EE 204 - SIGNALS AND SYSTEMS - LABORATORY 11

1. Images can be considered as signals of two independent variables. A gray scale digital image can be represented with a 2D discrete function $x[m, n]$ such that $x[m, n]$ denotes the light intensity of the (m, n)th pixel of the image (m and n are integers). Since practical images are of finite size (such as 512 *x* 512, 1024 *x* 1024 etc.), we usually have $x[m, n] = 0$ for $m \notin [0, M_x - 1]$ or $n \notin [0, N_x - 1]$ where the image size is *Mx x Nx*.

Place the m-files named **ReadMyImage.m** and **DisplayMyImage.m** and the image named **Part5.bmp** under the current directory of Matlab.

```
function [x] = ReadMyImage(string)
x=double((rgb2gray(imread(string))));
x=x-min(min(x));
x=x/max(max(x));
x=x-0.5;
end
```

```
function []=DisplayMyImage(Image)
Image=Image-min(min(Image));
figure;
imshow(uint8(255*Image/max(max(abs(Image)))));
```

Issue the command **A=ReadMyImage('Part5.bmp');**

You will see that a 512 x 512 matrix of type double will be created and stored in the workspace. This matrix contains the image. The original image is a color image, but it is converted to a gray scale image by the **ReadMyImage** function. Next, issue the command **DisplayMyImage(A);** A new figure window will open and the image will be displayed.

2. Let's define the two dimensional discrete impulse signal as,

$$\delta[m, n] = \begin{cases} 1 & \text{if } m = 0, n = 0 \\ 0 & \text{otherwise} \end{cases}$$

Here formula for 2D convolution of x[m, n] and h[m, n] that we denote as x[m, n] **h[m, n] is given,

$$\begin{aligned} y[m,n] &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k,l]h[m-k,n-l] \\ &= x[m,n] * *h[m,n] \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h[k,l]x[m-k,n-l] \end{aligned}$$

This formula for 2D convolution of x[m, n] and h[m, n] will be used later in this lab. work.

3. A Matlab function **DSLSI2D(h,x)** is defined and saved that computes the output when a finite sized input image *x*[*m, n*] (of size *Mx x Nx*) is input to a 2D FIR DS LSI system whose impulse response *h*[*m, n*] is of size *Mh x Nh*.

```
function [y] = DSLSI2D(h,x)
        [Mh, Nh] = size(h);
        [Mx, Nx] = size(x);
y = zeros(Mh+Mx-1, Nh+Nx-1);
        for k = 0:Mh-1
                for l = 0:Nh-1
                y(k+1:k+Mx, l+1:l+Nx) = y(k+1:k+Mx, l+1:l+Nx)+h(k+1, l+1)*x;
end
end
end
```

Check this function for input signal x[m, n] and impulse response h[m, n] and compute the output 2D discrete signal y[m, n]:

x = [1 0 2; -1 3 1; -2 4 0];

h = [1 -1; 0 2];

y = DSLSI2D(h, x)

$$y = \begin{matrix} 1 & -1 & 2 & -2 \\ -1 & 6 & -2 & 3 \\ -2 & 4 & 2 & 2 \\ 0 & -4 & 8 & 0 \end{matrix}$$

4. **Image Denoising:**

Read the picture **Part4.bmp** in a matrix x and display it. This picture is corrupted by noise. In this part, we will try to rescue this image from noise without disturbing the image itself as much as possible.

If we know that the noise on the picture has high frequency content and typical daily life images taken by a typical camera have low-frequency content, then we can apply a low pass filter to the noisy image and try to eliminate the noise.

A typical used 2D FIR low pass filter has the following impulse response:

$$h[m,n] = \begin{cases} 0 & \text{if } m < 0 \text{ or } n < 0 \text{ or } m > M_h - 1 \text{ or } n > N_h - 1 \\ \text{sinc}\left\{B\left(m - \frac{M_h-1}{2}\right)\right\}\text{sinc}\left\{B\left(n - \frac{N_h-1}{2}\right)\right\} & \text{otherwise} \end{cases}$$

Here, B is a free parameter that determines the bandwidth of the filter. We should select *B* between 0 and 1.

Now, let Mh = Nh = 30 + D7 and B = 0.7. We will prepare the h matrix that represents h[m, n].

Using the code that you developed in Part 3, process the noisy image with this filter. Display the output

image. Repeat the exercise with *B* = 0.4 and *B* = 0.1 as well. Which value of *B* seems to be the most appropriate one? As you see, it is possible to greatly clarify the information in a corrupted signal using only the simple concepts of convolution, LSI systems, etc. In this part, we tried to remove the noise from a corrupted image. Such problems are called **image denoising** problems. Many algorithms have been developed by many researchers that try to clear the images from the corrupting noise while giving minimum damage to the original image.

