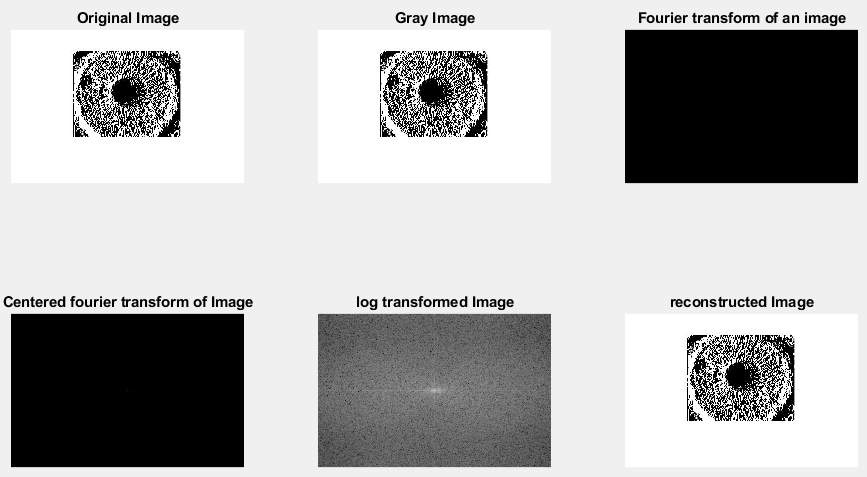
DWT



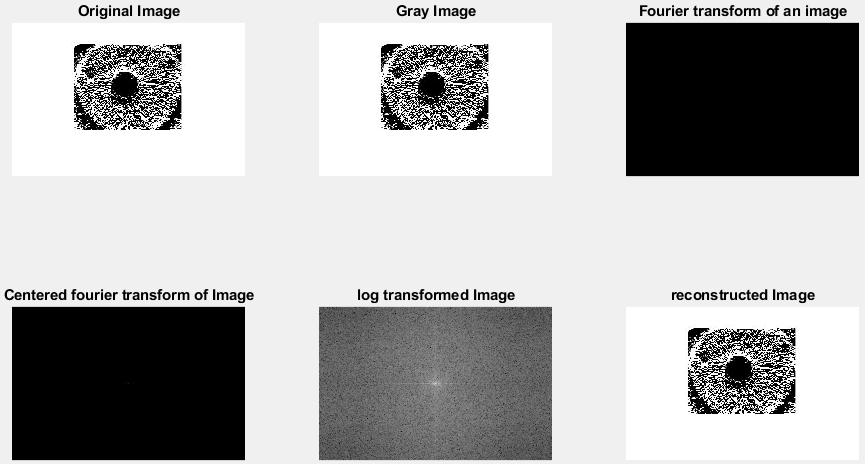
Fourier Part for LL image



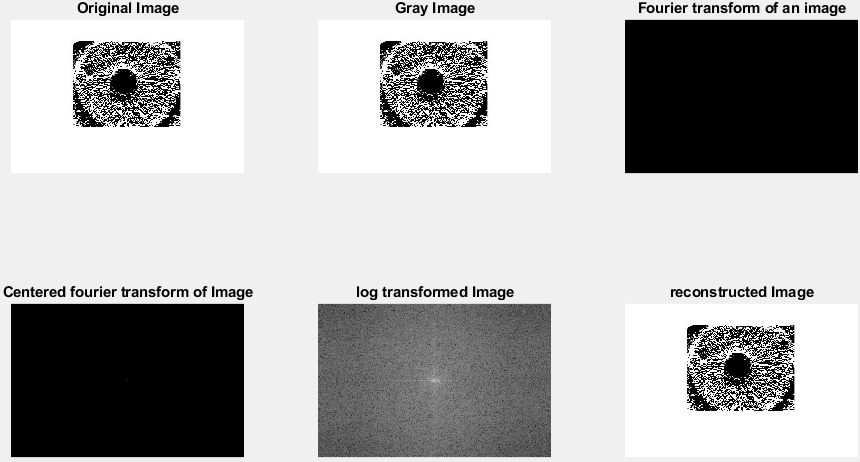
LH Fourier



