Lane Segmentation

Topics to be covered

1. Introduction
2. Pre-requisites
3. Architecture 1.0
4. Challenges Faced.
5. Architecture 2.0
6. Final Results.
7. Limitations.
8. Conclusion.
9. Future Scope.

Problem Statement

Lane segmentation is a common problem in the field of autonomous vehicles.

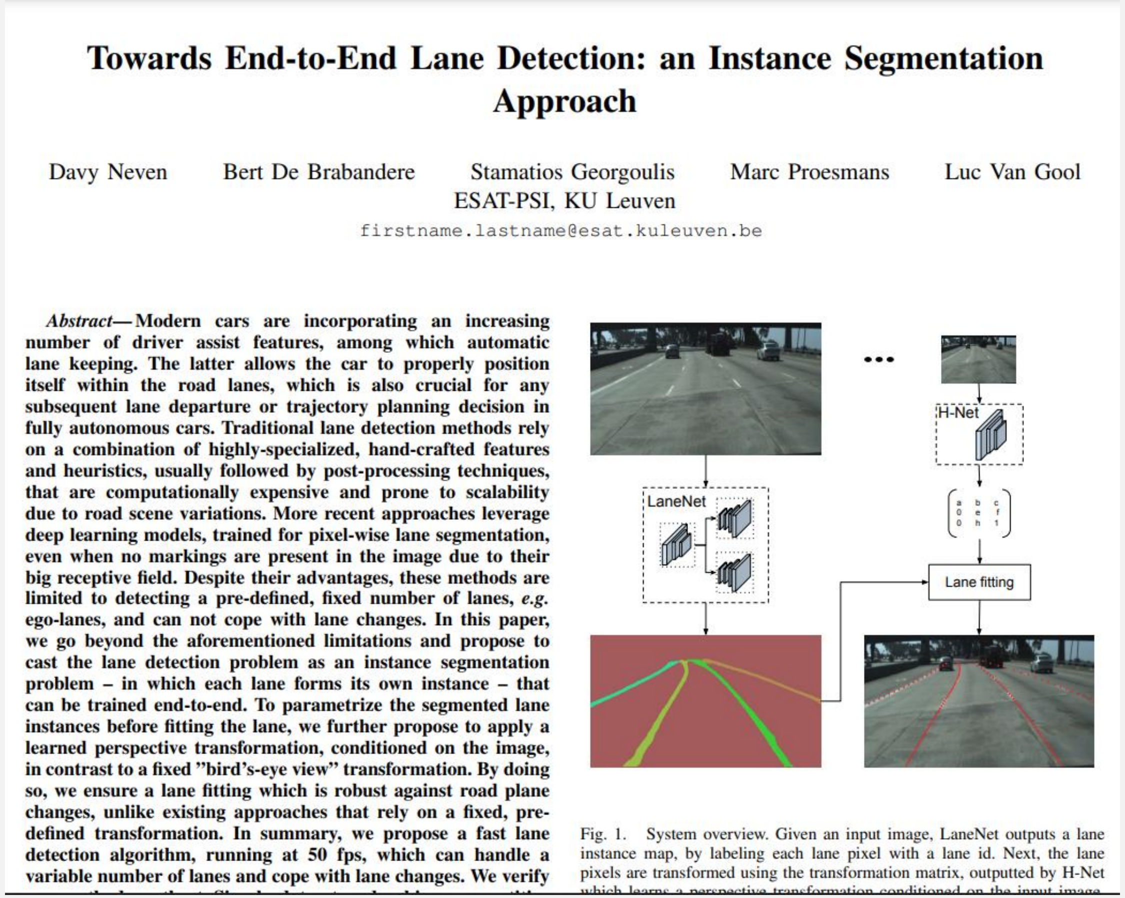
It involves using Computer Vision techniques to identify and distinguish the different lanes on a road.

This is an important task, as it allows the vehicle to understand its surroundings and navigate safely.

Inspiration

List of companies working on Self-Driving cars.

Related Research

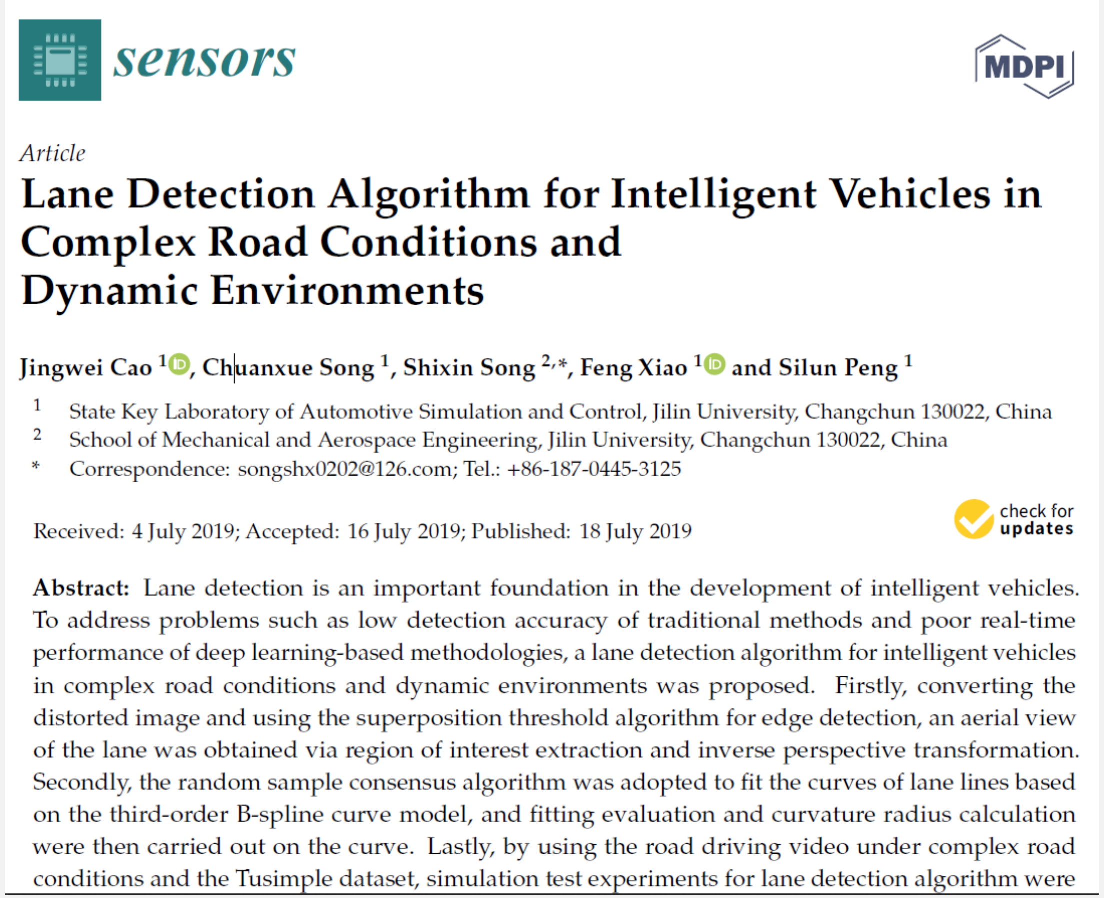


Towards End-to-End Lane Detection: an Instance Segmentation Approach

Davy Neven Bert De Brabandere Stamatios Georgoulis Marc Proesmans Luc Van Gool
ESAT-PSI, KU Leuven
firstname.lastname@esat.kuleuven.be

***Abstract*—Modern cars are incorporating an increasing number of driver assist features, among which automatic lane keeping. The latter allows the car to properly position itself within the road lanes, which is also crucial for any subsequent lane departure or trajectory planning decision in fully autonomous cars. Traditional lane detection methods rely on a combination of highly-specialized, hand-crafted features and heuristics, usually followed by post-processing techniques, that are computationally expensive and prone to scalability due to road scene variations. More recent approaches leverage deep learning models, trained for pixel-wise lane segmentation, even when no markings are present in the image due to their big receptive field. Despite their advantages, these methods are limited to detecting a pre-defined, fixed number of lanes, *e.g.* ego-lanes, and can not cope with lane changes. In this paper, we go beyond the aforementioned limitations and propose to cast the lane detection problem as an instance segmentation problem – in which each lane forms its own instance – that can be trained end-to-end. To parametrize the segmented lane instances before fitting the lane, we further propose to apply a learned perspective transformation, conditioned on the image, in contrast to a fixed "bird's-eye view" transformation. By doing so, we ensure a lane fitting which is robust against road plane changes, unlike existing approaches that rely on a fixed, pre-defined transformation. In summary, we propose a fast lane detection algorithm, running at 50 fps, which can handle a variable number of lanes and cope with lane changes. We verify**

Fig. 1. System overview. Given an input image, LaneNet outputs a lane instance map, by labeling each lane pixel with a lane id. Next, the lane pixels are transformed using the transformation matrix, outputted by H-Net



sensors
MDPI

Article

Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments

Jingwei Cao [1], Chuanxue Song [1], Shixin Song [2,*], Feng Xiao [1] and Silun Peng [1]

1 State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China
2 School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130022, China
* Correspondence: songshx0202@126.com; Tel.: +86-187-0445-3125

Received: 4 July 2019; Accepted: 16 July 2019; Published: 18 July 2019
check for updates

Abstract: Lane detection is an important foundation in the development of intelligent vehicles. To address problems such as low detection accuracy of traditional methods and poor real-time performance of deep learning-based methodologies, a lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments was proposed. Firstly, converting the distorted image and using the superposition threshold algorithm for edge detection, an aerial view of the lane was obtained via region of interest extraction and inverse perspective transformation. Secondly, the random sample consensus algorithm was adopted to fit the curves of lane lines based on the third-order B-spline curve model, and fitting evaluation and curvature radius calculation were then carried out on the curve. Lastly, by using the road driving video under complex road conditions and the Tusimple dataset, simulation test experiments for lane detection algorithm were

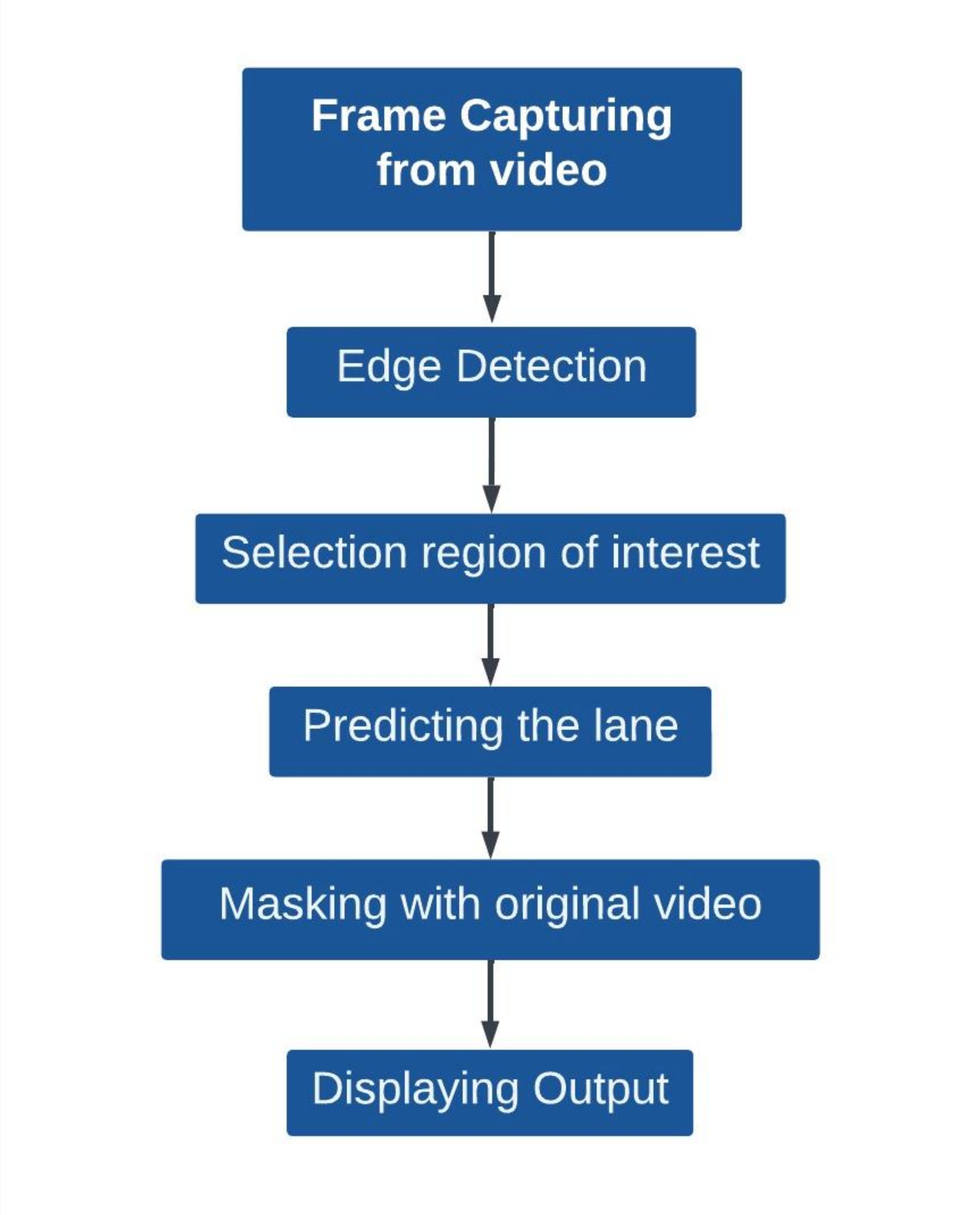
Proposed Solution

This project is to develop an effective Lane Segmentation algorithm using OpenCV that can be used in autonomous vehicles.

