REPORT OF VR ASSIGNMENT-1

IMPLEMENTATION OF LINE DETECTION AND CIRCLE DETECTION ALGORITHM USING HOUGH TRANSFORMATION

Name:-Hussnain Ashraf
Roll No:- MT2021055
Semester :- 2nd

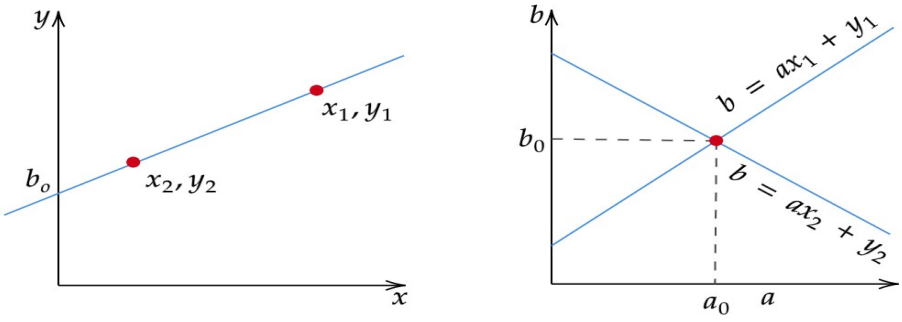
What Actually The Hough Transformation is :-

Hough transform is a feature extraction method for detecting simple shapes such as circles, lines etc in an image.

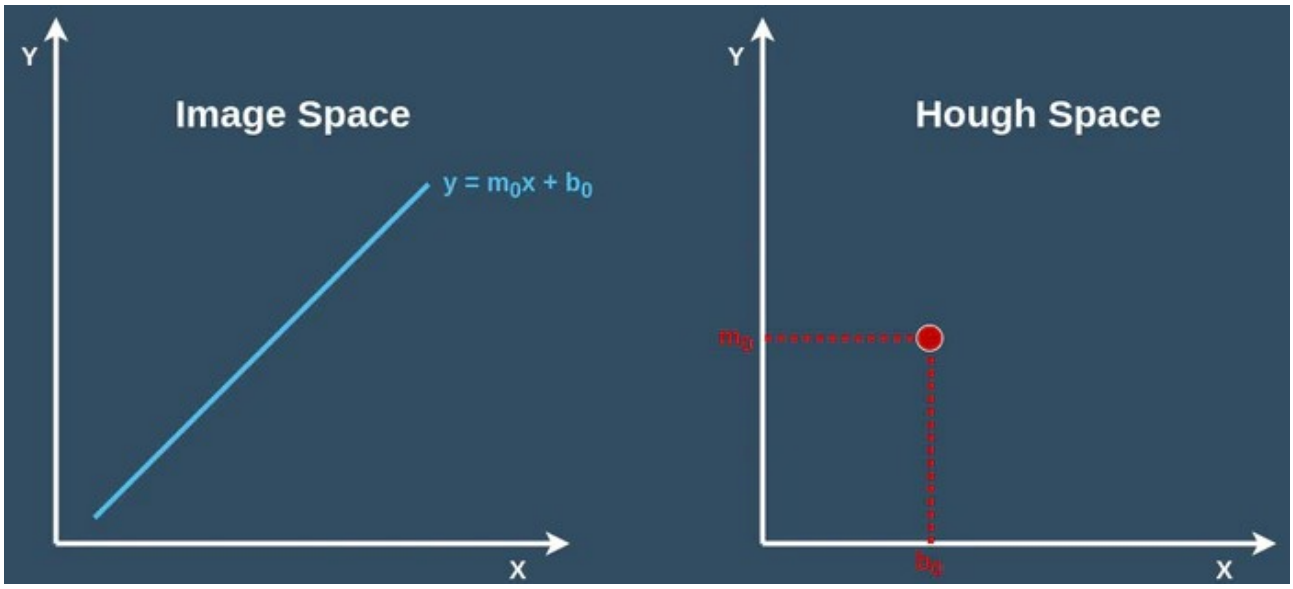
A simple shape is one that can be represented by only a few parameters. For example, a line can be represented by two parameters (slope, intercept) and a circle has three parameters — the coordinates of the center and the radius (x, y, r). Hough transform does an excellent job in finding such shapes in an image.

The main advantage of using the Hough transform is that it is insensitive to occlusion.

The Hough Space and the Mapping of Edge Points onto the Hough Space:-



The **Hough Space is a 2D plane that has a horizontal axis representing the slope and the vertical axis representing the intercept of a line** on the edge image. A line on an edge image is represented in the form of **y = ax + b** . One line on the edge image produces a point on the Hough Space since a **line is characterized by its slope a and intercept b.** On the other hand, an edge point (x_i,y_i) on the edge image can have an infinite number of lines pass through it. Therefore, **an edge point produces a line in the Hough Space in the form of $b = ax_i + y_i$.** In the Hough Transform algorithm, the Hough Space is used to determine whether a line exists in the edge image.

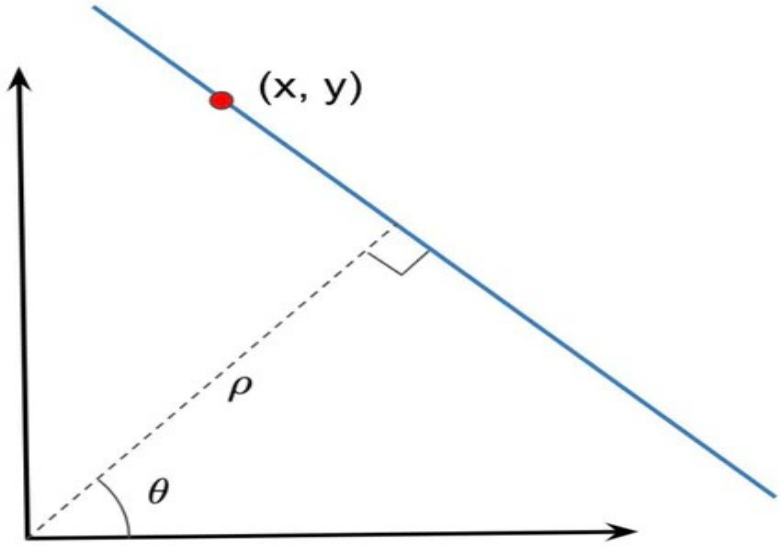


But we have a problem though, with `y = mx + b`, we cannot represent a vertical line, as the slope is infinite. So we need a better way parametrization, polar coordinates (rho and theta).