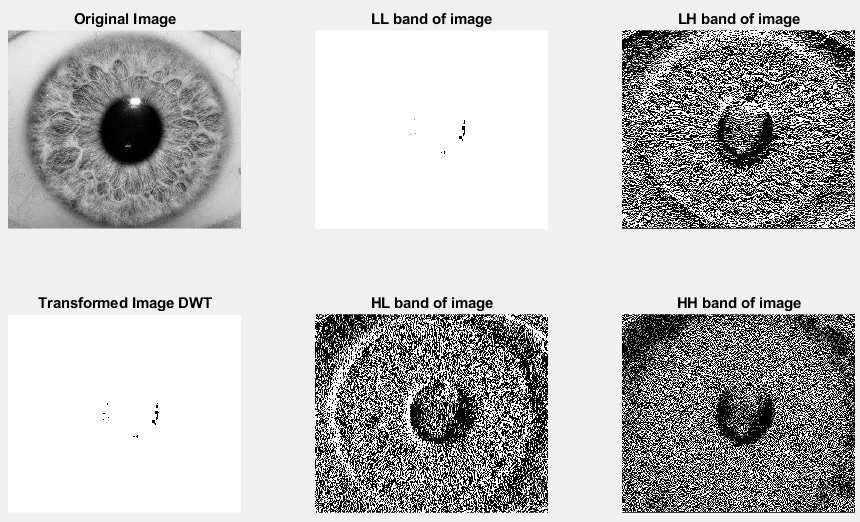
HL Fourier



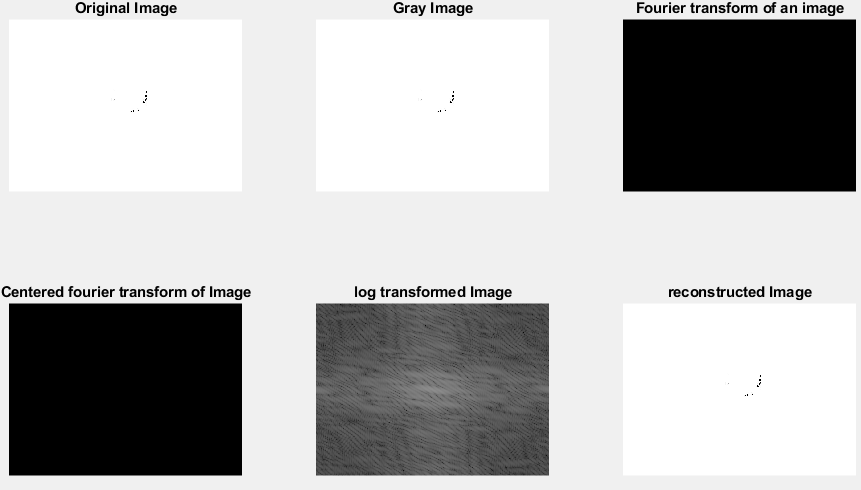
HH Fourier



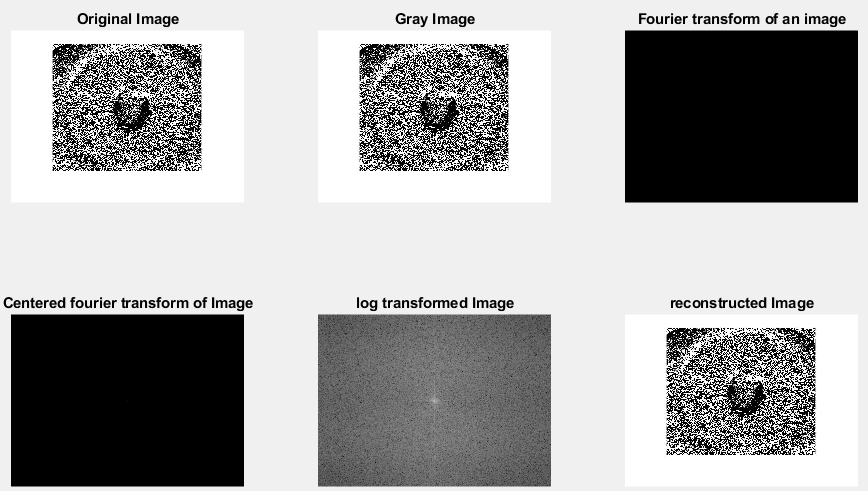
Iris, image 2 DWT



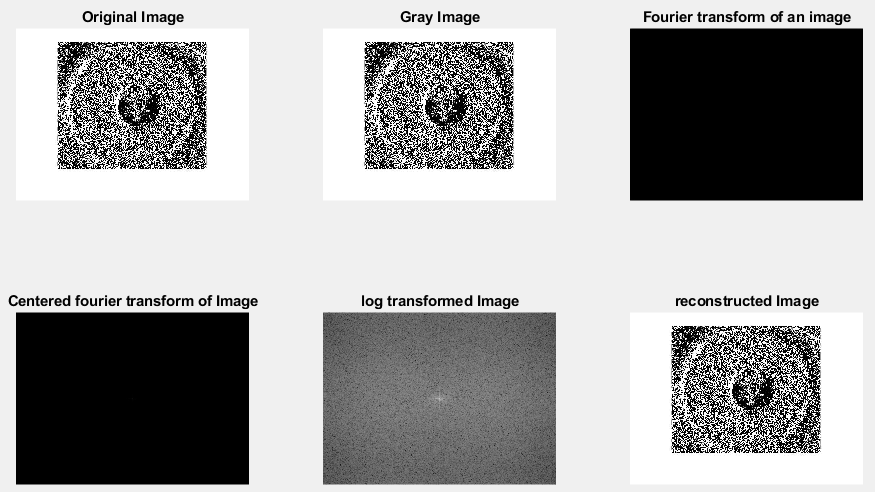