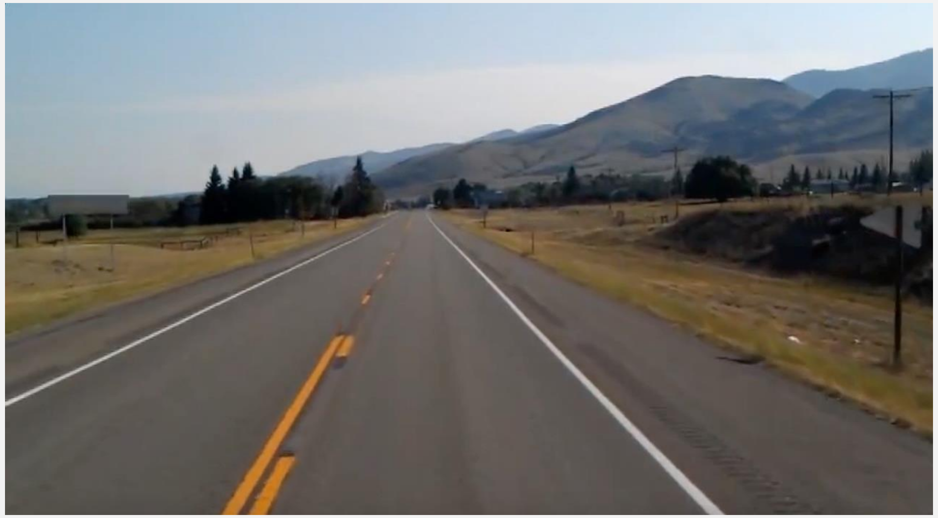
Pre-requisites

- Camera Angle to maintain the region of interest.
- Distinct visibility of lane.
- Sturdy Camera Mount.

Architecture 1



Procedure for edge detection



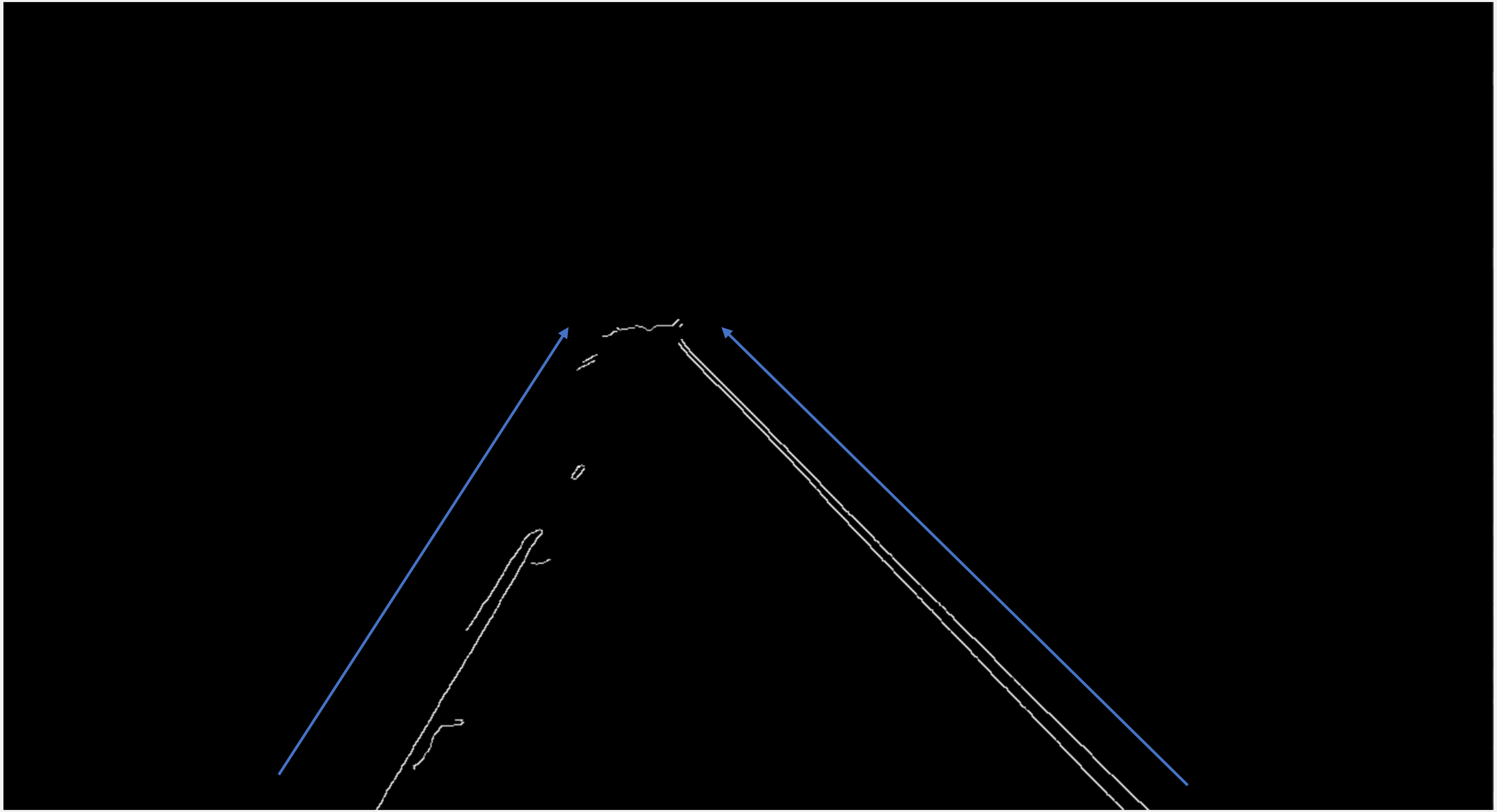
a. Normal Image

b. Convert to Grayscale

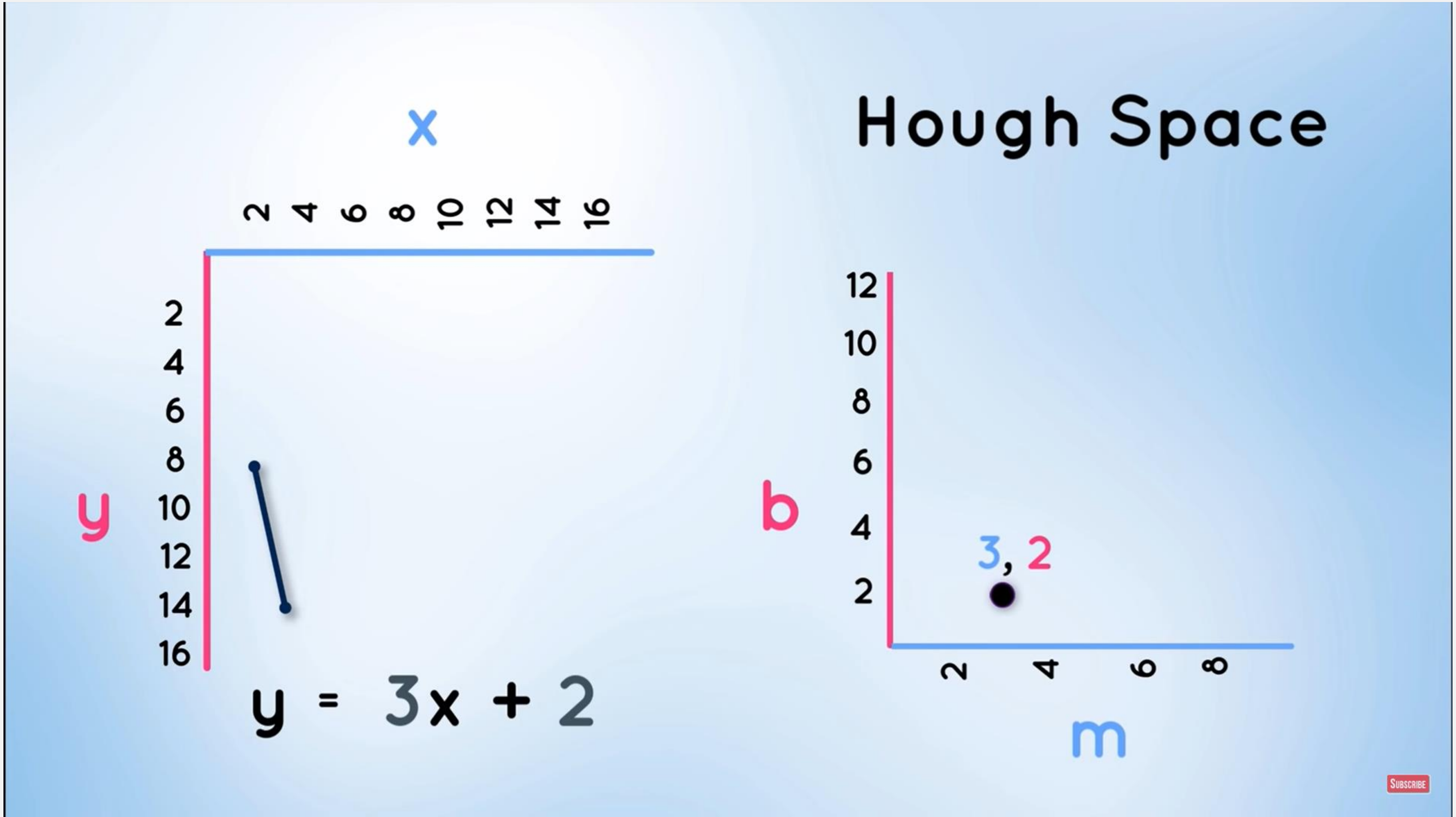
c. Applying Gaussian Blur

d. Applying Canny Edge Detection

Region of Interest



Hough Space



Hough Space
x
2 4 6 8 10 12 14 16
y
2 4 6 8 10 12 14 16
y = 3x + 2
b
12 10 8 6 4 2
3, 2
2 4 6 8
m
SUBSCRIBE