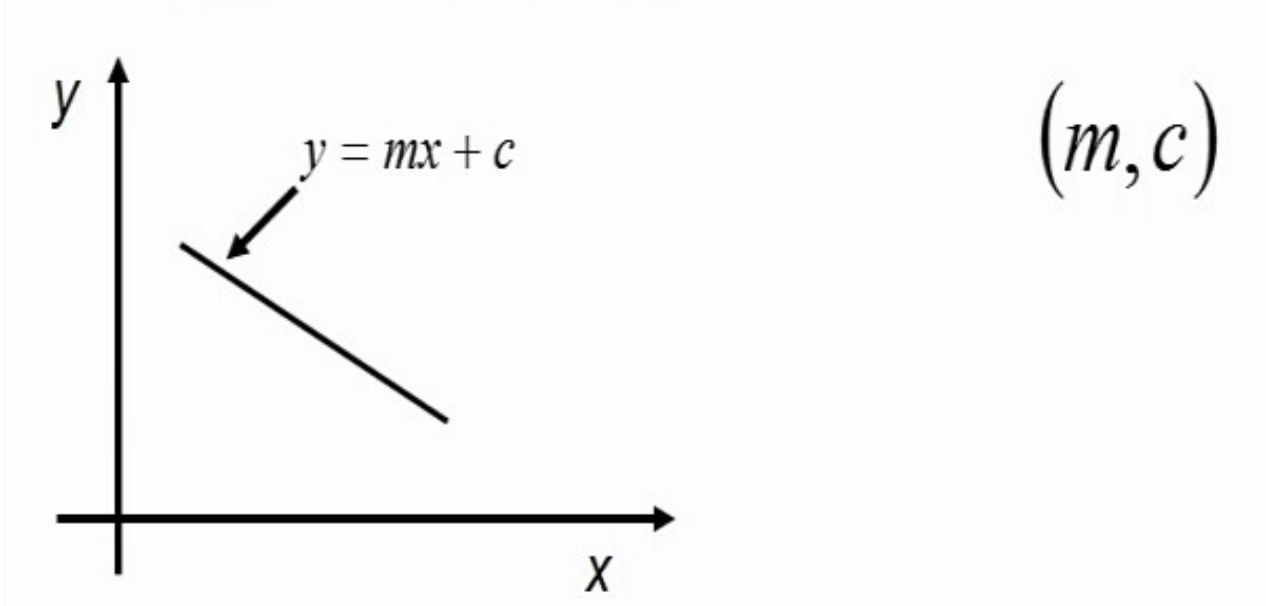
Hough space:-

rho: describes the distance of the line from the origin

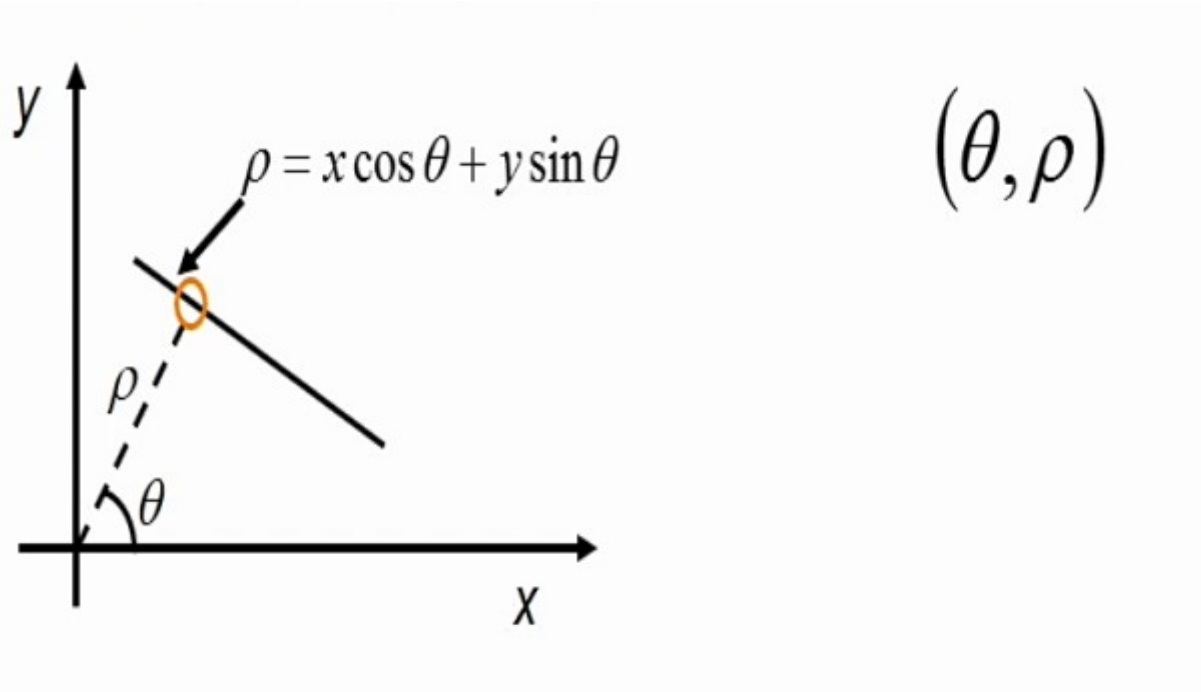
theta: describes the angle away from horizontal



We can use this transform to measure the length of a line. For example, suppose a line in an image needs to be detected, it can do that.