LL Fourier



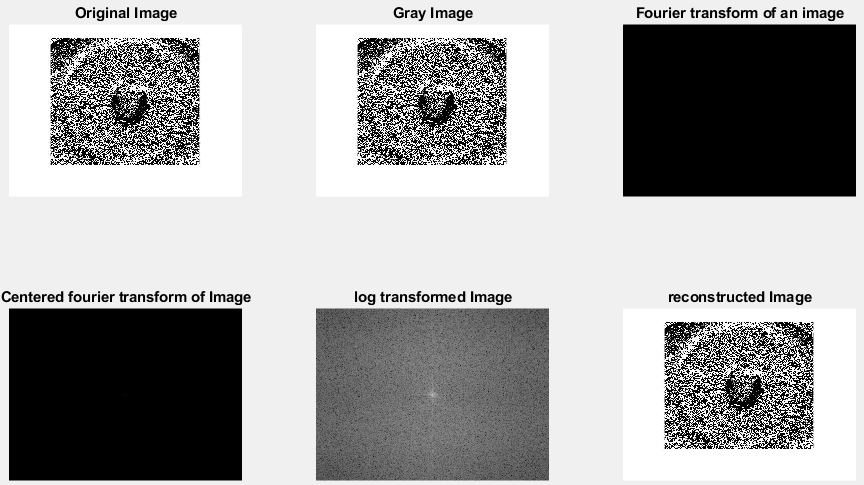
LH Fourier



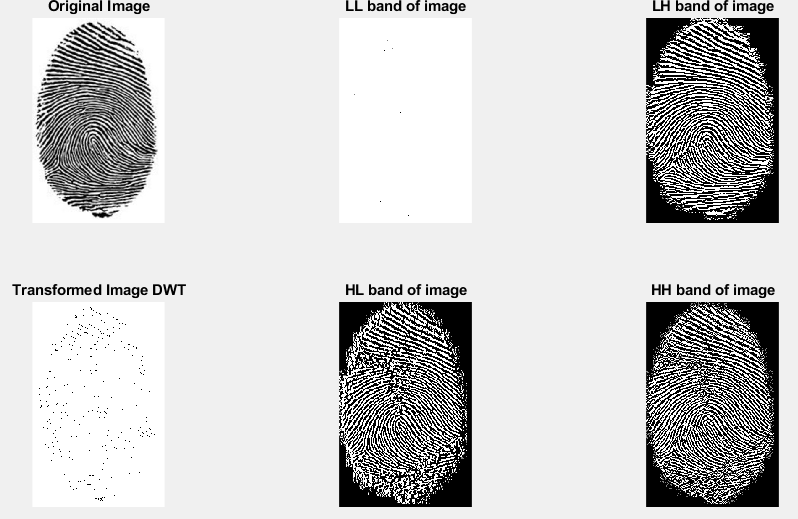
HL Fourier



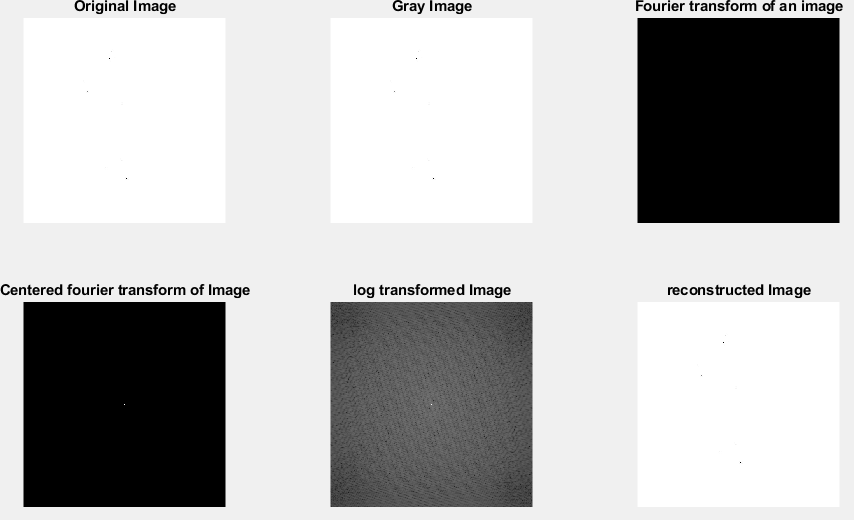
HH Fourier