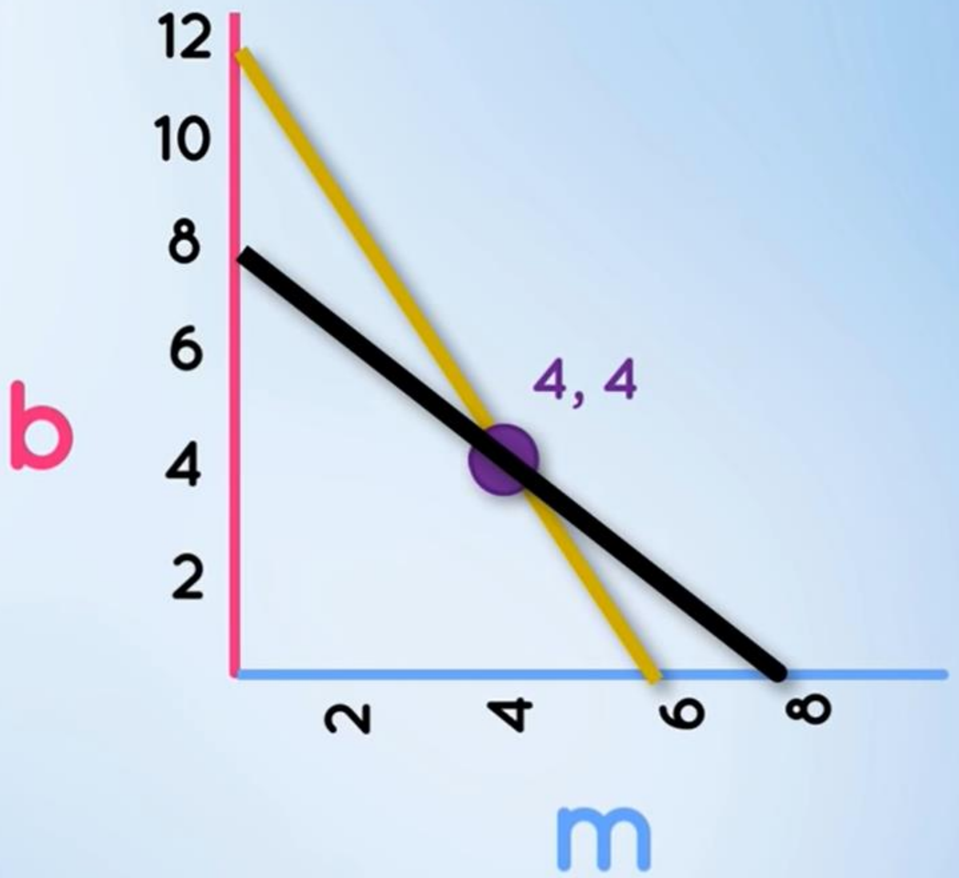

x
2 4 6 8 10 12 14 16
y
2 4 6 8 10 12 14 16

Hough Space

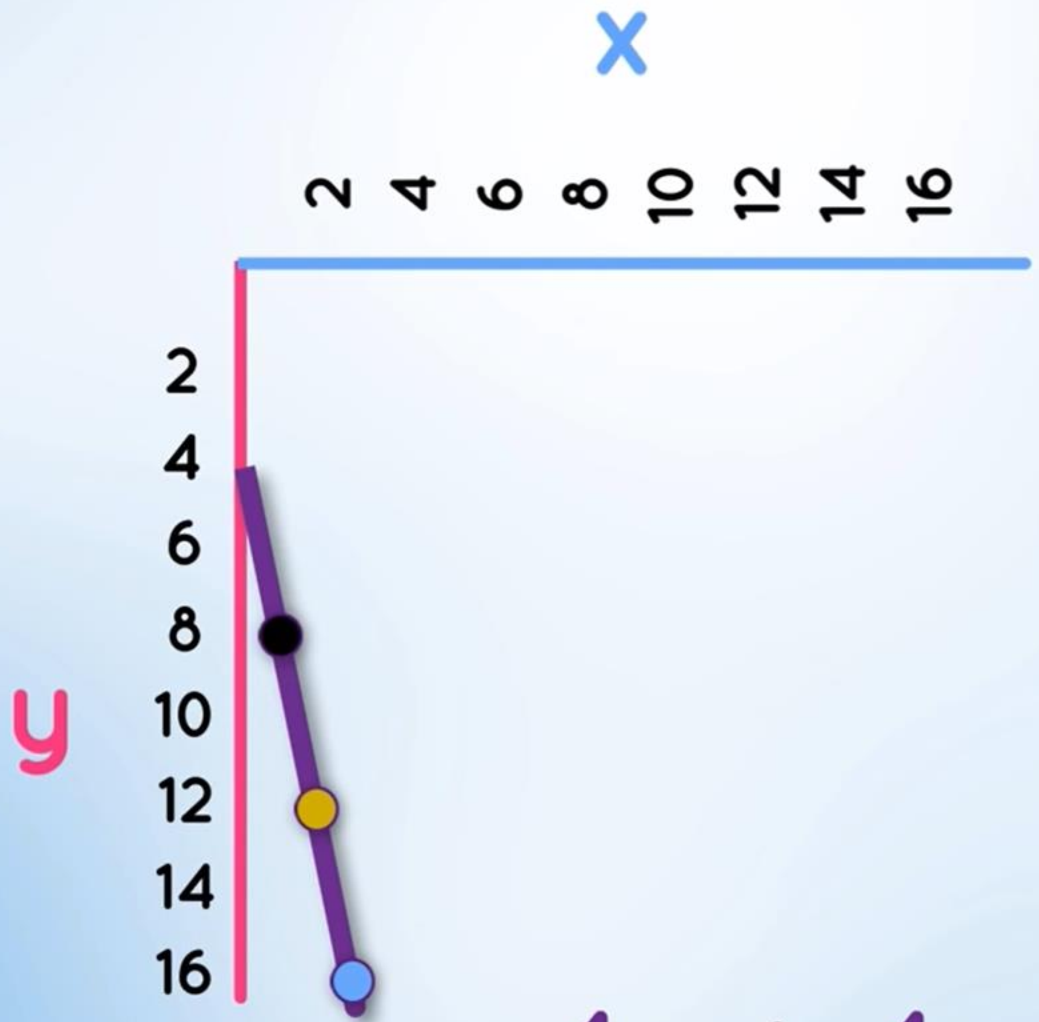

b
12 10 8 6 4 2
2, 8
3, 6
4, 4
5, 2
6, 0
2 4 6 8
m

Hough Space

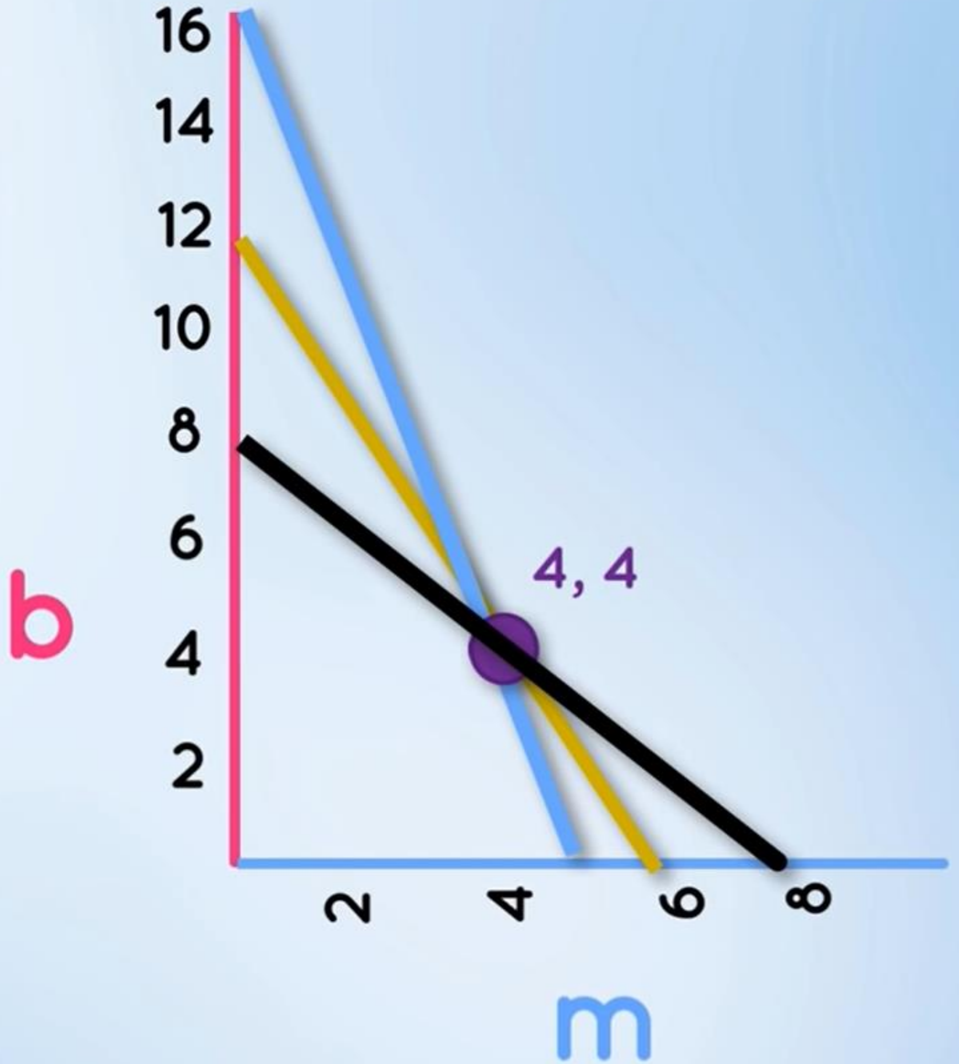
$$y = 4x + 4$$



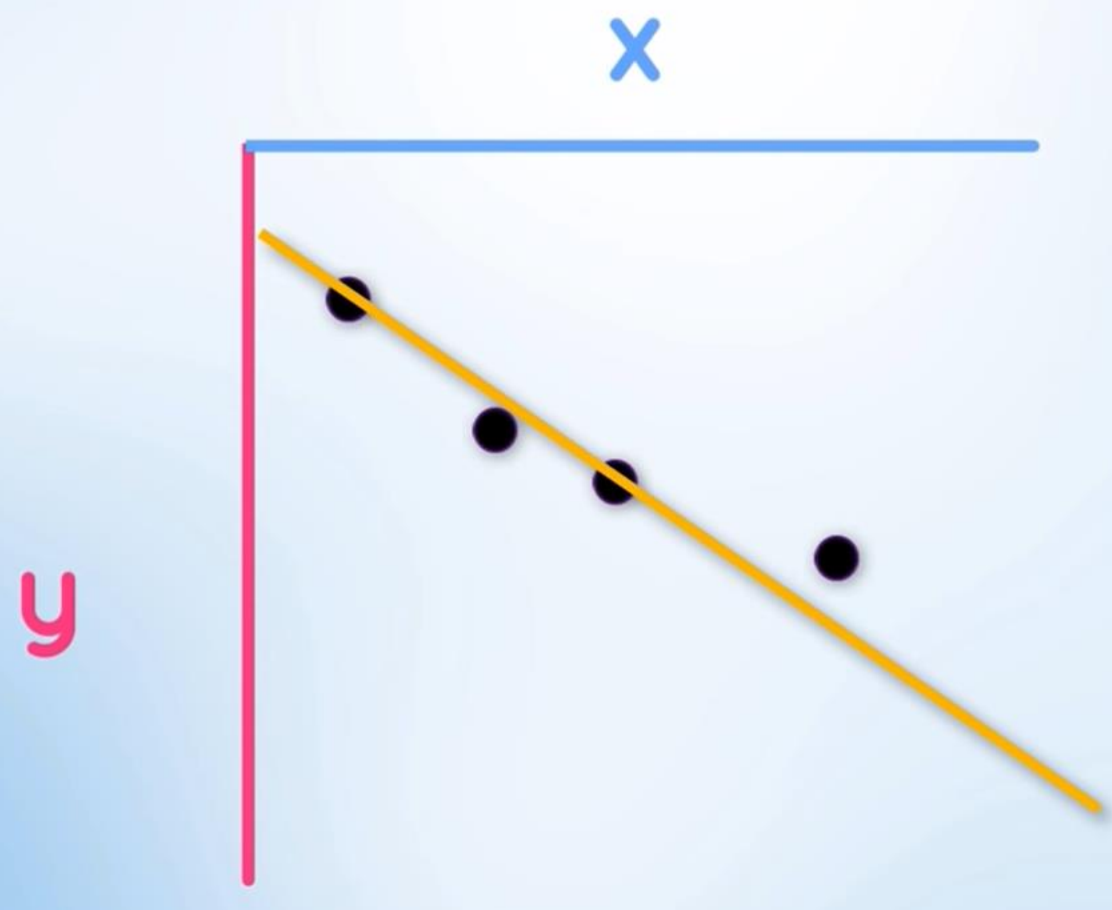
Hough Space



