In this lab, you will implement Gaussian smoothing and edge detection using first and second order derivative filters to grayscale images.

Important Note: You should complete the lab until the end of the lab hours and submit all your codes to SUCourse as a single zip file. Deadline for in-lab code submission to SUCourse is **15:30**.

Things to do:

Your functions must be as generic as possible, i.e., don't make any assumptions about the size, the type and the colors of the images. Your functions must convert the image to grayscale if it is colored and you must employ the row and column numbers of the images as variables.

Linear Filtering

- **Gaussian Filtering:** *Gaussian* filtering is a linear operation that involves the local convolution of a given image with a filter kernel of samples of the 2D Gaussian function. Gaussian filtering is used for smoothing the pictures by eliminating "outliers" in image values which are considered to be noise in a given context. In order to apply Gaussian smoothing you can use the kernel below to convolve it with the given image.

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

/273

Now write a function which takes the image in Figure 1 as the input and returns the "**Gaussian smoothed**" version of the image. You will use 5×5 approximated Gaussian filter kernel given above. Your function name should be "lab2gaussfilt.m".

Figure 1: Original image

Edge Detection

First Order Derivative Filters

- **Prewitt Operator:** *Prewitt* filtering is a discrete 2D 1^{st} order derivative operation which can be applied with the following kernels

-1	0	1
-1	0	1
-1	0	1

X Filter

-1	-1	-1
0	0	0
1	1	1

Y Filter

To detect edges in a grayscale image, the gradient image obtained by the horizontal and the vertical Prewitt operators is binarized by a threshold. The gradient image is calculated as

$$G(p) = \sqrt{G_x(p)^2 + G_y(p)^2} \tag{1}$$

> Now write a function which takes the image shown in Figure 2 and a threshold value as inputs and utilize "**Prewitt filters**" to return a binary image of detected edges. Your function name should be "lab2prewitt.m".

- **Sobel Operator** *Sobel* filtering is a discrete 2D 1^{st} order derivative operation which can be applied with the following kernels

-1	0	1
-2	0	2
-1	0	1

X Filter

-1	-2	-1
0	0	0
1	2	1

Y Filter

To detect edges in a grayscale image by employing Sobel operator, the same procedure is applied as Prewitt based edge detection except the kernels are changed.

> Now write a function which takes the image shown in Figure 2 and a threshold value as inputs and utilize "**Sobel filters**" to return a binary image of detected edges. Your function name should be "lab2sobel.m".

Figure 2: Original image

Second Order Derivative Filter

- **Laplacian of Gaussian Smoothed Image:** In this problem we will detect edges using *Laplacian of Gaussian* (LoG) filter. As Laplace operator may detect edges as well as noise, it is desirable to smooth the image first by a Gaussian filter. Applying the Laplacian to a Gaussian filtered image can be done in one step of convolution with the following kernel

0	1	0
1	-4	1
0	1	0

Different from the Sobel based edge detection algorithm, zero crossing points represent the edge pixels. Extract the gradient profile from a line segment of the LoG filtered image belonging to an edge region and investigate the zero crossing behavior as shown in the results below.

Now write a function which takes the original image in Figure 3 as the input and utilize "**Laplacian of Gaussian**" to return the LoG filtered image. Show the LoG filtered image along with a sample gradient profile from a pre-selected line segment of the filtered image. Your function name should be "lab2log.m".

Your results should look as follows:

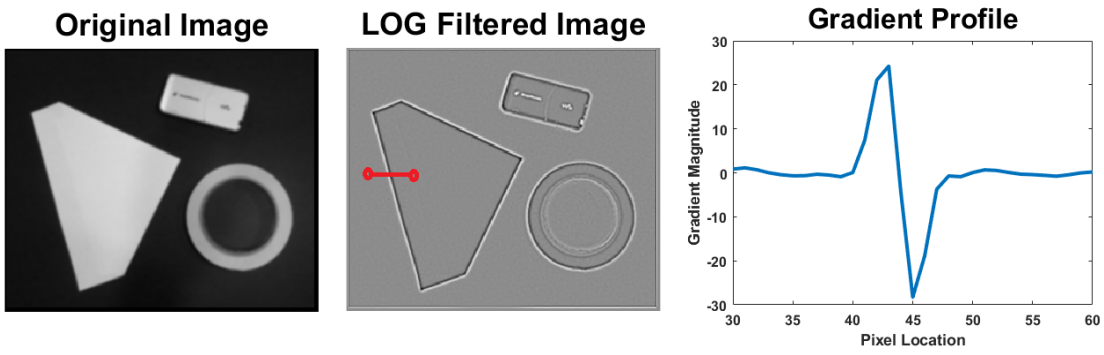

Figure 3: LoG filtering results

Post Lab

Post lab reports must include brief explanations of the functions that you implemented in this lab.

- Provide resulting images by utilizing all these functions with different parameters such as window size. Compare different filters and discuss your results.

Deadline for post lab report submission to SUCourse: **31 October 2022, 23:55**.