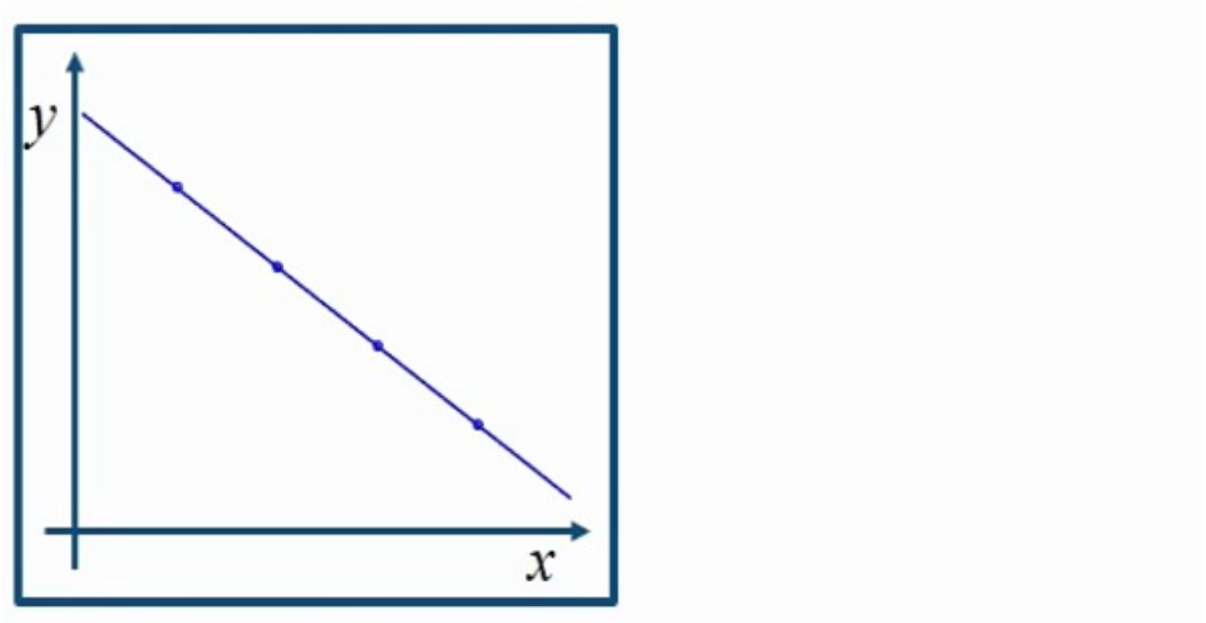
The general expression of a line is y=mx+c. It means that you can map a line into a coordinate pair, m, c. However, vertical lines have a problem, since the slope value is unbounded. In this case, the hough transform uses a different parametric representation. **The parameters are θ and ρ. Here, θ is the angle between the axis and the origin line connecting that closest point. ρ corresponds to the distance from the origin to the nearest point on the straight line.**



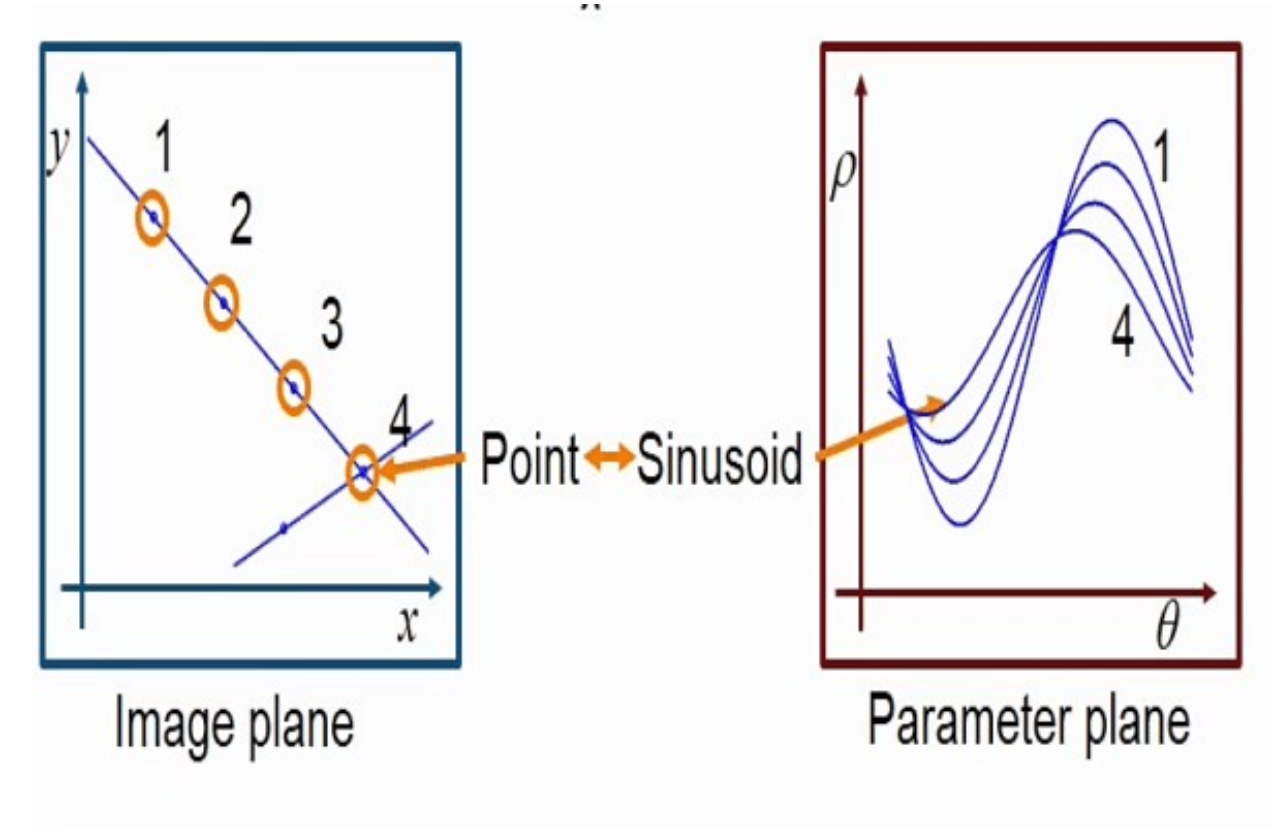
Conceptually, hough transform is the mapping of image coordinate in the x and y into the parametric coordinate θ,ρ

