**Assignment 4**

**Recognize circles and squares with the formula for roundness**

With the OpenCV, I found the method called “Canny Edge Detection”. This theory is an edge detection algorithm which was developed by John F. Canny in 1986. There are many stages of this method:

* Function canny\_edge\_detector: Convert image to grayscale, remove noise, detect edge.

Text

Description automatically generated

* Function compute\_grayscale: will be called in canny\_edge\_detector. Transfer the input to grayscale image

Text

Description automatically generated

First, noise reduction uses for removing to the noise in the image with a 5x5 Gaussian filter. Function compute\_blur: will be called in canny\_edge\_detector to blur the image and remove noise.

Text

Description automatically generated

* Second, finding intensity gradient of the image. After smooth the image, the image is filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in two directions. Function compute\_gradient will return edge gradient and direction for each pixel:

A picture containing text

Description automatically generated

Text

Description automatically generated

Third, non-maximum suppression is a full scan of image to remove any unwanted pixels which may not constitute the edge. The result of this stage is a binary image with thin edges.

A screenshot of a computer

Description automatically generated with medium confidence

Final, hysteresis thresholding is the stage deciding which are edges or not. This stage is important in choosing the minVal and max Val. Any edges with intensity gradient more than maxVal are sure to be edges and below the minVal are sure to be non-edges. Also consider the edges lie between based on their connection. Function filter\_strong\_edges removes small pixels noises and the final result is strong edges in the image.

Text

Description automatically generated

Import necessary library and open image. Create an output image to save the result

Text

Description automatically generated

Determine the maximum and minimum radius of circle can have. Using the function canny\_edge\_detector with the input image as parameter. Draw the border line at the circles with are detected.

Text

Description automatically generated

The result:

Bubble chart

Description automatically generated with low confidence

Detecting circles and squares by using OpenCV:

Read and convert the image into grayscale mode. Finding the edges around the object become easy when we work with the grayscale image.

Text

Description automatically generated

Find the threshold by using function threshold, this function finds out the threshold frequency of the gray image. 240 is the threshold value, and 250 is the maximum threshold value. Based on the thresholds the function will find out all the contours present in the grayscale image. The contours are the boundaries of the object or the continuous line around an object.

Graphical user interface, text

Description automatically generated

Using for loop for every contour and detect the shape. The function approPoLyDP() return s all polygons curve based on the contour with precision. True parameters specify close contour and curve. This function returns the approximates curves. Ravel()[0] return the x coordinates of the contour and ravel()[1] return y coordinates of the contour. The len() function can find out the total number of curves present in the close loop. With the square we have specified with 4 curves or edges. For the squares, the width and height aspect ratio is 1.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Shape, square

Description automatically generated

Shape

Description automatically generated

**Refereneces:**

https://opencv24-python-tutorials.readthedocs.io/en/latest/py\_tutorials/py\_imgproc/py\_canny/py\_canny.html