İhsan Doğramacı Bilkent University
Electrical and Electronical Engineering Department

EEE391 Matlab Assignment 2

Assignment: 07/12/2018 Friday
Deadline: 28/12/2018 Friday (17:30)

You can leave your report to M. Anjum Qureshi (EE207/EE212) on 28/12/2018 between 15:40- 17:30. You can also submit your homework earlier than this deadline if you see one of the course TAs. Please do not leave your homework to a mailbox or a TA's desk and make sure that you submit your homework to one of the course TAs.

In addition to turning in your handwritten assignment, please e-mail the source code eksioglu@ee.bilkent.edu.tr with the subject "EEE391 MA2 Submission" until deadline. Source file should be named `EEE391_YOURBILKENTID_MA2_SRC.m`. Don't forget to replace `YOURBILKENTID` with your actual Bilkent ID in the file names.

In this assignment, you will do some signal processing operations on an image. First, download the associated zip file at `http://kilyos.ee.bilkent.edu.tr/~eee391h/EEE391_1819_MA2.zip` and extract the files to a folder in your computer. Then run Matlab and choose your Current Folder as the folder that the files are extracted.

In the zip file, you will see a .png file and two .m files. First, load the PNG file by typing

```
image_rgb = imread('fruits.png');
image = im2double(rgb2gray(image_rgb));
```

where the output includes a matrix whose elements represent a grayscale image. Display the image using the following code

```
imshow(image, [])
```

Include this image to your report. Note that, `[]` sign is required to be used whenever you use imshow command, as it allows you to display the image in full scale. Then, compute the 2D Fourier transform of this image using

```
FT_image = FourierTransform(image)
```

where `FourierTransform` is an m file which is provided to you to compute the 2D Fourier transform of an input signal. Note that, although the Fourier transform is a continuous operation, here we have the sampled version of it. The variable `FT_image` contains the samples of the 2D Fourier transform of the image in the frequency interval $\hat{\omega}_x \in [-\pi, \pi)$ and $\hat{\omega}_y \in [-\pi, \pi)$. Let the size of the image be $M \times N$, where M is the length along the vertical dimension and N is the length along the horizontal direction. Also, let the Matlab indices along vertical and horizontal dimensions be m and n, respectively. So, for the Fourier transforms, $m = 1$ corresponds to $\hat{\omega}_y = -\pi$ rad along the vertical dimension and $n = 1$ corresponds to $\hat{\omega}_x = -\pi$ along the horizontal dimension. Similarly, $m = M$ corresponds to $\hat{\omega}_y = \pi - 2\pi/M$ rad and $n = N$ corresponds to $\hat{\omega}_x = \pi - 2\pi/N$ rad along vertical and horizontal dimensions, respectively.

- What is the corresponding 2D frequency along vertical and horizontal dimensions for an arbitrary $[m, n]$ pair? Show your calculations.
- Display the magnitude and phase of `FT_image` in two separate images as grayscale images and include these to your report.

1 High-Pass & Low-Pass Filters

Now you will generate some filters in the Fourier domain, where the sizes of the filters are $M \times N$, which represents the continuous frequency interval $\hat{\omega} \in [-\pi, \pi)$.

- Generate a low-pass filter, LP, defined as:

$$LP = \begin{cases} 1 & \text{if } \frac{-\pi}{4} \leq \hat{\omega}_x \leq \frac{\pi}{4}, \frac{-\pi}{4} \leq \hat{\omega}_y \leq \frac{\pi}{4} \\ 0 & \text{otherwhise} \end{cases}$$

Now, you will apply these filters to the image in the Fourier domain. That is, you are going to multiply LP and `FT_image`. Then, by using InverseFourierTransform command, you will generate the filtered images. To do this, you can use the following code:

```
Filtered_Image = InverseFourierTransform(LP.*FT_image);
imshow(real(Filtered_Image),[])
```

where, real operator is included to discard negligibly small imaginary parts of the output images, that are the results of the computational errors. Include the images to your report and make your comments on the effect of the filter. Discuss how output image would change according to changes in the bandwidth of the filter.

Now you will generate following high-pass filter, **using the low-pass filter that you generated**.

$$HP = \begin{cases} 0 & \text{if } \frac{-\pi}{4} \leq \hat{\omega}_x \leq \frac{\pi}{4}, \frac{-\pi}{4} \leq \hat{\omega}_y \leq \frac{\pi}{4} \\ 1 & \text{otherwhise} \end{cases}$$

Then, apply this high-pass filter to the image in the Fourier domain and include the filtered image in the space domain to your report. Make your comments on the output images. Discuss how the output image would change if you changed bandwidth of the filter.

2 Custom Filters

In this chapter, you'll apply filters on the space domain (not the frequency domain), by using 2D convolution. 2D convolution operation can be defined as following:

$$f[y,x] * g[y,x] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[y,x] g[y-m, x-n]. \tag{1}$$

Write a function that can calculate 2D convolution. Note that you cannot use `conv2` or similar ready-built convolution functions, but you can check your function by comparing the outputs. You may want to visit `https://github.com/vdumoulin/conv_arithmetic` or `https://arxiv.org/pdf/1603.07285.pdf` (page 7) about how convolution arithmetic works with 2D inputs and filters.

Create two simple filters with impulse responses h_x and h_y, which are defined as:

$$\begin{aligned} h_x[m,n] &= -\delta[m, n-1] + \delta[m, n+1] \\ h_y[m,n] &= -\delta[m-1, n] + \delta[m+1, n]. \end{aligned} \tag{2}$$

Let A denote your input image, calculate $A * h_x$ and $A * h_y$. Display and discuss the resulting images. Then calculate $h_{x,y} = h_x * h_y$ analytically, and write the equation for $h_{x,y}$ similar to (2). Calculate $A * h_{x,y}$, display and discuss the resulting image. Finally calculate $A * h_x * h_y$ by applying the filters on the input image consecutively. Display and discuss the result. Do not forget to add the output images to your report.

3 Sobel Operator

In this part, you'll create a Sobel Operator. First, create two filters with following impulse responses:

$$\begin{aligned} h_x[m,n] &= \delta[m-1,n-1] - \delta[m-1,n+1] + \\ &+ 2\delta[m,n-1] - 2\delta[m,n+1] + \delta[m+1,n-1] - \delta[m+1,n+1] \\ h_y[m,n] &= \delta[m-1,n-1] + 2\delta[m-1,n] + \\ &+ \delta[m-1,n+1] - \delta[m+1,n-1] + -2\delta[m+1,n] - \delta[m+1,n+1] \end{aligned} \tag{3}$$

Then, calculate $G_x = h_x * A$ and $G_y = h_y * A$. Finally calculate $G = \sqrt{G_x^2 + G_y^2}$ and display G. Discuss the result and also comment on differences between the this output and the output image in Section 2. Add the output image to your report.