Using hough transform to detect a line:-

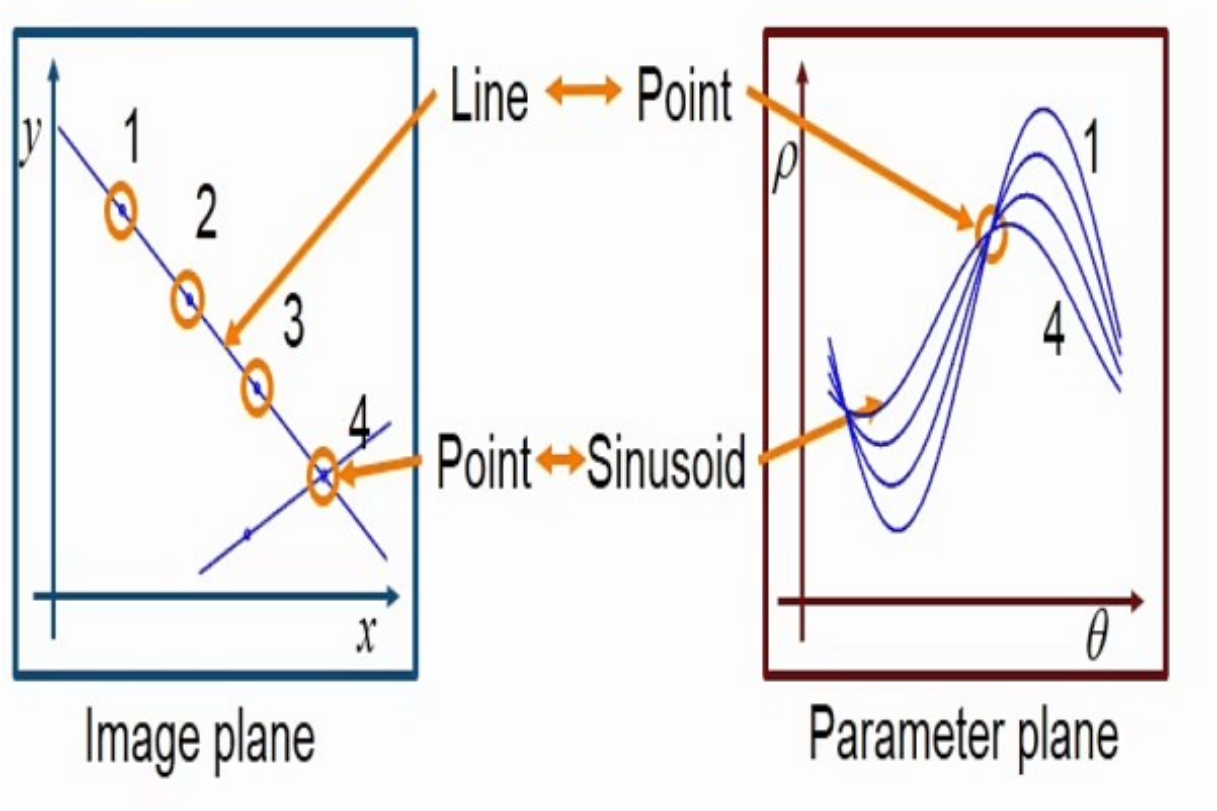
Suppose an image consists of a line as shown below;



A line is made up of multiple points. Similarly, multiple lines can pass through the same point. **We visualize multiple lines in a hough transform plane where the parameters θ are on the x-axis and ρ on the y-axis**. Every point on the image is transformed into a sinusoid. A sinusoid is a signal in the shape of a sine wave.



It makes sense because many lines may pass through a given point in an image plane. **It translates into a sinusoid in the hough plane. If two or more of these lines coincide to form a line in the image plane, they will intersect in the hough transform plane**. This intersection corresponding to the θ,ρ pair corresponds to the detected line.



The Algorithm:-

1. **Decide on the range of ρ and θ. Often, the range of θ is [0, 180] degrees and ρis [-d, d] where d is the length of the edge image's diagonal.** It is important to quantize the range of ρ and θ meaning there should be a finite number of possible values.
2. **Create a 2D array called the accumulator representing the Hough Space with dimension (num_rhos, num_thetas)** and initialize all its values to zero.

3. Perform edge detection on the original image. This can be done with any edge detection algorithm of your choice.

4. For every pixel on the edge image, check whether the pixel is an edge pixel. If it is an edge pixel, loop through all possible values

of θ, calculate the corresponding ρ, find the θ and ρ index in the accumulator, and increment the accumulator base on those index pairs.

5. Loop through all the values in the accumulator. If the value is larger than a certain threshold, get the ρ and θ index, get the value of ρ and θ from the index pair which can then be converted back to the form of **y = ax + b.**

In Summary what actual the algorithm is :-

Extract edges of the image How ? using Canny

1. initialize parameter space rs, thetas
2. Create accumulator array and initialize to zero
3. for each edge pixel
4. for each theta
5. calculate r = x cos(theta) + y sin(theta)
6. Increment accumulator at r, theta
7. Find Maximum values in accumulator (lines)
   Extract related r, theta

I have also implementedf the Canny image detector from scratch by the help of googling the stuffs and from youtube :-

Implementing Canny Edge Detector:-

The basic steps involved in this algorithm are:

•Noise reduction using Gaussian filter

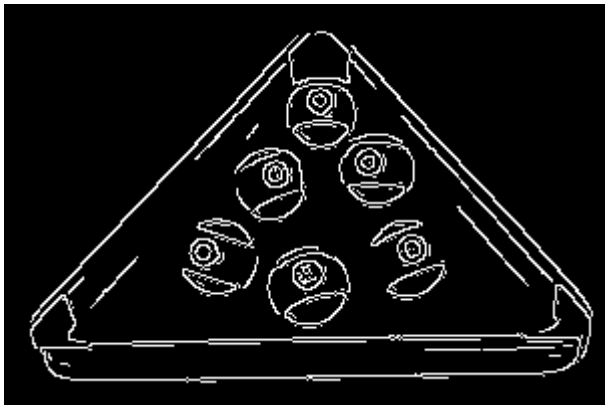
•Gradient calculation along the horizontal and vertical axis

