

# Computer Vision Homework 9

## How to run it

```
python image.py lena.bmp
```

## Principal code fragment

```
def L2NormMagnitude(neighbors, masks, threshold):
    num = len(masks)
    sizeX = len(masks[0])
    sizeY = len(masks[0][0])
    magnitude = []

    for i in xrange(num):
        r = 0
        for row in xrange(sizeY):
            for col in xrange(sizeX):
                r += neighbors[col][row]*masks[i][col][row]
            magnitude.append(r**2)
    return (math.sqrt(sum(magnitude)) > threshold)

def MaxMagnitude(neighbors, masks, threshold):
    num = len(masks)
    sizeX = len(masks[0])
    sizeY = len(masks[0][0])
    magnitude = []

    for i in xrange(num):
        r = 0
        for row in xrange(sizeY):
            for col in xrange(sizeX):
                r += neighbors[col][row]*masks[i][col][row]
            magnitude.append(r)
    return (max(magnitude) > threshold)

def robertDetector(origin, image, threshold):
    masks = [[[-1, 0], [0, 1]], [[0, -1], [1, 0]]]
    neighbors = []
    x = origin[0]
    y = origin[1]
    neighbors.append([ getValue(image,x,y), getValue(image,x+1,y) ])
    neighbors.append([ getValue(image,x,y+1), getValue(image,x+1,y+1) ])
    return BLACK if L2NormMagnitude(neighbors, masks, threshold) is True else WHITE

def prewittDetector(origin, image, threshold):
    masks = [[[-1, -1, -1], [0, 0, 0], [1, 1, 1]], [[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]]
    neighbors = getNeighbors( image, origin, [3, 3])
    return BLACK if L2NormMagnitude(neighbors, masks, threshold) is True else WHITE

def sobelDetector(origin, image, threshold):
    masks = [[[-1, -2, -1], [0, 0, 0], [1, 2, 1]], [[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]]
    neighbors = getNeighbors(image, origin, [3, 3])
    return BLACK if L2NormMagnitude(neighbors, masks, threshold) is True else WHITE

def freichenDetector(origin, image, threshold):
    sqrt2 = math.sqrt(2)
    masks = [[[-1, -sqrt2, -1], [0, 0, 0], [1, sqrt2, 1]], [[-1, 0, 1], [-sqrt2, 0, sqrt2], [-1, 0, 1]]]
    neighbors = getNeighbors(image, origin, [3, 3])
    return BLACK if L2NormMagnitude(neighbors, masks, threshold) is True else WHITE
```

## Description

implement Robert, Prewitt, Sobel, Frei & Chen, Kirsch, Robinson, and Nevatia-Babu's edge detectors with these threshold values:

- Robert's Operator: 12
- Prewitt's Edge Detector: 24
- Sobel's Edge Detector: 38
- Frei and Chen's Gradient Operator: 30
- Kirsch's Compass Operator: 135
- Robinson's Compass Operator: 43
- Nevatia-Babu 5x5 Operator: 12500

## Result



Robert's Operator: 12



Prewitt's Edge Detector: 24



Sobel's Edge Detector: 38



Frei and Chen's Gradient Operator: 30



Kirsch's Compass Operator: 135



Robinson's Compass Operator: 43



Nevatia-Babu 5x5 Operator: 12500