York University
Lassonde School of Engineering
Department of Electrical Engineering and Computer Science

EECS4640/5640: Medical Imaging Techniques

EA-1: Imaging Signals and Systems, and Image Quality

Objectives

The goal of this experiential assignment (EA) is to become familiar and work with a number of important functions in **Matlab** to analyze imaging signals and systems, and the concepts of image denoising, sampling, aliasing, resolution and contrast.

Your work will be evaluated based on your EA report. When preparing your report, keep in mind these points:

- Prepare the report with the complete details of your work for each part.
- When you are asked to plot a signal, do not forget to save the figure and include the image in your report.
- Include the **Matlab** code for each section in your report.
- When plotting signals, make sure to appropriately label the axes.
- Submit your report (PDF file) in eClass **(Experiential Assignment 1 Report)** by the deadline.

Note: You can use the help command in case you need more information about a **Matlab** function, for example:

```
>>help plot
>>help stem
>>help xlabel
```

1. Plotting Continuous and Discrete Signals

a) Plot `Sin(x)` for the period of -π to π. Use `plot(x,y)` where x is the argument and `y=sin(x)`. You can define the range of x using: `x=-pi:0.01:pi`, where 0.01 defines the step size. Put a semicolon after a variable assignment to prevent unnecessary echoing, i.e. `x=-pi:0.01:pi;`

b) For plotting discrete signals we can use `stem(x,y)`. In **Matlab**, the unit step function is defined by: `Heaviside(n)`. Plot this function for the interval of -10 to 10 using `stem`.

c) Write a **Matlab** script that plots the function `f[n]`, defined below. You should add a title to the plot and labels to x and y axes using `title`, `xlabel`, and `ylabel`. Save .m file as `sinplot.m`. Plot for the interval `n = -10:10`.

$$f[n] = \sin(\omega_k n) \;\; where\; \omega_k = \frac{2\pi k}{3} \; and \; k = 2$$

d) Plot the `f[n]` defined above for k =1, 4. Plot for all values of k (k=1, 2, 4) in the same figure using `subplot`. How many unique signals have you plotted? If two signals are identical, explain how different ω_k results in the same signal.

2. Special Signals

When plotting the signals in this section, do not forget to add proper title and labels for x and y axes.

a) In **Matlab**, the delta function (point impulse function) is defined using `dirac(x)`. What is the value of `dirac(x)` at origin, i.e. `dirac(0)`?

b) The `dirac(x)` is defined for continuous signals. For discrete signals, we should change the value of function at origin to 1. Modify `dirac`, then plot discrete `dirac` using stem.

c) Plot Rect function using `rectangularPulse` on the interval `x= -5:0.1:5`.

d) Using `Sinc()` function in **Matlab**, plot $\frac{sin\frac{\pi}{2}x}{x}$ on the interval `x=-5*pi:0.1:5*pi`.

e) Run the following script in **Matlab** and plot the results for different values of u_0 and u_1. Interpret the results.

```
u0 = 1;
u1=1;

[x,y] = meshgrid(-3:0.05:3);
z =sin(2*pi*(u0*x+u1*y));

figure;
surf(x,y,z,'EdgeColor','None');
colormap gray
view(2);

xlabel('x')
ylabel('y')
```

3. Signals and Systems Properties

a) Consider the following signal $x[n]$. Use `conv` function in **Matlab** to calculate $y[n] = x[n] * x[n]$. Then Plot $y[n]$.

$$x[n] = \begin{cases} n & 0 \le n \le 5 \\ 0 & otherwise \end{cases}$$

b) Use `conv` and discrete `dirac` you defined in 2-b to show the following properties of Dirac delta function by selecting an arbitrary function $f[n]$:

$$f[n] * \delta[n] = f[n]$$

$$f[n] * \delta[n-m] = f[n-m]$$

c) We can also use 2D convolution for image filtering. First load **'T1.bmp'** as a grayscale image using `imread` and `rgb2gray`. Then define a 3×3 averaging (low-pass) filter `h=(1/9)*ones(3,3)`. Convolve image with h using `imfilter(image,h,'conv')`.