•Non-Maximum suppression of false edges

•Double thresholding for segregating strong and weak edges

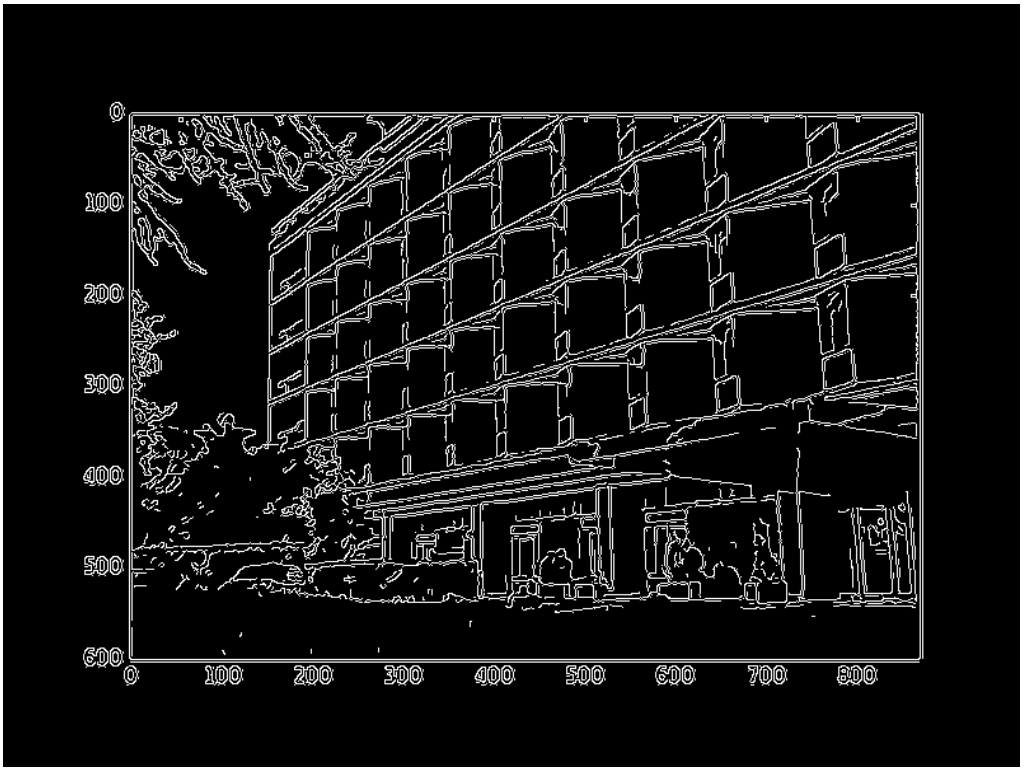
•Edge tracking by hysteresis

Coding part implementation of canny edge detection algorithm:-

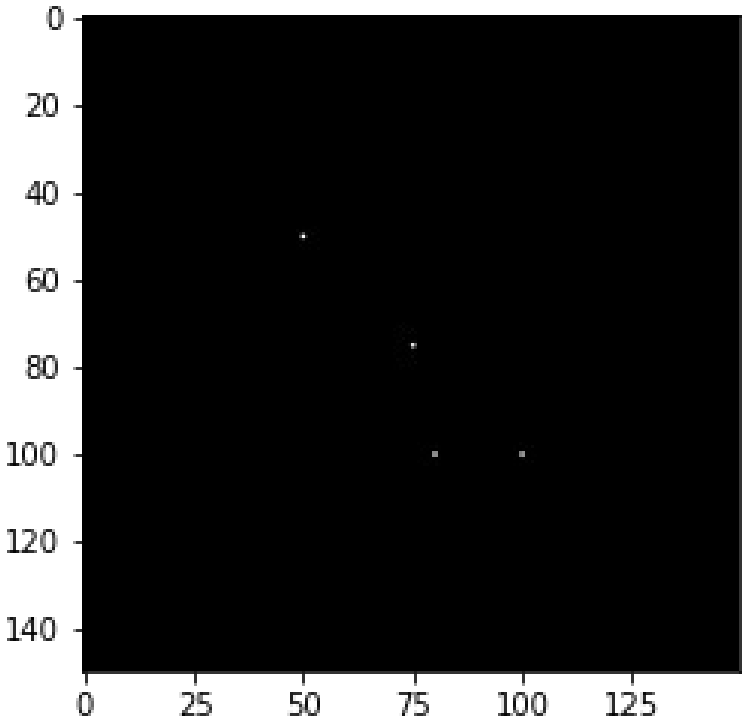


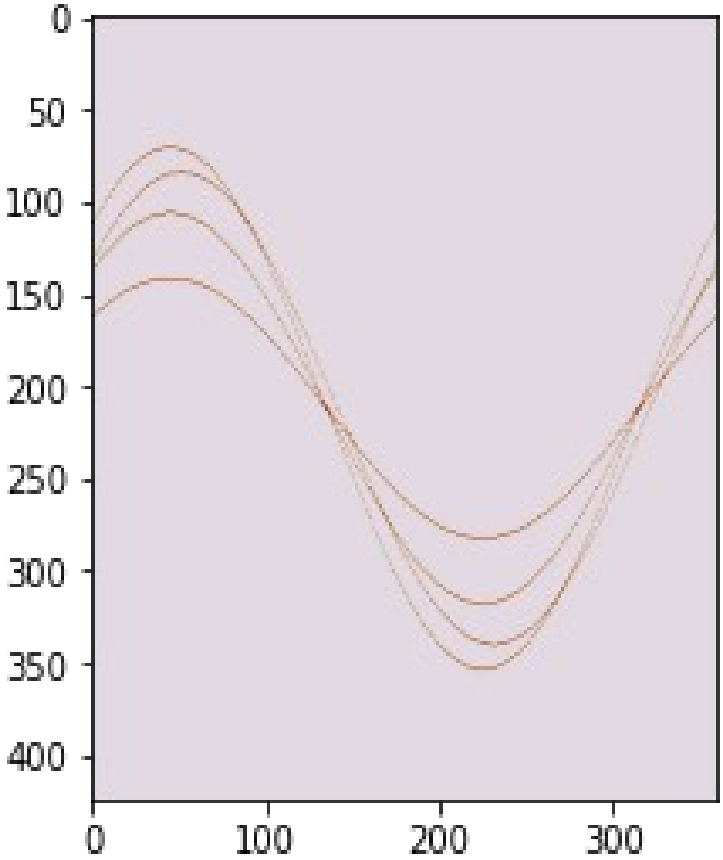
