

# Lab

## Edge Detection techniques

### Objective:

- Apply and notice the differences between edge detection techniques
- Understand the effect of different parameters used in edge detection techniques.
- Learn and implement “Sobel operator” and “LOG” edge detection techniques.

### Experiment 1: apply different built in edge detection techniques

In this exercise we experiment [canny – sobel –LOG- Roberts and Prewitt] edge detection functions and test the effect of their parameters.

- Read the image “circuit.tif” then convert it to gray level.
- Apply these edge detection techniques.
- Choose proper thresholds/sigma and other parameters and report the effect of each one and state the difference of each technique.
  - ✓ ☐ “Prewitt” edge detection (prewitt)
  - ✓ ☐ “Roberts” edge detection (roberts)
  - ✓ ☐ “sobel” for horizontal edges (sobel\_h)
  - ✓ ☐ “sobel” for vertical edges (sobel\_v)
  - ✓ ☐ “sobel” for vertical and horizontal edges (sobel)
  - ✓ ☐ “Canny” edge detection (canny)

## Experiment 2: implement “sobel” edge detection

In this exercise we make our own implementation for the sobel edge detection technique, and comment on the results.

### Sobel Algorithm Steps

✓ 1- Create the two filters hx and hy

$$h_y = \begin{Bmatrix} -1, 0, 1 \\ -2, 0, 2 \\ -1, 0, 1 \end{Bmatrix}$$

$$h_x = \begin{Bmatrix} -1, -2, -1, \\ 0, 0, 0 \\ 1, 2, 1 \end{Bmatrix}$$

2- for every 3\*3 window of the image, multiply the two filters each at once, store the new result for the central pixel in two new images, one for the x value and one for the y value.

3- create a new image and fill it with the final value for edge from x and y filters together according to the following equation ( edgestrength =  $\sqrt{x^2 + y^2}$  )

4- Apply threshold to the edge strength, in which edges above this threshold will be visible and others will disappear.

5- Print the image resulted from the (x filter only), (y filter only) and (from x and y filter) together. ✓ ✓ ✓

## Experiment 3: implement “LOG” edge detection

In this exercise we make our own implementation for the LOG edge detection technique, and comment on the results.

### LOG Algorithm Steps

1- Apply “Gaussian filter on the image first to remove the noise with suitable threshold.

2- Create two types of LOG filters f1 and f2

$$\begin{array}{ll} F1 = \begin{Bmatrix} -1, -1, -1 \\ -1, 8, -1 \\ -1, -1, -1 \end{Bmatrix} & f2 = \begin{Bmatrix} 0, 1, 0, \\ 1, -4, 1 \\ 0, 1, 0 \end{Bmatrix} \end{array}$$

3- for every 3\*3 window of the image, multiply the two filters each at once, store the new result for the central pixel in two new images, one for the F1 value and one for the F2 value.

4- Apply threshold to the edge strength, in which edges above this threshold will be visible and others will disappear.

5- Print the image resulted from the (F1 filter) and (F2 filter).





Name	Attribute or Function	Usage
np.sqrt	Function	Get square root of the matrix element-wise.