Display the results using `imshow`. Also display the results for 5×5 and 9×9 filters, e.g. `h=(1/25)*ones(5,5)`. Note that the sum of all samples in each filter is 1. What is the effect of convolving filter h with image? What happens when you increase the size of filter h? Why?

d) Design a 3×3 filter, h, that enhances vertical edges of an image, similar to Figure 1. Apply your filter to image **'T1.bmp'** and display the result.

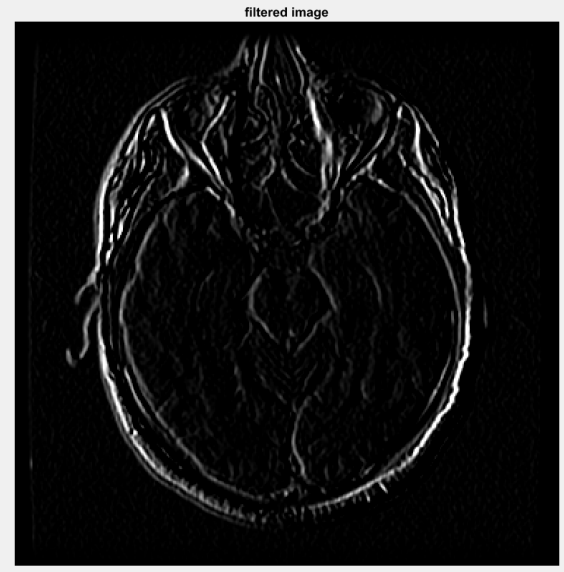

Figure 1 - Vertically Enhanced Image

e) Repeat part (d) for horizontal edges.

4. Fourier Transform

4.1. Background

The Discrete Time Fourier Transform (DTFT) is used to represent discrete time signals in terms of complex exponential $e^{j\omega n}$. The DTFT of a discrete-time signal $x[n]$ is given by,

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

Note that DTFT is periodic with period 2π. The major limitation for any numerical calculation of DTFT is that $x[n]$ must have a finite length. Otherwise it will have an infinite summation? The Discrete Fourier Transform (DFT) is the sampled (discrete) version of the DTFT in frequency-domain. It can be used to numerically calculate the frequency spectrum of a discrete-time signal using computers (computer implementations cannot handle infinite number of spectrum values). In order to compute DFT of a signal $x[n]$ using **Matlab,** we should first truncate the signal $x[n]$ to make sure it has a finite length (by truncation, the signal is estimated with a periodic signal). This is not necessarily a limitation because in real-world applications we usually deal with sequences of finite length (or periodic). In DFT, $X(\omega)$ is only computed at $\omega_k = \frac{2\pi k}{N}$ for $k = 0,1, \dots, N-1$. For a discrete signal $x[n]$ that is truncated *outside* the interval $0 \leq n \leq M-1$ the DFT is defined by,

$$X[k] = \sum_{n=0}^{M-1} x[n] e^{-j2\pi kn/N}, k = 0,1, \dots, N-1$$

The inverse DFT is given by,

$$x[n] = \sum_{k=0}^{N-1} x[k] e^{\frac{j2\pi kn}{N}}, n = 0,1, \dots, M-1$$

The Fast Fourier Transform (FFT) is a computationally-efficient implementation of DFT. If x is a vector containing $x[n]$ for $0 \leq n \leq M-1$ and $N > M$, then `X=fft(x,N)` computes N evenly spaced samples of the DTFT of x and stores these samples in the vector X. If $N < M$, then the **MATLAB** function `fft` truncates x to its first N samples before computing the DTFT, thus yielding incorrect values for the samples of the DTFT. The function `ifft` can be used to compute the IDFT efficiently. Also note that the fft function computes the DFT in the interval [0, 2π]. To reorder the samples in the interval [−π, π] you can use the function `fftshift`. You can use `abs` and `angle` functions to obtain the magnitude and phase of the samples.

4.2. FFT for Image Analysis

We can obtain fft of 2-dimensional signals using the the **MATLAB** function `fft2`. In this section we will use 2D fft for image analysis.

a) Display the fft magnitude and phase of **'T1.bmp'** and **'T2.bmp'**.

b) Swap the phases of two images with each other, i.e. replace the phase information of **'T1.bmp'** with that of **'T2.bmp'** and vice versa. Then use `ifft2` to recover each image from fft magnitude and phase. Display the results. Which one contains more information of an image, fft phase or magnitude? Why?

c) Plot the fft magnitude for different values of u_0 and u_1 in part 2-e. Explain the results.

d) Display the fft magnitude of the filtered images in part 3-c and compare them to the original image. By increasing filter size, what happens to fft magnitude of the image? Why?