Another example :-


0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

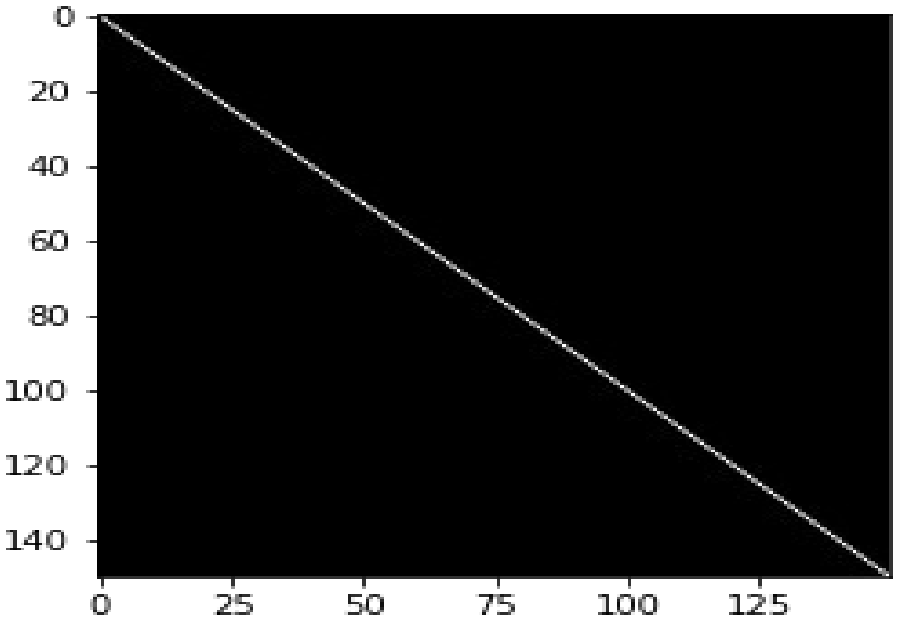

0
100
200
300
400
500
600
0
100
200
300
400
500
600
700
800

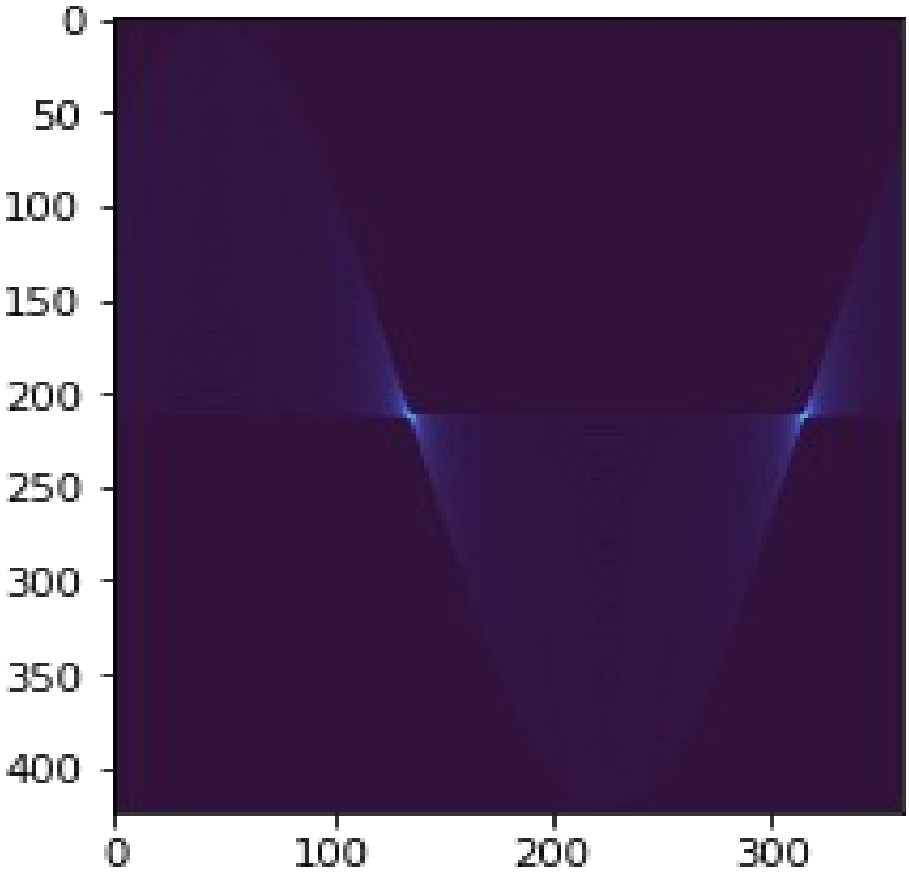
Houghline implementation example which i have done in my coding :-



