**Shri Ramdeobaba College of Engineering and Management, Nagpur**

**Department of Electronics Engineering**

**Digital Image Processing (ENT 355-3)**

**Name:** Prajwal Pandurang Shette

**Roll No:** B1- 12

**Experiment No: 09**

**Aim:** Write a python codeCanny edge detection

**Theory**: Canny Edge Detection is a popular edge detection algorithm. It was developed by John F. Canny in

1. It is a multi-stage algorithm and we will go through each stages.
2. **Noise Reduction**

Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter. We have already seen this in previous chapters.

1. **Finding Intensity Gradient of the Image**

Smoothened image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction ( Gx) and vertical direction ( Gy). From these two images, we can find edge gradient and direction for each pixel as follows:

Edge\_Gradient(G)=G2x+G2y−−−−−−−√Angle(θ)=tan−1(GyGx)

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

**Code:**

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

img = cv.imread('lake.jpg',0)

edges = cv.Canny(img,200,250)

plt.subplot(121),

plt.imshow(img,*cmap* = 'gray')

plt.title('Original Image'),

plt.xticks([]), plt.yticks([])

plt.subplot(122),

plt.imshow(edges,*cmap* = 'gray')

plt.title('Edge Image'),

plt.xticks([]), plt.yticks([])

plt.show()

**Output:**

****

**Conclusion:** OpenCV puts all the above in single function, **[c](https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html" \l "ga04723e007ed888ddf11d9ba04e2232de" \o "Finds edges in an image using the Canny algorithm  . )**[v.Canny()](https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html" \l "ga04723e007ed888ddf11d9ba04e2232de" \o "Finds edges in an image using the Canny algorithm  . ). We will see how to use it. First argument is our input image. Second and third arguments are our minVal and maxVal respectively. Fourth argument is aperture\_size. It is the size of Sobel kernel used for find image gradients