5. Denoising

a) Load image 'T1_1.bmp' and Display its histogram using imhist. Now add Gaussian noise with a mean of 0.2 to the image using imnoise. Discuss the effect of adding Gaussian noise on the histogram of the original image.

b) Calculate and display the magnitude and phase of the fft of the noisy image corrupted with Gaussian noise. Discuss the differences of fft phase and magnitude with original image. What is the effect of increasing noise mean and variance on fft phase and magnitude?

c) Denoise images in part (b) using a Weiner filter (`wiener2`). Now calculate the fft of the denoised images and display their fft phase and magnitude. Compare fft phase and magnitude of these images with original image.

6. Sampling / Resolution

a) Write a function that down-samples an input image I with the factor of n, i.e. out of each n pixels of image, keeps one pixel. For example, if the input image is 256×256 pixels and we down-sample it with the factor of 2, it becomes 128×128 pixels.

b) Load image 'skin.jpg' and down-sample it with the factor of 5 using the function in part (a). Resize image to its original size using `imresize`. Compare two images and discuss the results. Also display the aliasing effect.

c) This time pre-filter the image using Gaussian filter first and then down-sample it. Compare the results with part (b). Increase SIGMA to see the effect of Gaussian filter better. Also discuss the result of down-sampling on the resolution of the image.

d) Write a function to up-sample an image with the factor of n (You can do this by repeating pixels, *i.e.*, making pixels bigger). Apply this function to 'skin.jpg' and compare the result with the original one. Also discuss whether up-sampling an image increases its physical resolution.

7. Contrast

a) Create a small white circle in the center of a gray background. Suppose this image goes through a system which its PSF is a 5x5 averaging filter. Display the output of the system. Calculate the modulation for the input and output image and also calculate the modulation ratio. Describe the effect of the system on contrast.

Hint: For creating circle use the following code

```
rect = uint8(zeros(800,800));
for x=1:800
    for y=1:800
      if((x-400)^2+(y-400)^2<2)
       rect(x,y)=255;
      else
       rect(x,y)=30;
      end
   end
end
```

b) Load image 'mammo_lc.jpg'. Display it using `imshow`. This is an x-ray mammogram and as you can see, the image has a low contrast. Sometimes it is possible to enhance the contrast of an acquired image through post-processing (this process is modality/application-dependent). Display the histogram of the loaded image using `imhist` (You may need to use rgb2gray to make the image 1-channel.) As it can be seen, most of the pixels have an intensity value somewhere between 90 and 150. Here we can enhance the contrast by increasing the range of intensity values form [90:150] to [0:255]. To do so we define a linear point function $g(f) = af + b$ which generates a new image B(i,j) out of image I(i,j), i.e. $B(i,j) = g(I(i,j))$. Design $g(f)$ in such a way that it increases range of intensity values and consequently, the image contrast. If p(f) is the histogram of input image (the left histogram in Figure 2), p(g) should be similar to the right histogram in Figure 2. This process is called contrast enhancement using linear histogram stretching. **Hint:** For arithmetic operations you may need to convert image data type to `double`. To display the image convert it back to `uint8`.

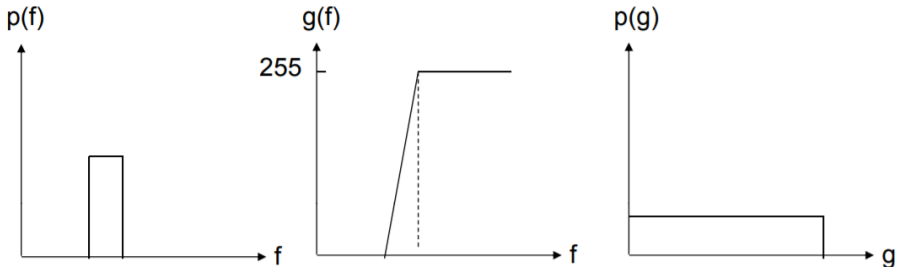

Figure 2 *- Contrast enhancement using linear stretching*

c) Display and discuss the effect of applying the point function $g(f)$ on the magnitude and phase of the Fourier transform of the image in parts (b).