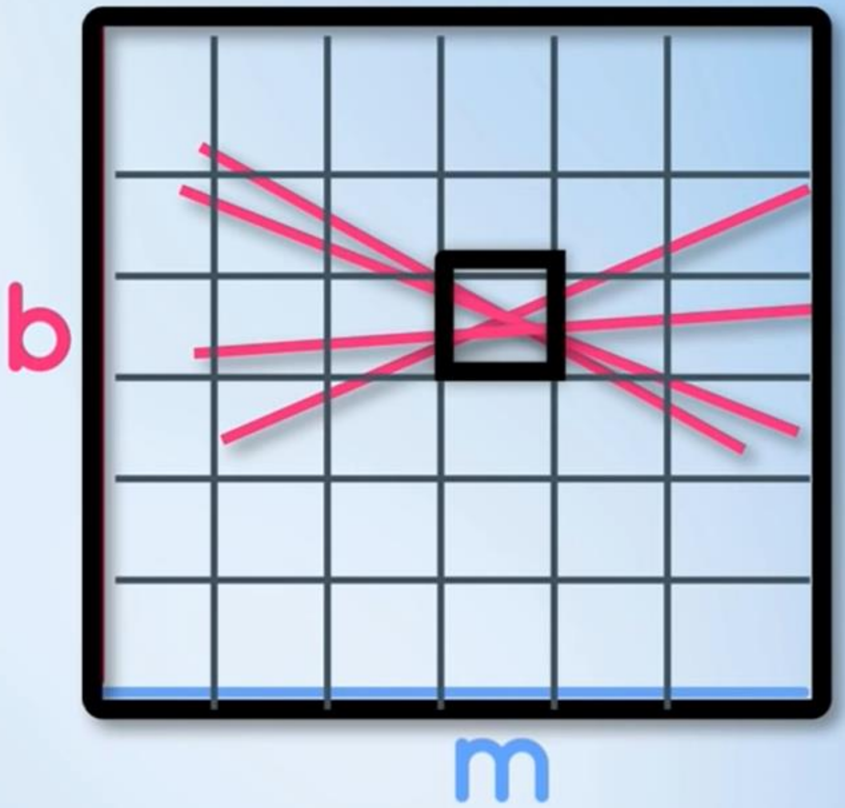
$$y = 4x + 4$$

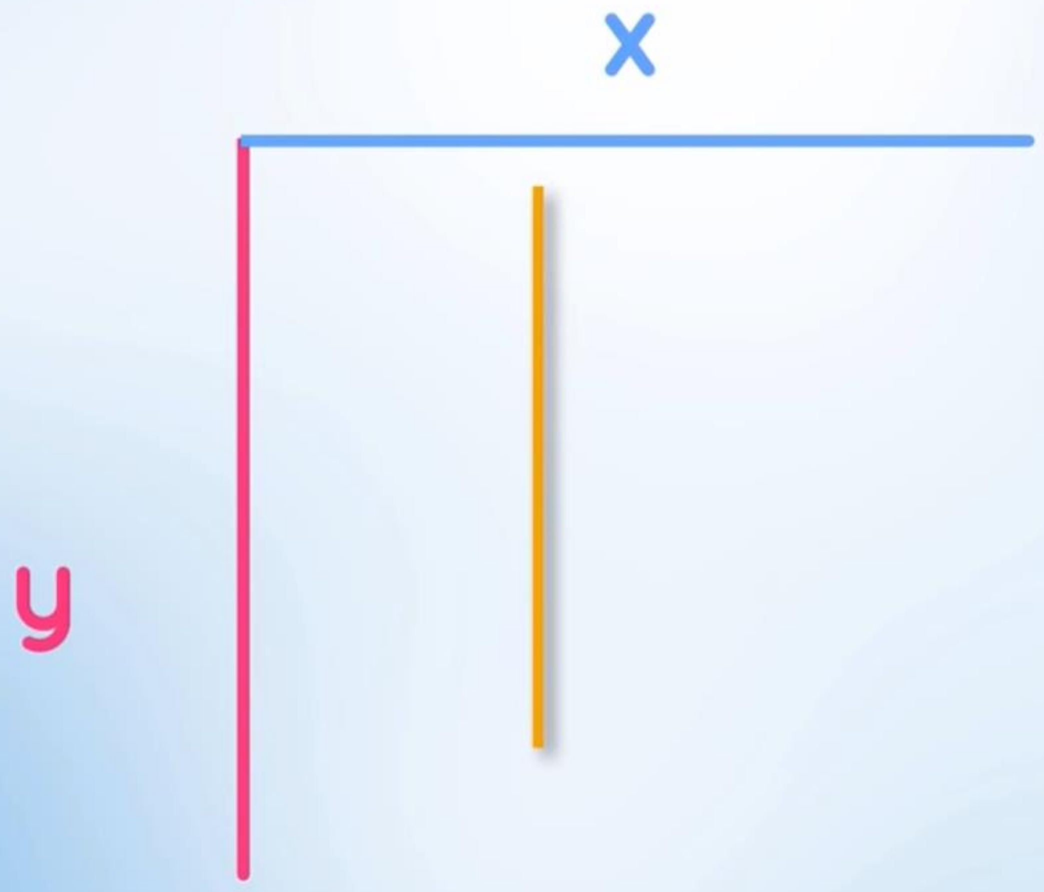


Hough Space

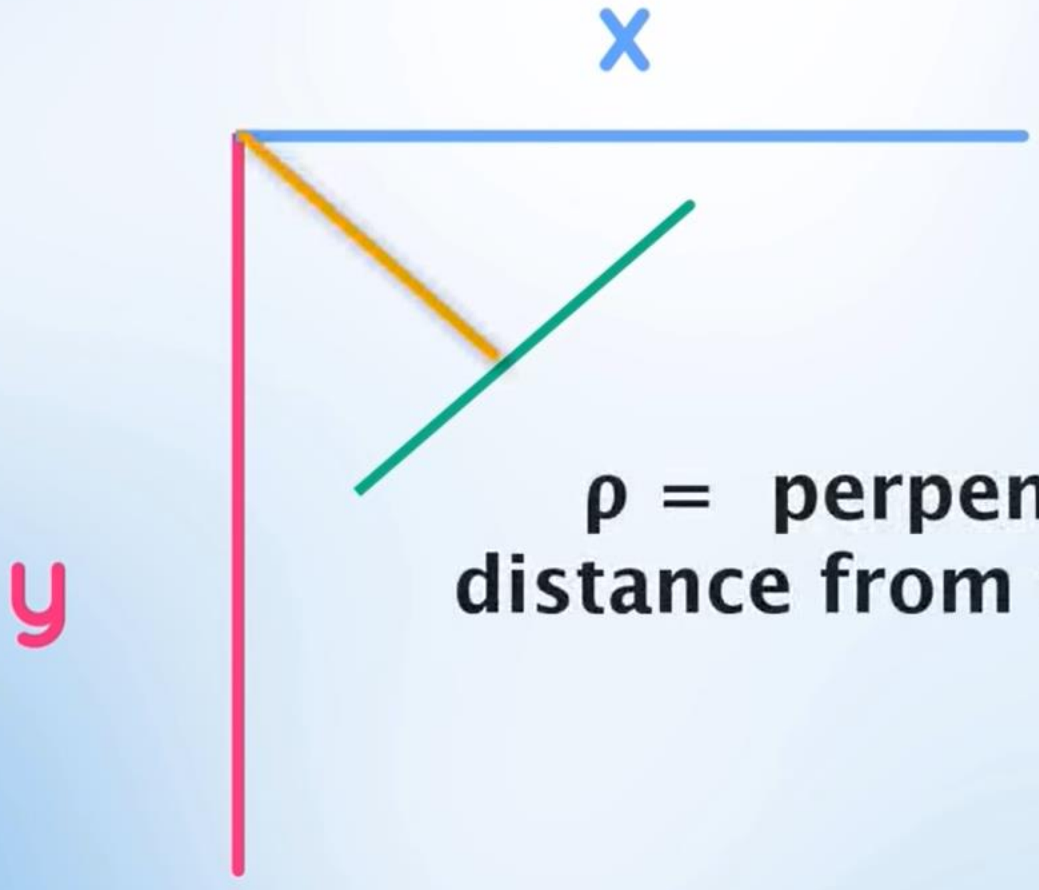


$$y = mx + b$$



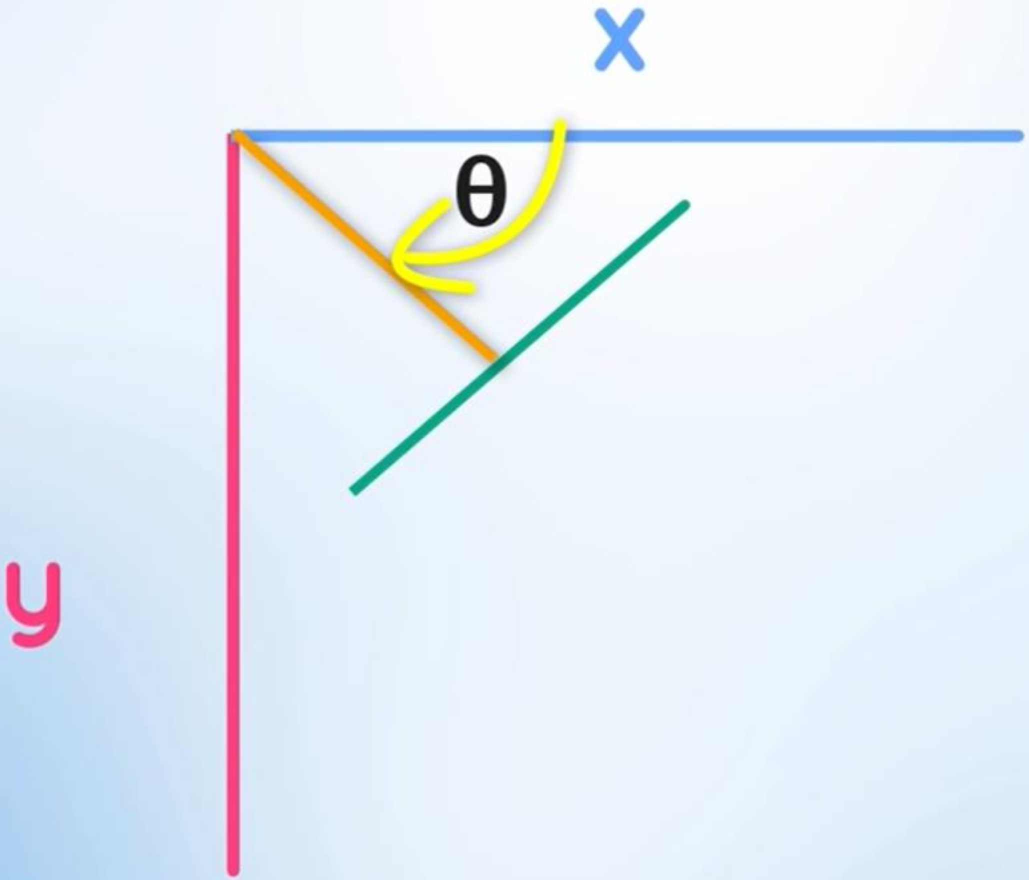


$y = mx + b$ where m is infinite



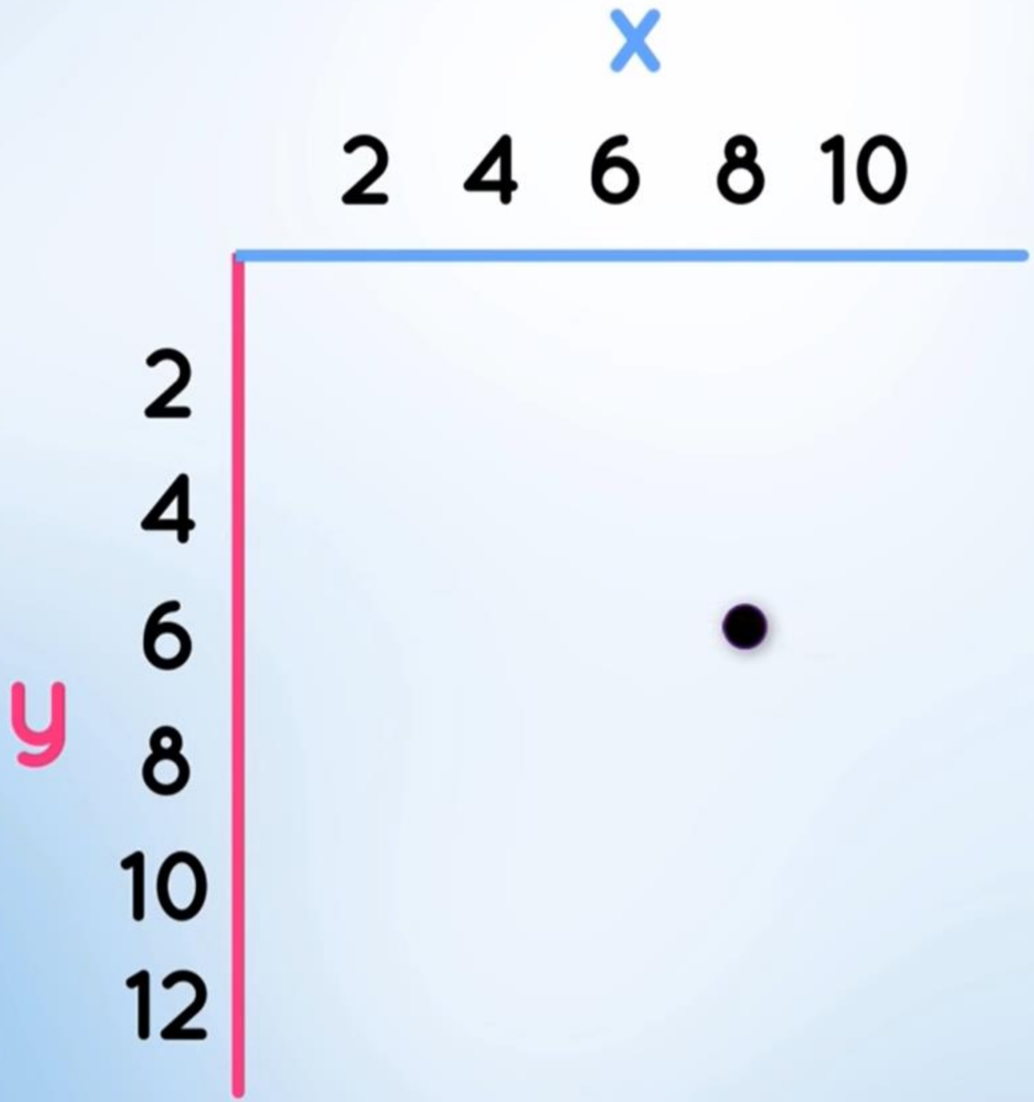
ρ = perpendicular distance from the origin