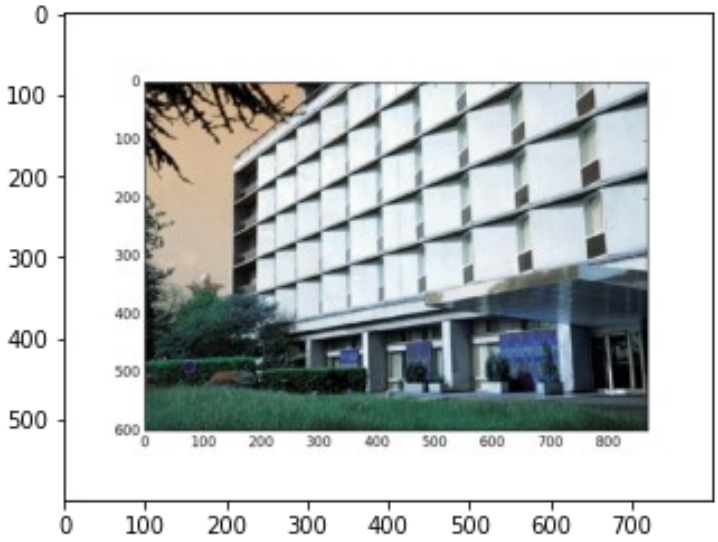


IMPLEMENTATION ON STARIGHT LINE NOW AFTER IMPLEMENTING ON POINTS:-

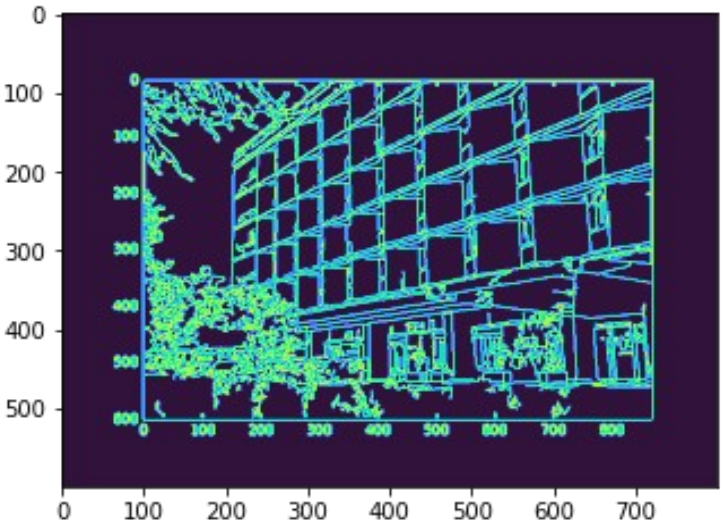




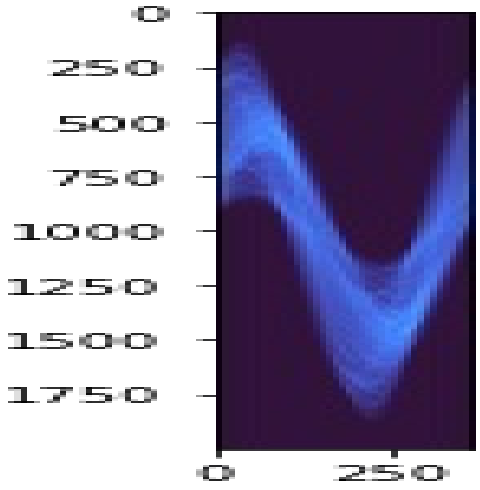
INPUT ORIGINAL IMAGE:-



CANNY EDGE DETECTION ON ORIGINAL IMAGE:-



COSINE CURVE INTERSECTION OF ORIGINAL IMAGE AFTER DETECTION OF STARIGHT LINES:-



Advantages & Disadvantages :-

The advantage of the Hough transform :

is that pixels lying on one line need not all be contiguous. This can be very useful when trying to detect lines with short breaks in them due to noise, or when objects are partially occluded

The disadvantages of the Hough transform:

• It can give misleading results when objects happen to be aligned by chance. • Detected lines are infinite lines described by their (m,c) values, rather than finite lines with defined end points.

IMPLEMENTING CIRCLE DETECTION ALGORITHM USING HOUGH TRANSFORM:-

The Hough Circle Transform works in a roughly analogous way to the Hough Line Transform explained

- In the line detection case, a line was defined by two parameters (r,θ)
- In the circle case, we need three parameters to define a circle:

C:(xcenter ycenter,r) where (xcenter,ycenter)define the center position and r is the radius, which allows us to completely define a circle.

The logic behind Hough Circle Transform algorithm comes from the mathematical expression of a circle in polar coordinate system as:

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

where,

$(x_0 \text{ and } y_0)$ *is the center and* r *is the radius.*

We need to transform our 2 dimensional input edge image E(x,y) to 3 dimensional Accumulator matrix(x0,y0,r).

The general equation of a circle is as follows:

$(x - a)\ 2 + (b - y)\ 2 = r\ 2$

Using the basics of trigonometry and a given radius, any point on a circle can be

$$a = x - Rcos(t)$$

$$b = y - Rsin(t)$$

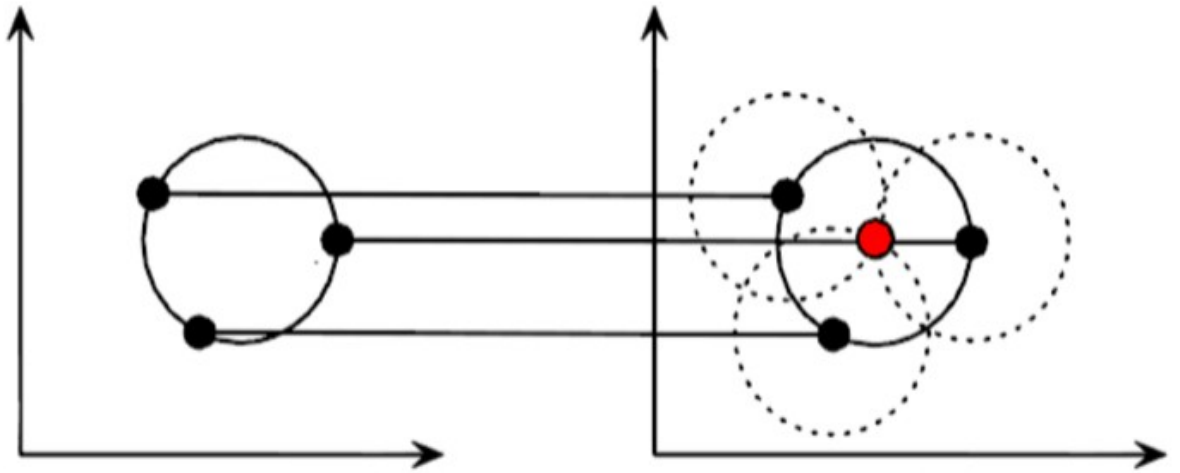
Where t is changes from 0 to 360 degrees and (a,b) are any one point around the circle of radius R centered at (x,y). Below figure shows the derivation of a and b given above using the unit circle and the Pythagorean Theorem. Knowing the center (x,y) and the radius, any circle can be drawn from those equations.

ALGORITHM:-

Below are the steps of the algorithm:-

- **Initializing the Accumulator Matrix**: Initialize a matrix of dimensions rows * cols * maxRadius with zeros.
- **Pre-processing the image:** Apply blurring, grayscale and an edge detector on the image. This is done to ensure the circles show as darkened image edges.
- **Looping through the points:** Pick a point on the image.
- **Fixing r and looping through a and b:** Use a double nested loop to find a value of r, varying a and b in the given range

```
for a in range(rows):
    for b in range(cols):
        r = math.sqrt((xi - a)**2 + (yi - b)**2)
        accum_matrix[a][b][r] += 1
```



Each point in geometric space (left) generates a circle in parameter space (right). The circles in parameter space intersect at the (a, b) that is the center in geometric space.