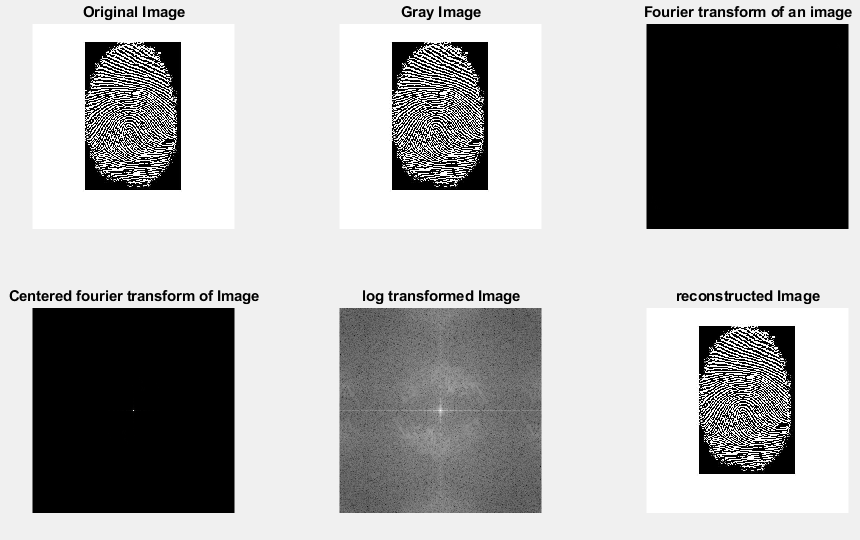
Finger Print DWT,



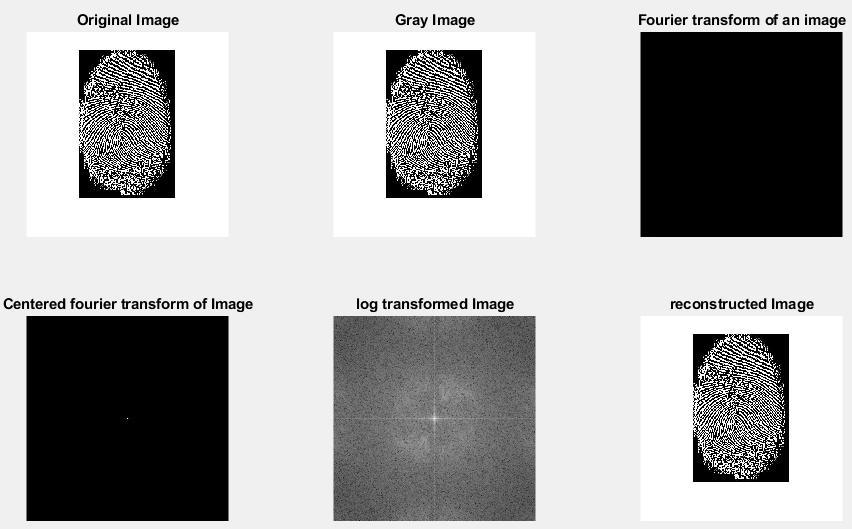
LL Fourier



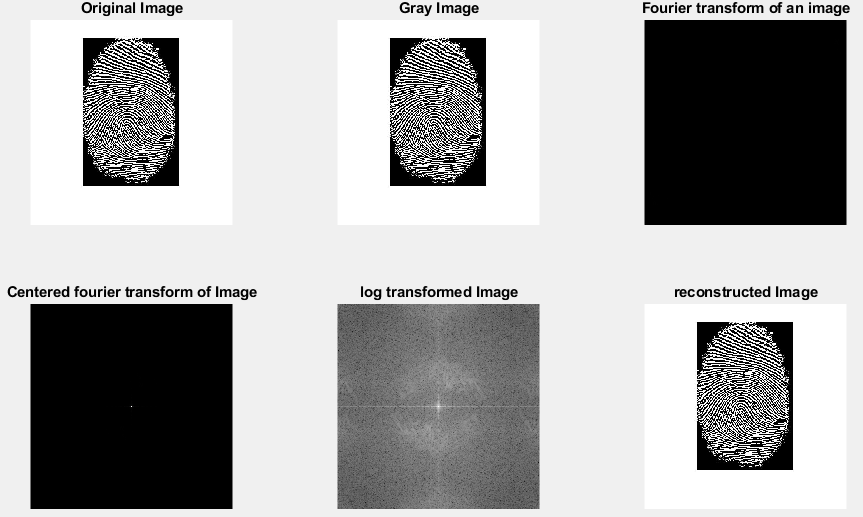
LH Fourier



HL Fourier