The matlab function that prepares h[m, n] is below:

```
function [h] = Prepareh(B)
h = zeros(30, 30);
for m = 0:29
        for n = 0:29
                h(m+1, n+1) = (sinc(B*(m-((30-1)/2)))*sinc(B*(n-((30-1)/2))));
end
end
end
```

In this fuction, B is the free parameter that determines the bandwith of the 2D FIR low pass filter. In this part, we change the value of B, and use the two functions that are named

Prepareh(B) and **DSLSI2D(h,x)** in order to rescue this image from the noise with giving minimum damage to original image.

Now, write the code below in Matlab and change the value of B as 0.7, 0.4, 0.1. Which B value is better?

```
DisplayMyImage(y);
X = ReadMyImage('Part4.bmp');
B = 0.7;
h = Prepare(B);
y = DSLSI2D(h, x);
DisplayMyImage(y);
```

5. **Edge Detection:**

In a typical image, edges are the set of points in the close vicinity of which the pixel values change abruptly. Since changes are sudden and great, edges are inherently associated with high frequencies. Therefore, we can make use of high pass filters to detect edges.

Read the picture named **Part5.bmp** in a matrix named **x** and display it.

Consider a 2D FIR DS LSI system whose impulse response $h1[m, n]$ is equal to:

$$h_1[m,n] = \begin{cases} 0.5 & \text{if } m=0, n=0 \\ -0.5 & \text{if } m=0, n=1 \\ 0 & \text{otherwise} \end{cases}$$

Prepare $h1[m, n]$ and process your image with this filter. Let $y1[m, n]$ denote the resulting image. Display the image which is defined as $s_1[m,n] = y_1^2[m,n]$. Which parts of the original image are emphasized?

Now, let;

$$h_2[m,n] = \begin{cases} 0.5 & \text{if } m=0, n=0 \\ -0.5 & \text{if } m=1, n=0 \\ 0 & \text{otherwise} \end{cases}.$$

As you see, $h2[m, n] = h1[n, m]$. Again, prepare $h2[m, n]$ and process your image with this filter. Let $y2[m, n]$ denote the resulting image. Display the image which is defined as $s_2[m,n] = y_2^2[m,n]$.

Which parts of the original image are emphasized now? Comment on the difference with the image you obtained with $h1[m, n]$.

Finally, process the original image with the filter $h3[m, n] = 0.5h1[m, n]+0.5h2[m, n]$. Let $y3[m, n]$ denote the resulting image. Display the image which is defined as $s_3[m,n] = y_3^2[m,n]$. Which parts of the original image are emphasized now? Comment on the difference with the images you obtained with $h1[m, n]$ and $h2[m, n]$.

First we will calculate the output voltage y1[m, n] and then calculate the square of the y1[m, n] denoted as s1[m,n].

```
x = ReadMyImage('Part5.bmp');
h1 = [0.5 -0.5];
y1 = DSLSI2D(h1, x);
DisplayMyImage(y1);

x = ReadMyImage('Part5.bmp');
h2 = [0.5; -0.5];
y2 = DSLSI2D(h2, x);
DisplayMyImage(y2);

x = ReadMyImage('Part5.bmp');
h3 = [0.5 -0.25; -0.25 0];
y3 = DSLSI2D(h3, x);
DisplayMyImage(y3);
```

From these impulse responses h1[m, n], h2[m, n] and h3[m, n], it is expected to obtain the following results;

In impulse response h1[m, n], vertical edges are visible, in impulse response h2[m, n] horizontal edges are visible and in h3[m, n] both horizontal and vertical edges are visible.

6. Pattern Recognition:

A main problem of image (or signal) processing is the detection of certain objects within a picture. Read and display the image named **Part6x.bmp**. You will see our national soccer team. Our purpose in this part is to detect the faces in the image. Such problems are named **pattern recognition** problems. We will use one of the basic methods which is called

matching filter method. Read the image **Part6h.bmp** into Matlab and think of it as the impulse response of a 2D DS LSI FIR system. Display the image to see how the impulse response looks like. You will see an inverted face. That is, to detect the faces within the image, we are using a 2D DS LSI system whose impulse response is an inverted face. Note that by inverted, we mean that the face is rotated by 180 degrees.

Now, pass the input image through this system and find the output image. Display the absolute value of the output image. Search for the points that look bright. Where do they occur? Do you always see a face at the location that you think there is a bright point, or do you sometimes find bright points where there is no face? To make the bright points more visible, compute the image $|y[m,n]|^3$ and display it. Also compute the image $|y[m,n]|^5$ and display it. Taking which power is sufficient for you to detect the faces without any confusion?

```
x = ReadMyImage('Part6x.bmp');
h = ReadMyImage('Part6h.bmp');
y = DSLSI2D(h, x);
absy = abs(y);
DisplayMyImage(absy);
```

This function is not appropriate for detecting faces, we see bright points almost the whole image. From this image, it is very hard to detect faces of soccer team. Now, let's find $|y[m,n]|^3$ and $|y[m,n]|^5$.

References:

Bilkent University, EEE Department, EEE321, Laboratory 10