$$\rho = x\cos\theta + y\sin\theta$$



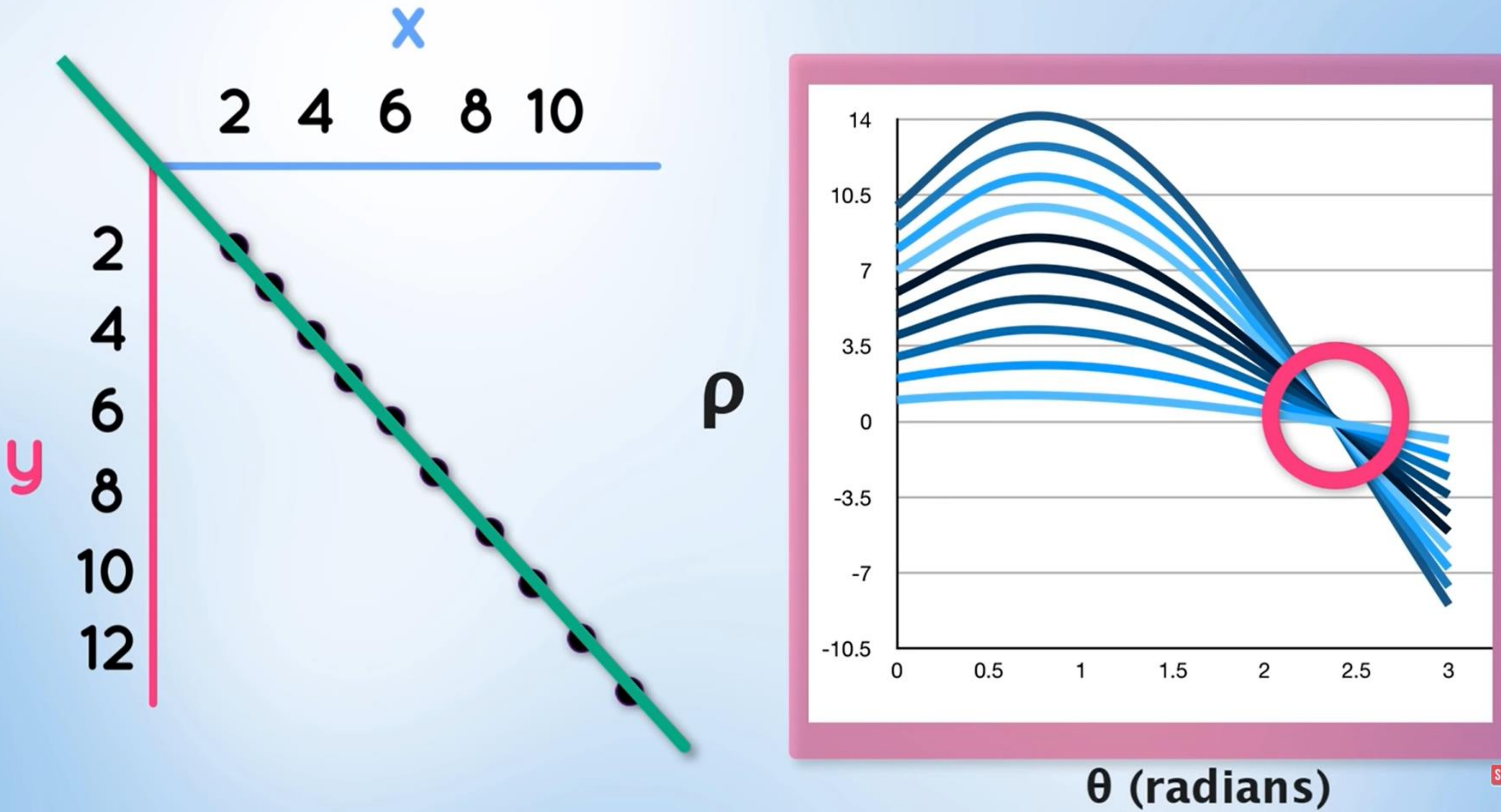
$$\rho = x\cos\theta + y\sin\theta$$

Hough Space

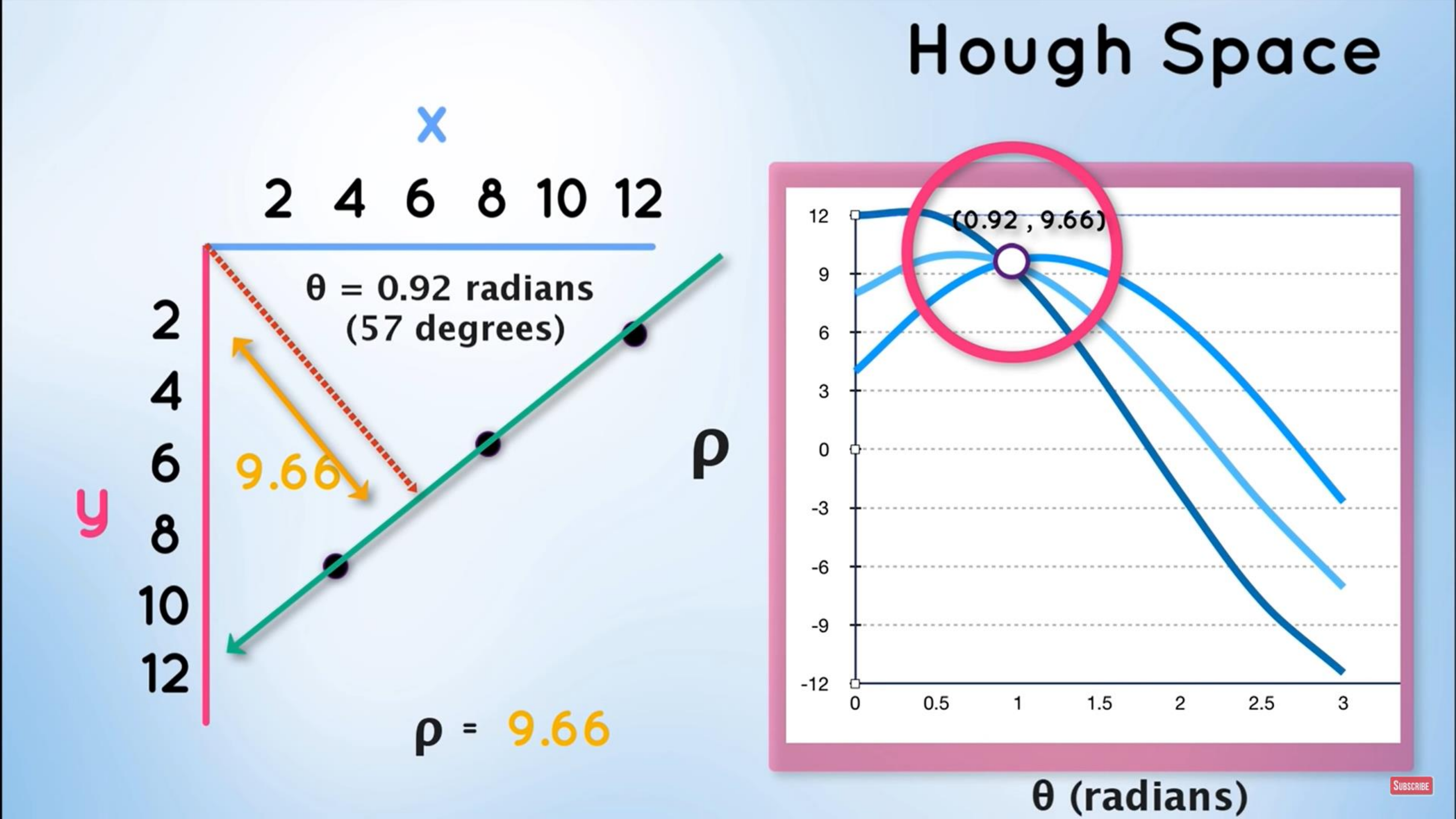

x
2 4 6 8 10
y
2
4
6
8
10
12


ρ
10
7.5
5
2.5
0
-2.5
-5
-7.5
-10
0 0.5 1 1.5 2 2.5 3
θ (radians)

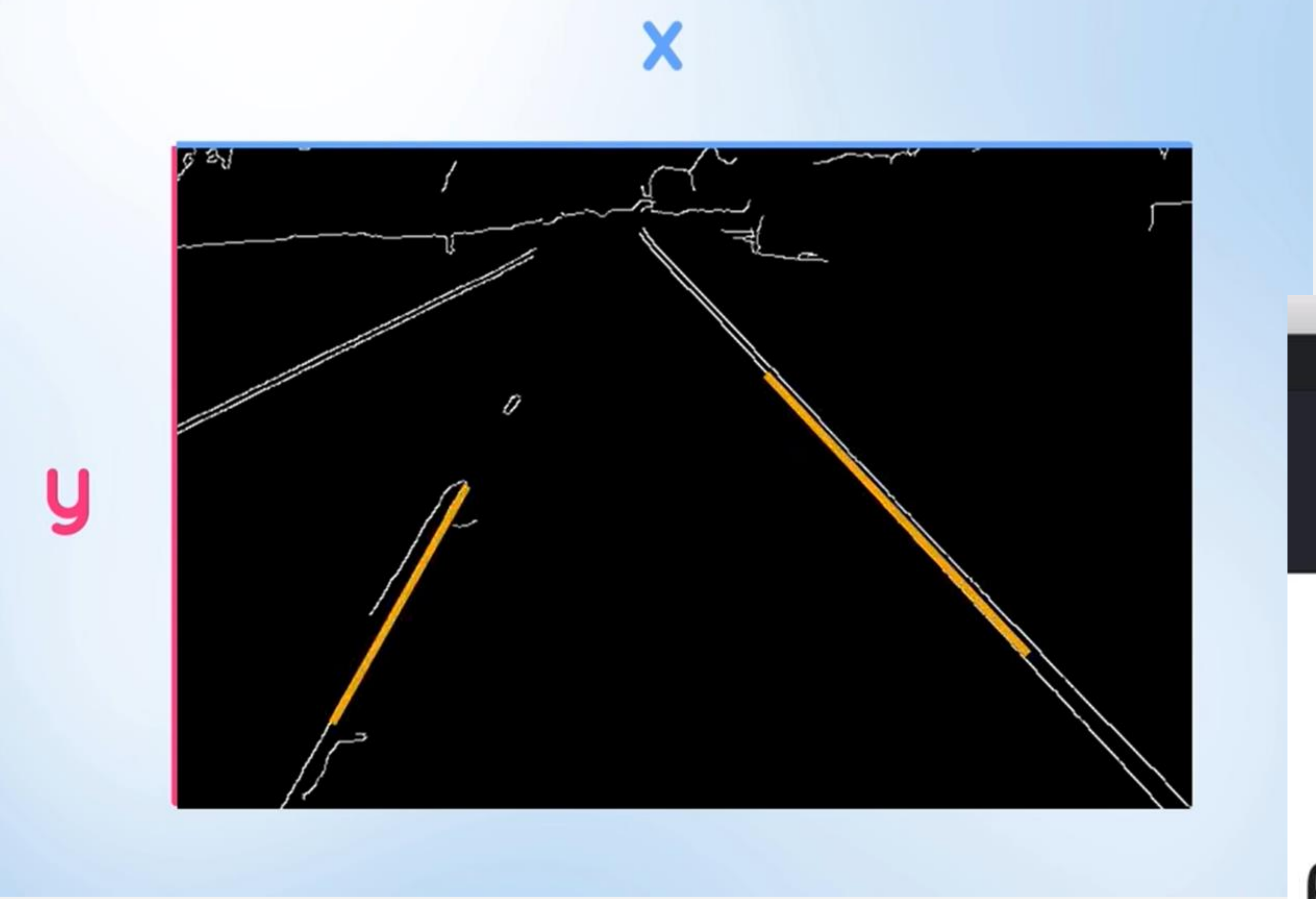
Hough Space

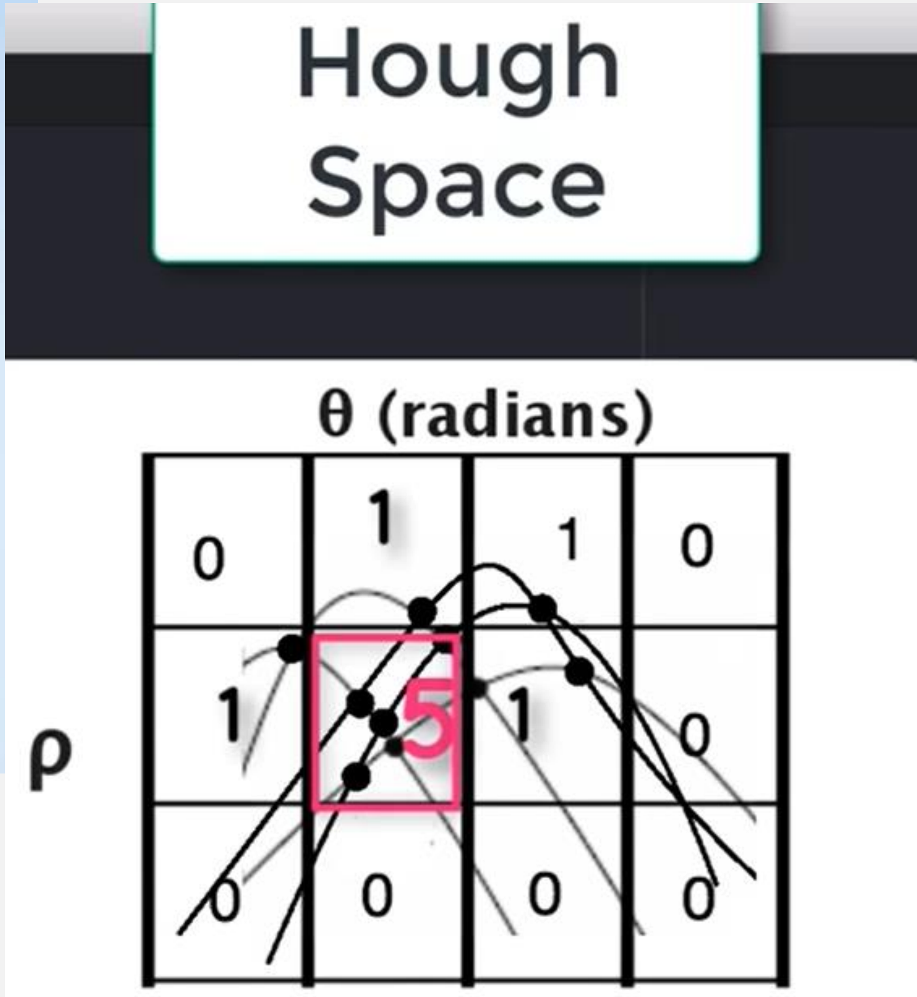


x
2 4 6 8 10
y
2
4
6
8
10
12
ρ
14
10.5
7
3.5
0
-3.5
-7
-10.5
0 0.5 1 1.5 2 2.5 3
θ (radians)

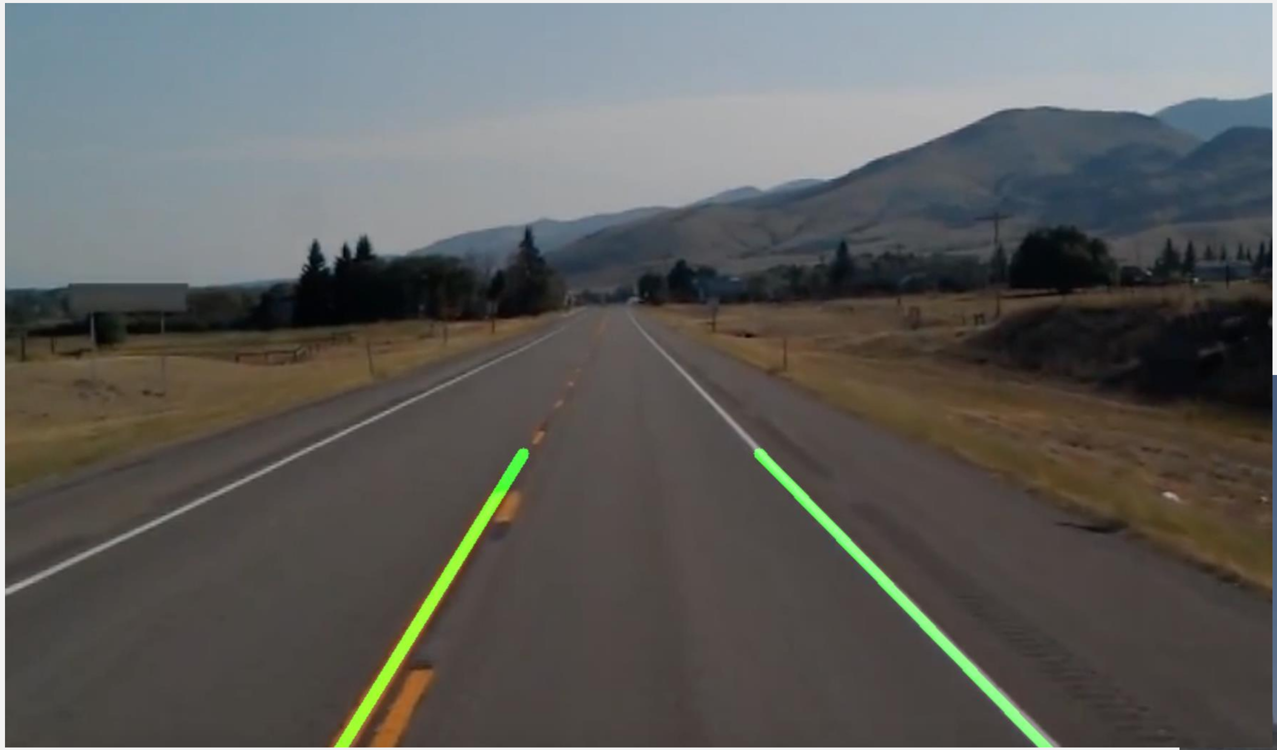


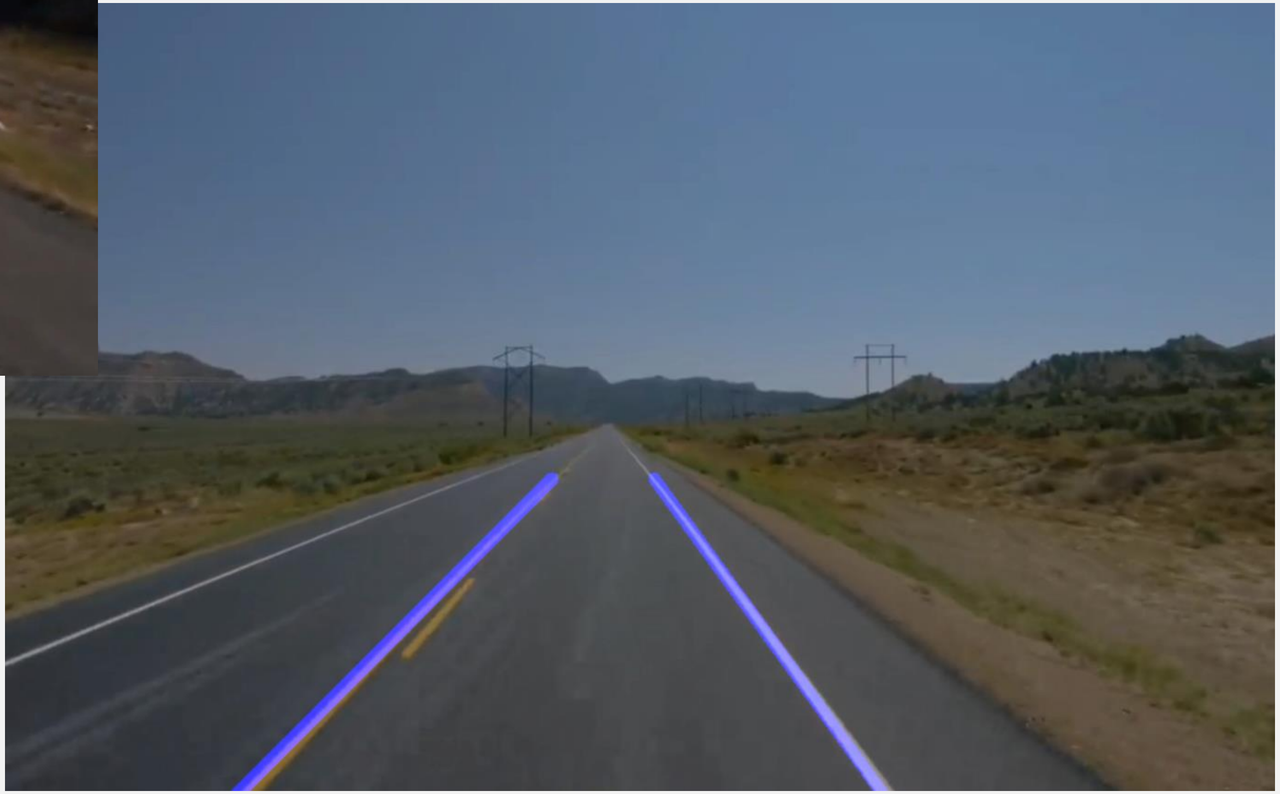
Hough Space
x
2 4 6 8 10 12
y
2
4
6
8
10
12
θ = 0.92 radians
(57 degrees)
9.66
ρ = 9.66
ρ
(0.92 , 9.66)
12
9
6
3
0
-3
-6
-9
-12
0
0.5
1
1.5
2
2.5
3
θ (radians)