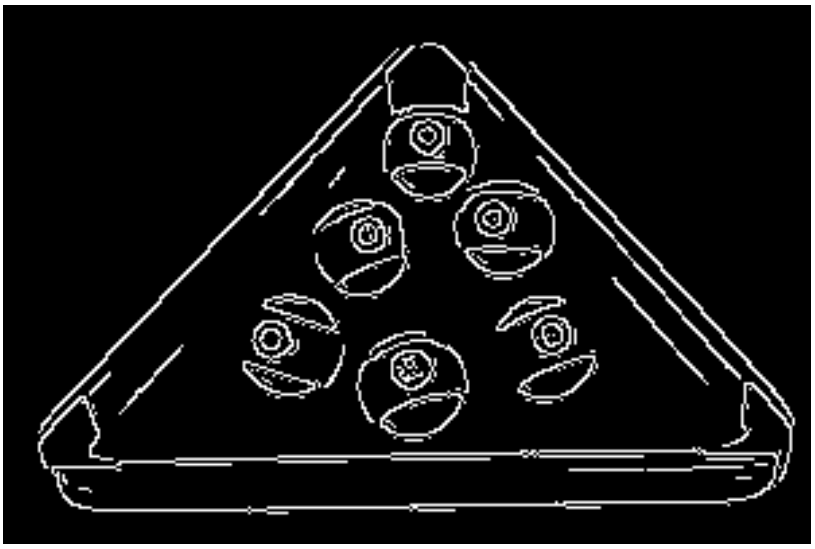
- **Voting:** Pick the points in the accumulator matrix with the maximum value. These are strong points that indicate the existence of a circle with a, b and r parameters. This gives us the Hough space of circles.

- **Finding Circles:** Finally, using the above circles as candidate circles, vote according to the image. The maximum voted circle in the accumulator matrix gives us the circle.

AFTER IMPLEMENTING THE HOUGH TRANSFORM ON INPUT IMAGE TO DETECT CIRCLE :-

INPUT IMAGE:-

CANNY IMAGE:-

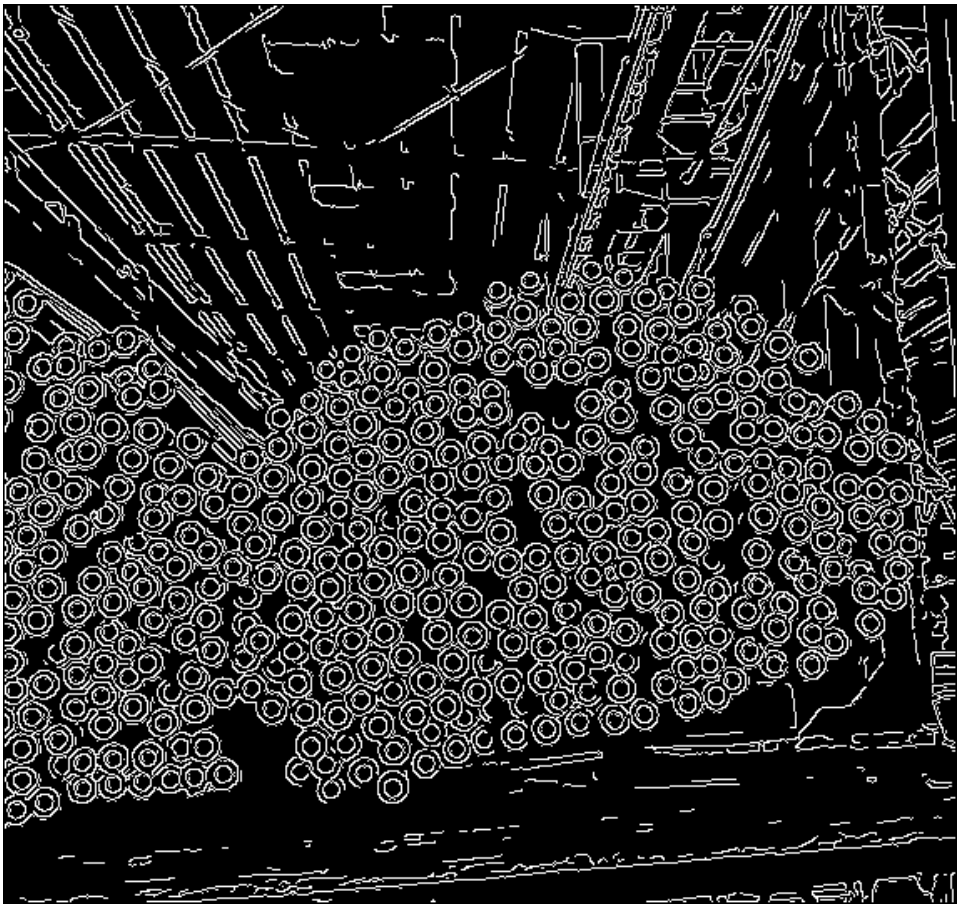


OUTPUT IMAGE AFTER DETECTING CIRCLES:-

ANOTHER EXAMPLE WHERE THERE I GOT A BEAUTIFUL EXPERIENCE BY CHANGING THE Min_Radius and Threshold:-

Input image:-

canny edge detection on input image:-



circle detected on input image:-

Limitations:-

Limitations When working on embedded platforms there will always be a tradeoff between memory, speed, and the accuracy. As discussed in the experiments, there are several limitations of implementing the Circle Hough Transform. The most critical issue is a large amount of memory needed for data storage. When searching a broad range of radii, the need for BRAM greatly increases. If larger images are needed, the size of the BRAM could be a bottleneck and the solution is to use the external memory (SDRAM), which will increase the latency. If images have too many edges, the voting array implementation is limited to an 8-bit counter. Many scenarios need to be considered when using this implementation to a

specific application. However, the design turns out to be highly advantageous where speed is a concern.