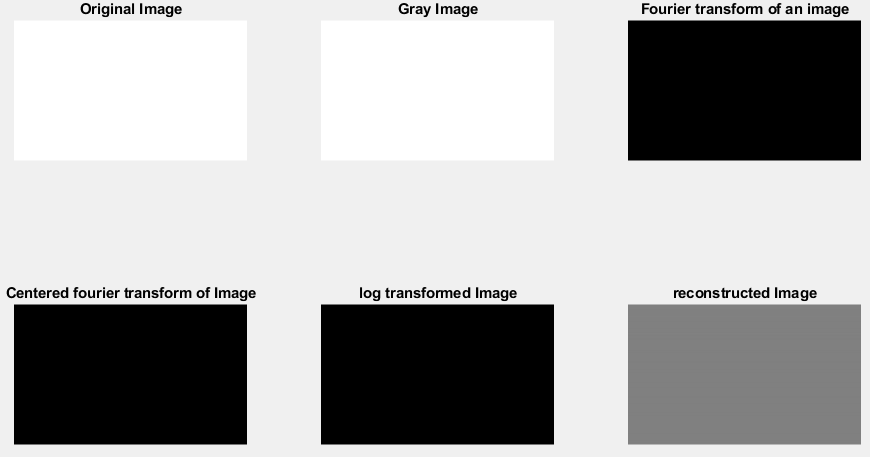
HH Fourier

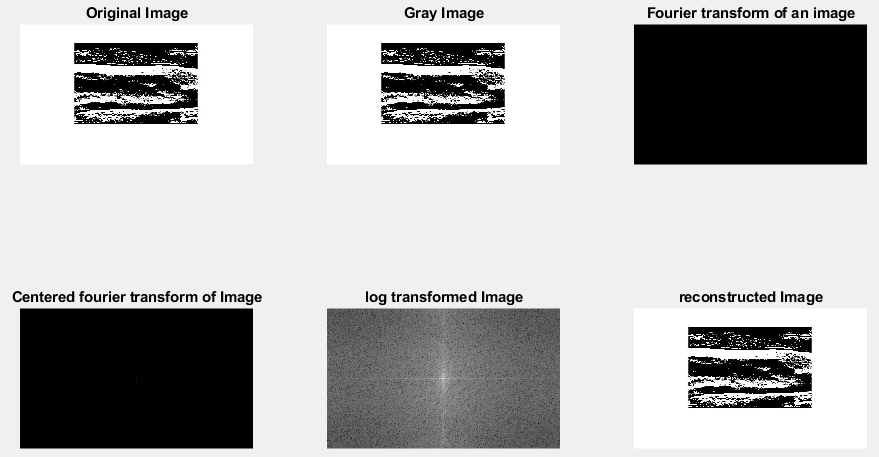


Finger Vein DWT,

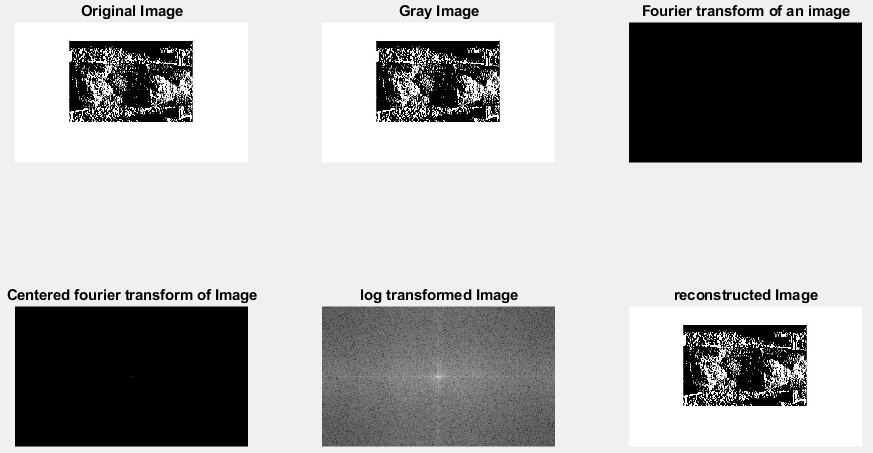


LL Fourier,