x
y

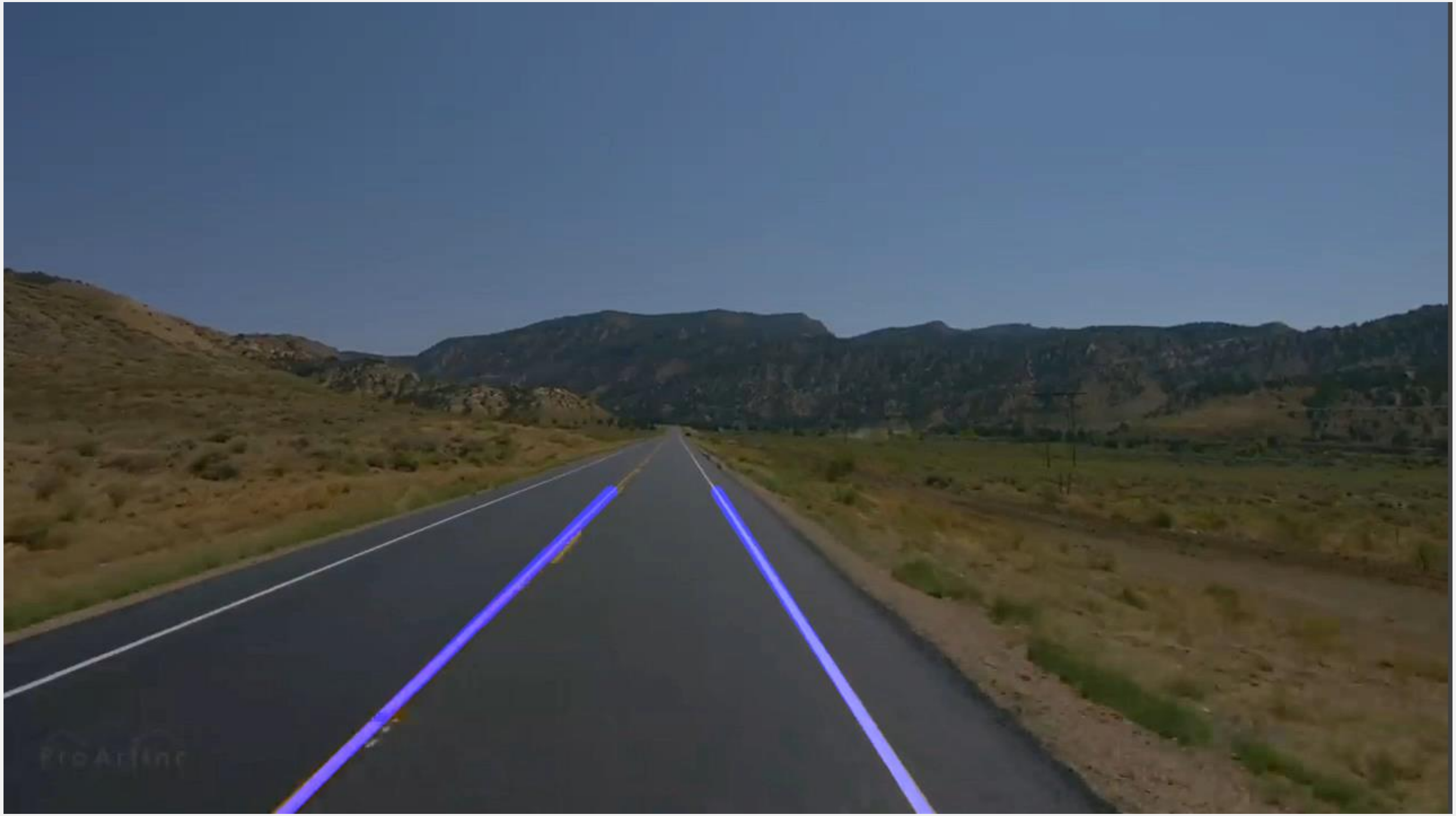

Hough
Space
θ (radians)
ρ
0
1
1
0
1
5
1
0
0
0
0
0

Lane prediction using Hough Transform



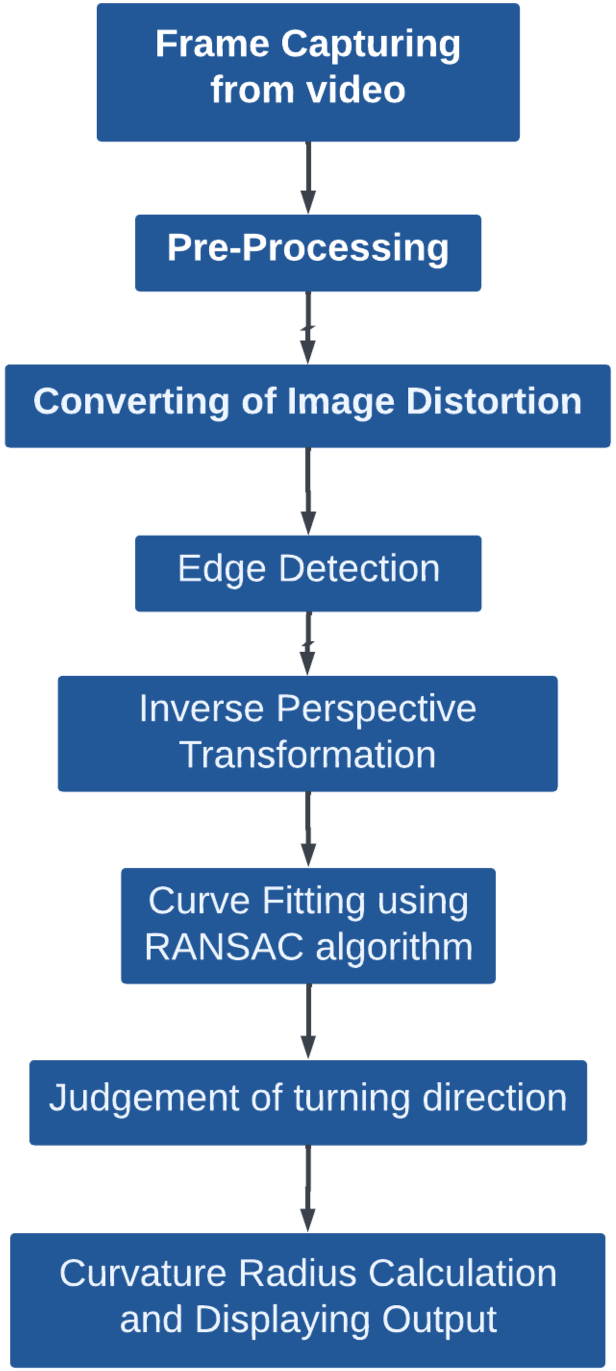


Lane segmentation on video

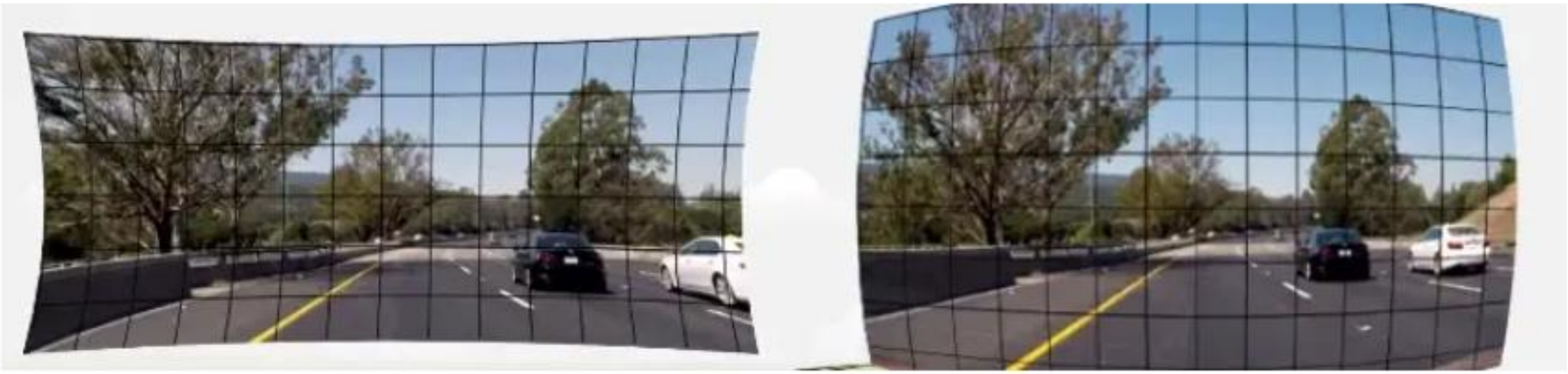


Challenges Occurred

Architecture 2.0

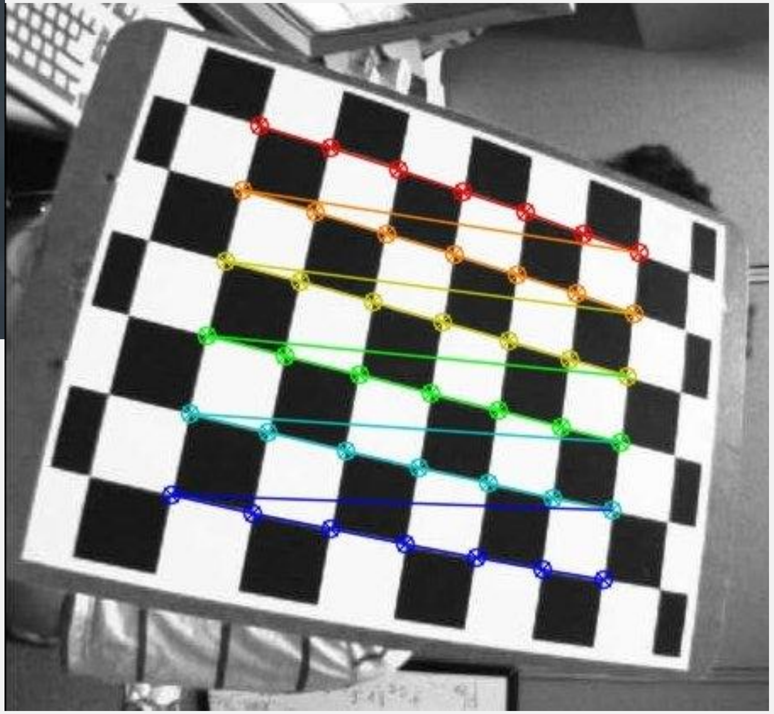


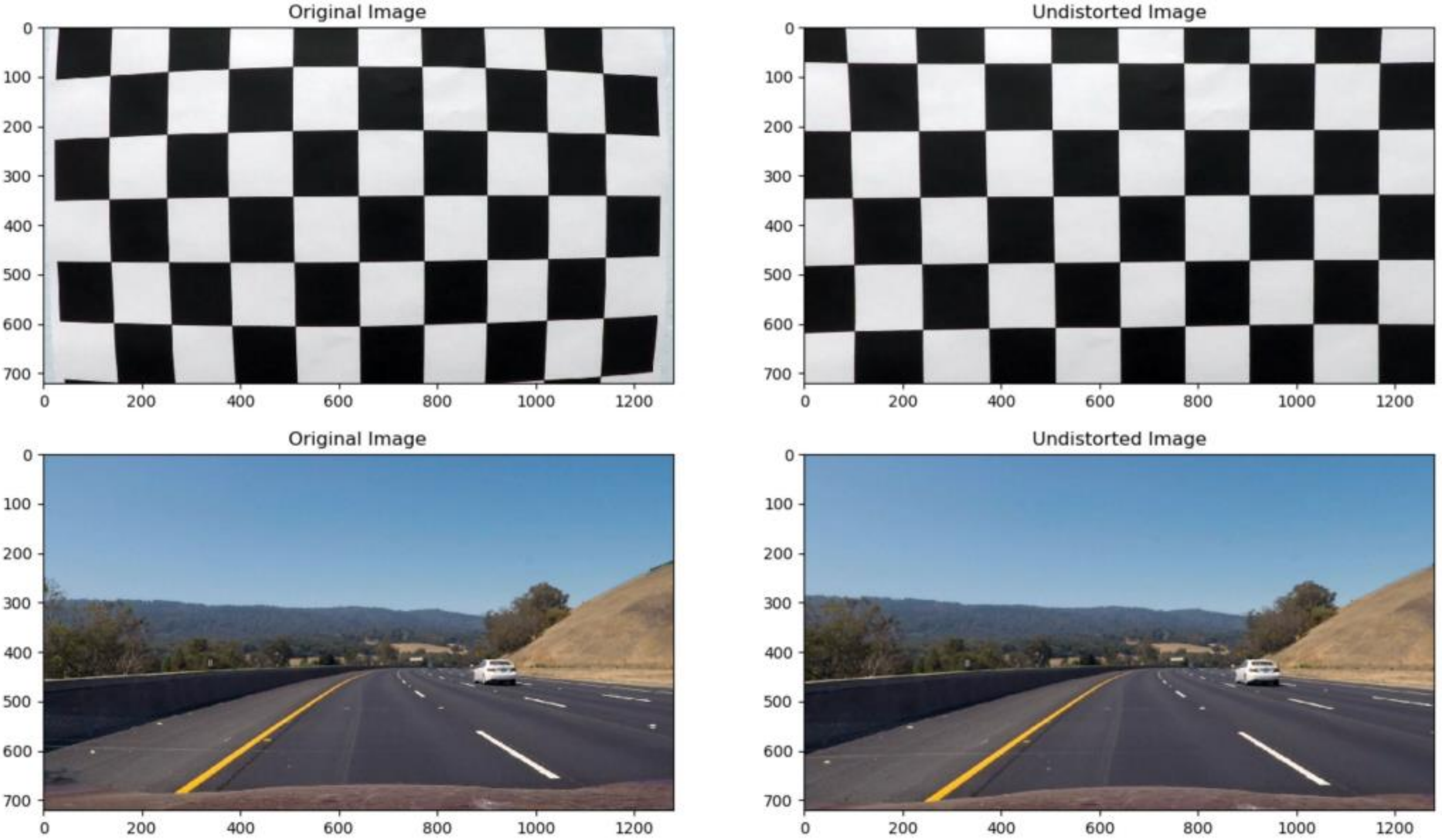
Camera Calibration



Radially Distorted by a camera

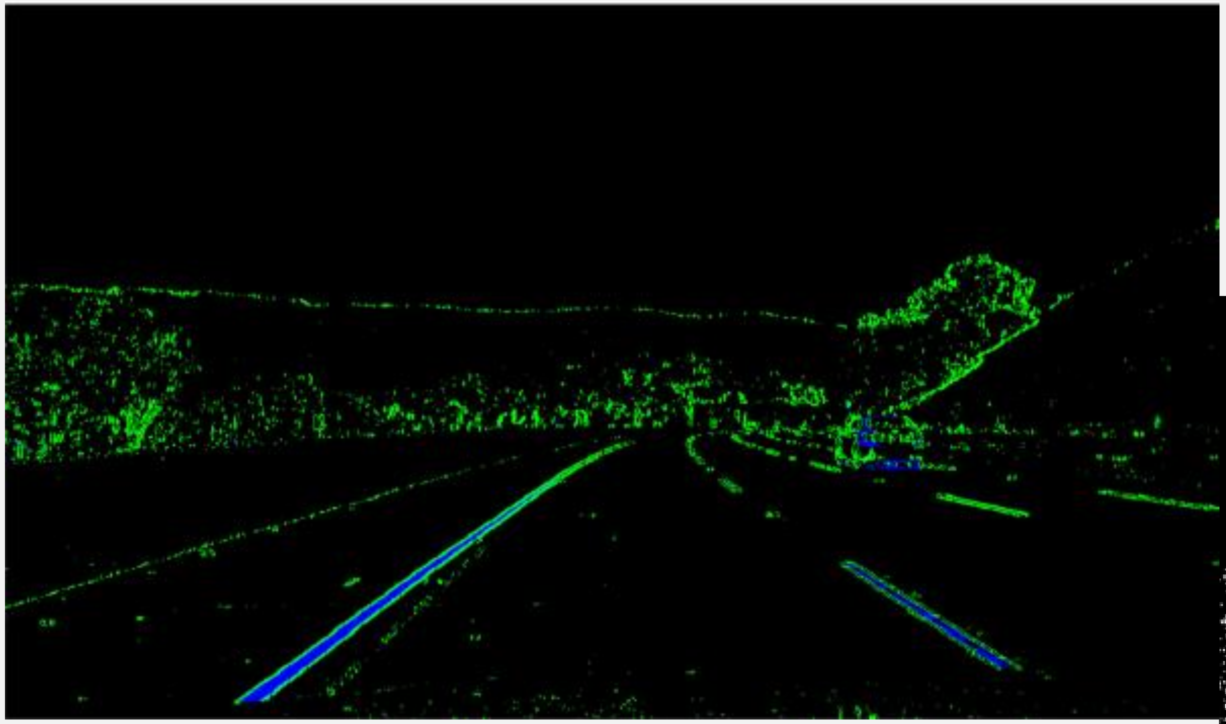
Calibrating the Camera using Chess Board.



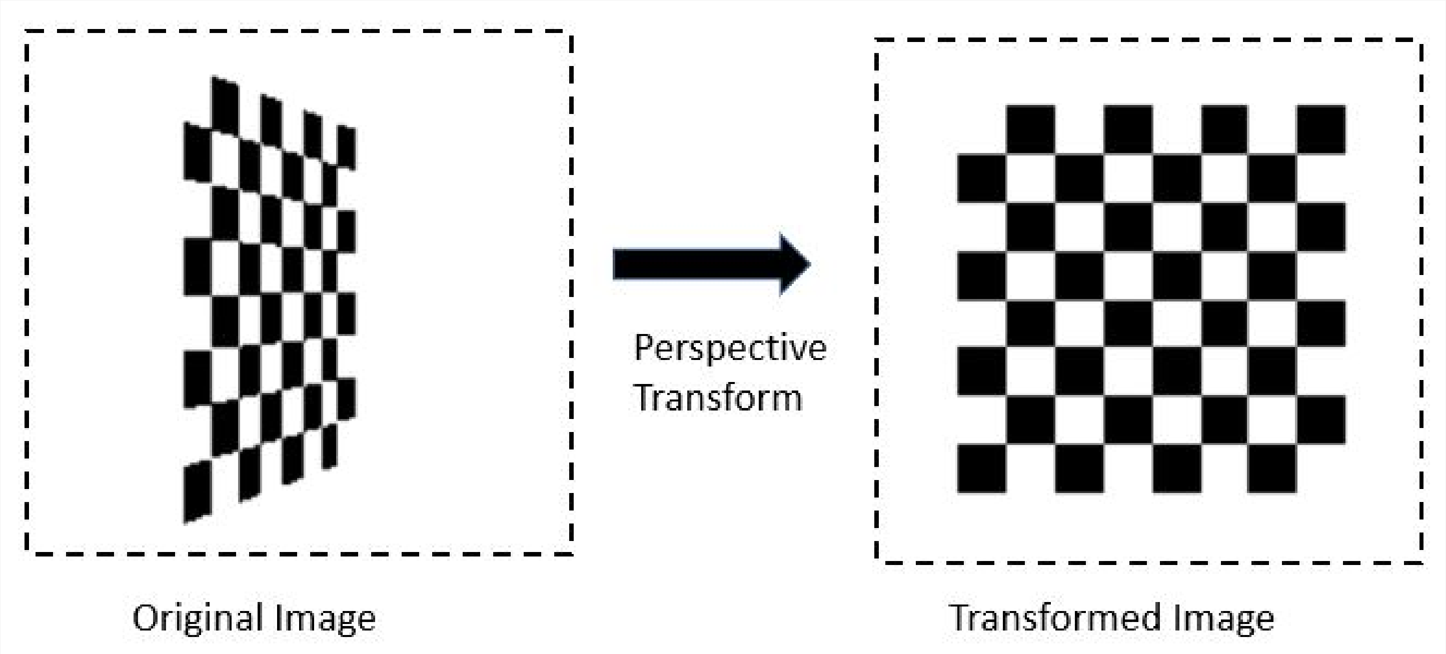

Original Image
Undistorted Image
Original Image
Undistorted Image
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

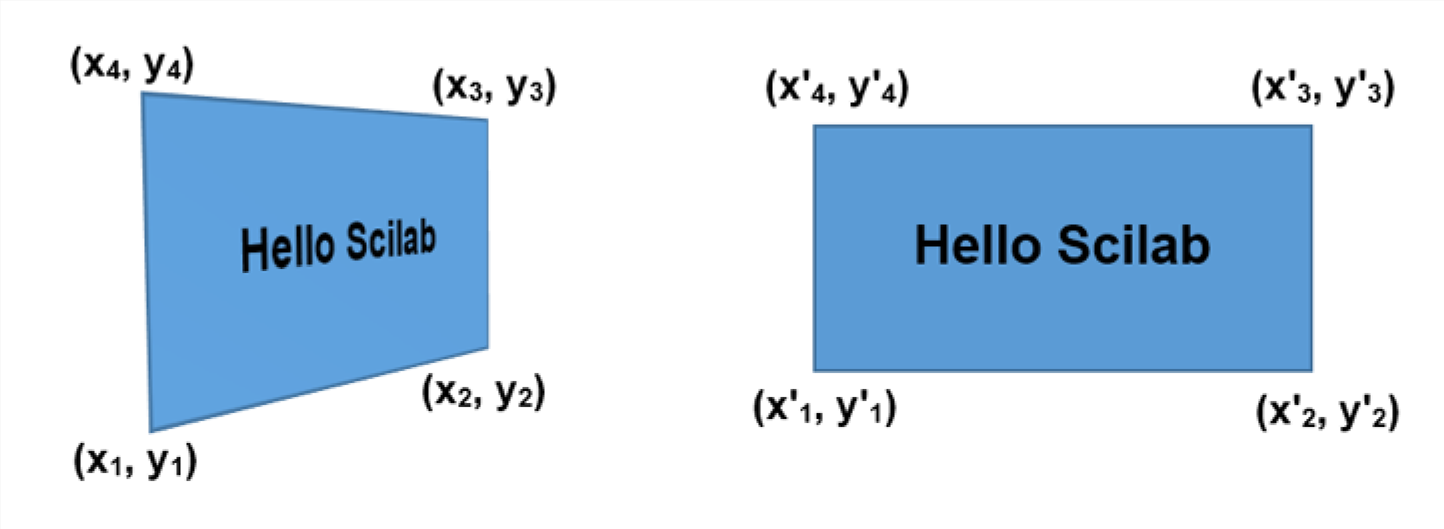
Thresholding

- Compared with the RGB color space, the HSL color space can better reflect the visual field perception characteristics of the naked eye