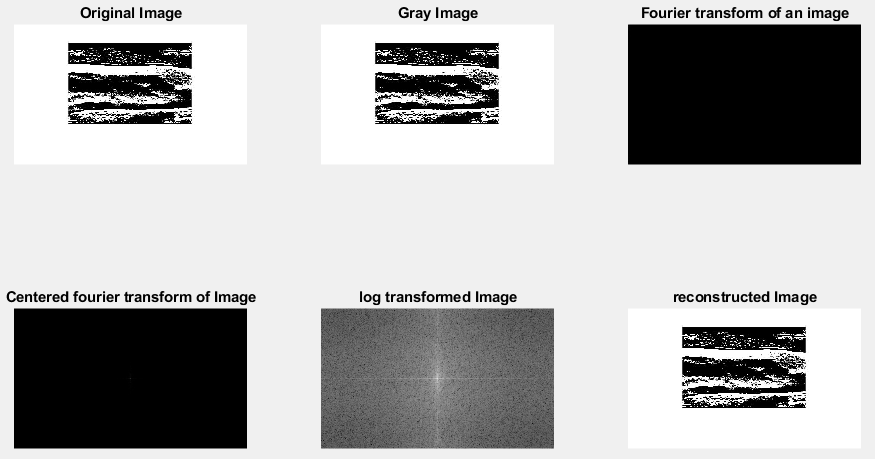


LH Fourier,

HL Fourier,



HH Fourier,



There are the displayed each image using the DWT, then each DWT output image is processed using the Fourier transform.