Inverse Perspective Transform



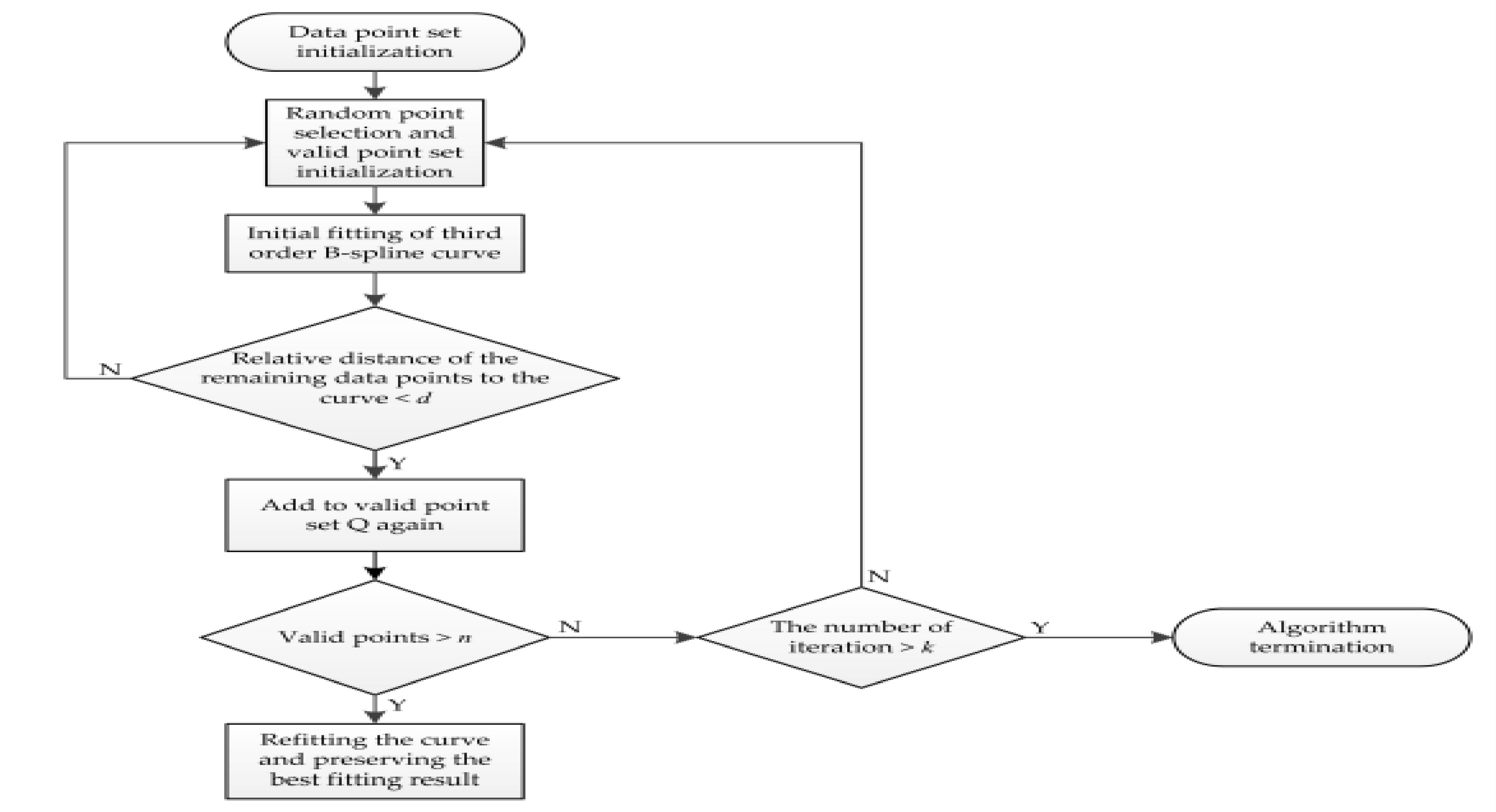


Region of Interest

Inverse Perspective Transformation

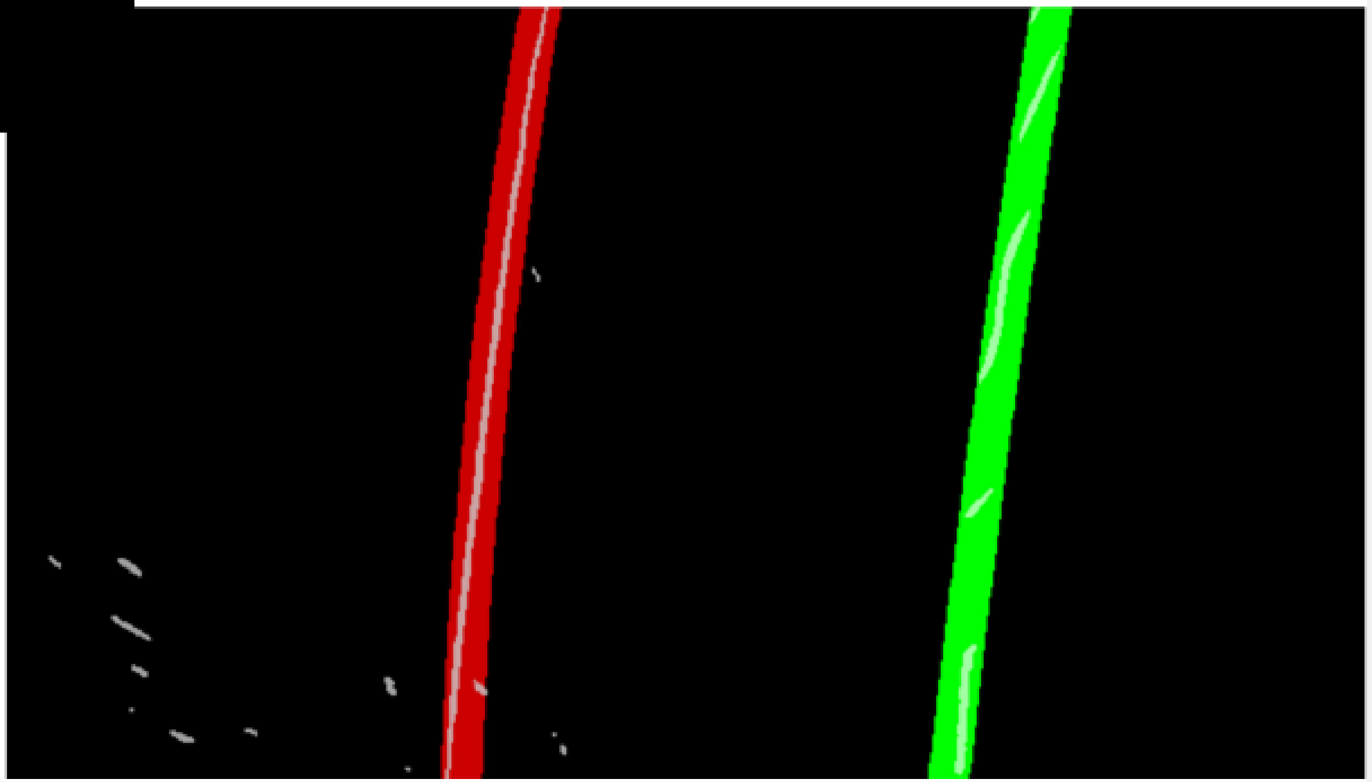
Using B- Spline to predict the lane line

- The RANSAC algorithm based on the third-order B-spline curve model was used to fit the lane lines

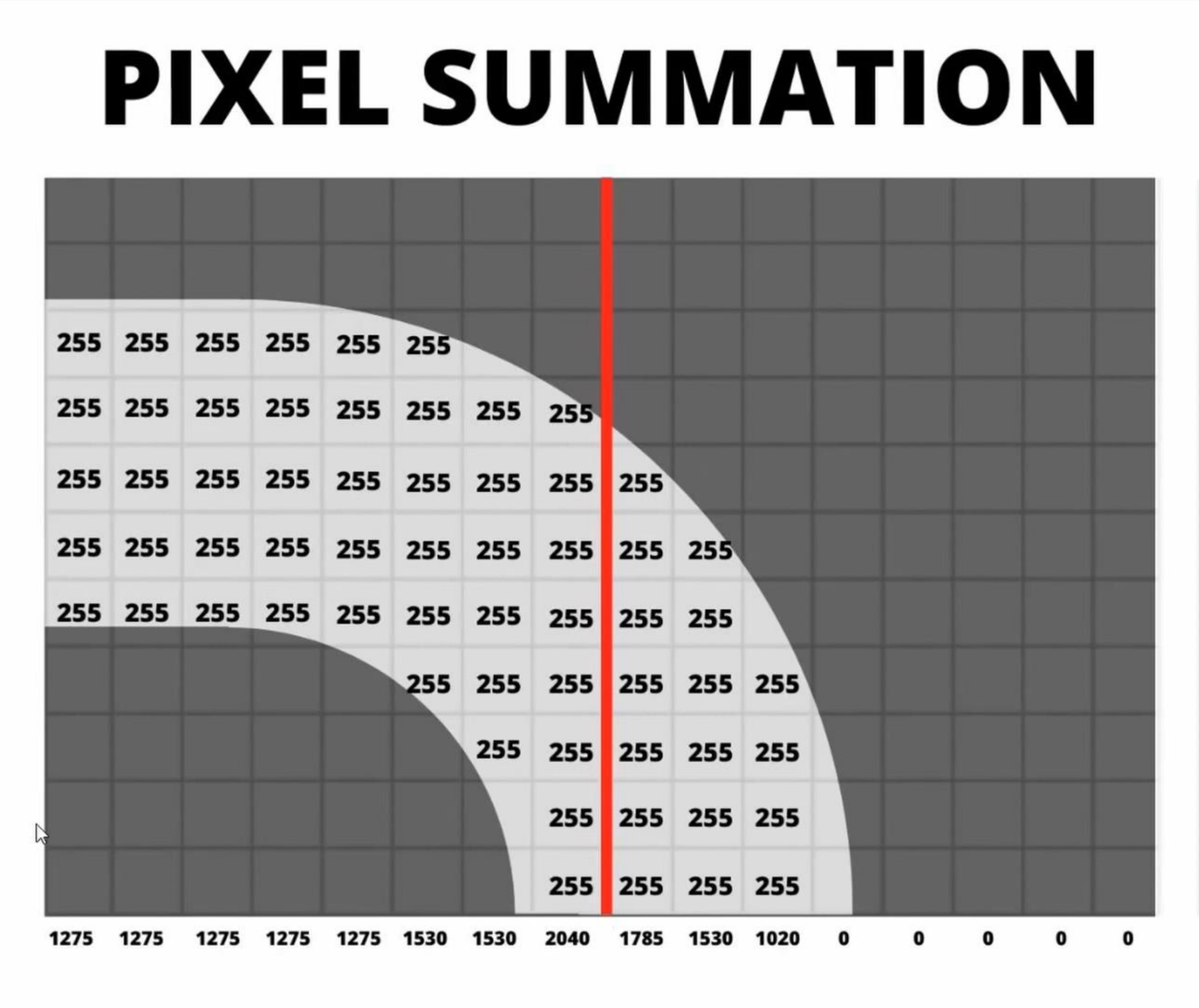


Inverse Perspective Transformation

Lane Detecting



Predicting the Turn



Displaying the Final Results

Limitations

- Needs Proper Lighting Conditions for functioning.
- Camera Calibration is different for every lens because each and every lens produce different levels of distortion.

Conclusion

Lane detection is an essential component of self-driving car technology.

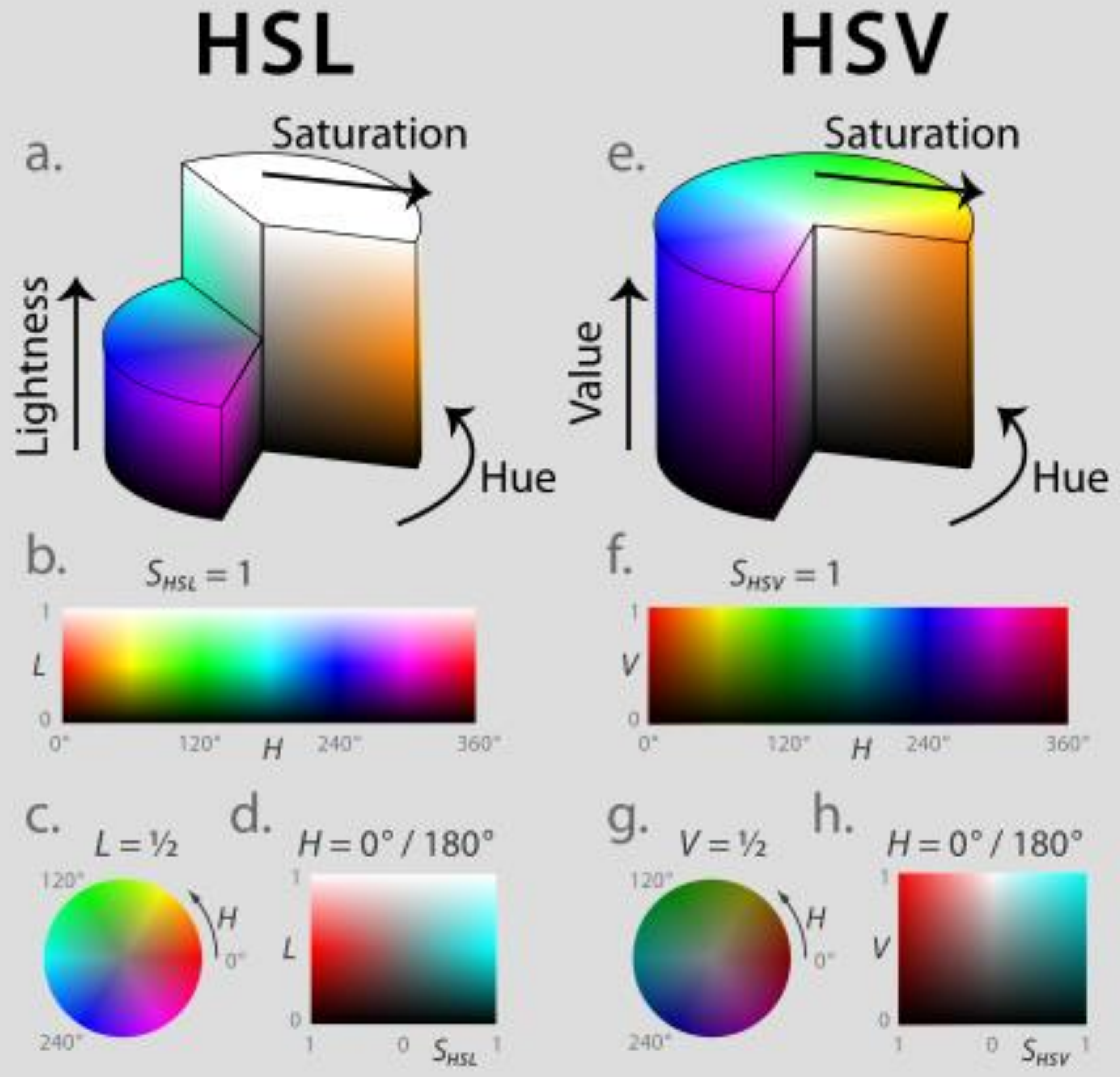
In this project, we have implemented a lane detection system using computer vision techniques.

Overall, this project has demonstrated the feasibility of using lane detection for self-driving cars, and has laid the foundation for further development and refinement of this technology.

Future Scope

- Path Planning.
- Applying Transformer Networks Architecture for computer vision.
- Data Collection of roads to build a 3D models of the City.
- Integrating with other systems, Lane detection is just one aspect of autonomous driving.
- The presence of features such as Detecting other Vehicles or Pedestrians.

Thank You!


HSL
HSV
a.
Saturation
Lightness
Hue
e.
Saturation
Value
Hue
b.
$S_{HSL} = 1$
L
H
0°
120°
240°
360°
f.
$S_{HSV} = 1$
V
H
0°
120°
240°
360°
c.
$L = ½$
120°
240°
0°
H
d.
$H = 0° / 180°$
L
S_{HSL}
g.
$V = ½$
120°
240°
0°
H
h.
$H = 0° / 180°$
V
S_{HSV}

4.2.1. Third-Order B-Spline Curve Model

An n-order B-spline curve is defined as

$$p(t) = \sum_{i=0}^{n} p_i B_{i,n}(t) \quad (0 \le t \le 1) \tag{6}$$

where $p_i(i = 0, 1, 2, \ldots, n)$ is the position vector of the vertex, and $B_{i,n}(t)$ is the n-order basis function presented as follows:

$$B_{i,n}(t) = \frac{n!}{i!(n-1)!} t^i (1-t)^{n-i} \tag{7}$$

In the actual road scene, considering that the lane under complicated road conditions tend to be tortuous and variable, the third-order B-spline curve model was used to fit the lane lines. The mathematical expression corresponding to the third-order B-spline curve is as follows:

$$p(t) = p_0 B_{0,3}(t) + p_1 B_{1,3}(t) + p_2 B_{2,3}(t) + p_3 B_{3,3}(t) \tag{8}$$

By substituting $i = 0, 1, 2, 3$ into Equation (7) separately, the above equation is converted to

$$p(t) = (1 - 3t + 3t^2 - t^3)p_0 + (3t - 6t^2 + 3t^3)p_1 + (3t^2 - 3t^3)p_2 + t^3 p_3 \tag{9}$$

This equation is then expressed as the following matrix:

$$p(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (0 \le t \le 1) \tag{10}$$

where p_0, p_1, p_2 and p_3 correspond to the four control points of the third-order B-spline curve, which can be adjusted accordingly based on real-time road conditions.

OPEN CV

Open cv is an open-source library that is very useful for computer vision applications such as video analysis, CCTV footage analysis, and image analysis. OpenCV is written in C++ and has more than 2,500 optimized algorithms. When we create applications for computer vision that we don’t want to build from scratch we can use this library to start focusing on real-world problems.

26) How many types of image filters in OpenCV ?

- Averaging
- Gaussian Filtering
- Median Filtering
- Bilateral Filtering

27) How many types of video filters in OpenCV ?

- Color Conversion
- Thresholding
- Smoothing
- Morphology Gradients Canny
- Edge
- Detection
- Contours
- Histograms