Introduction to Straight Lane Finding

A Gentle CV Introduction for Self-Driving Car

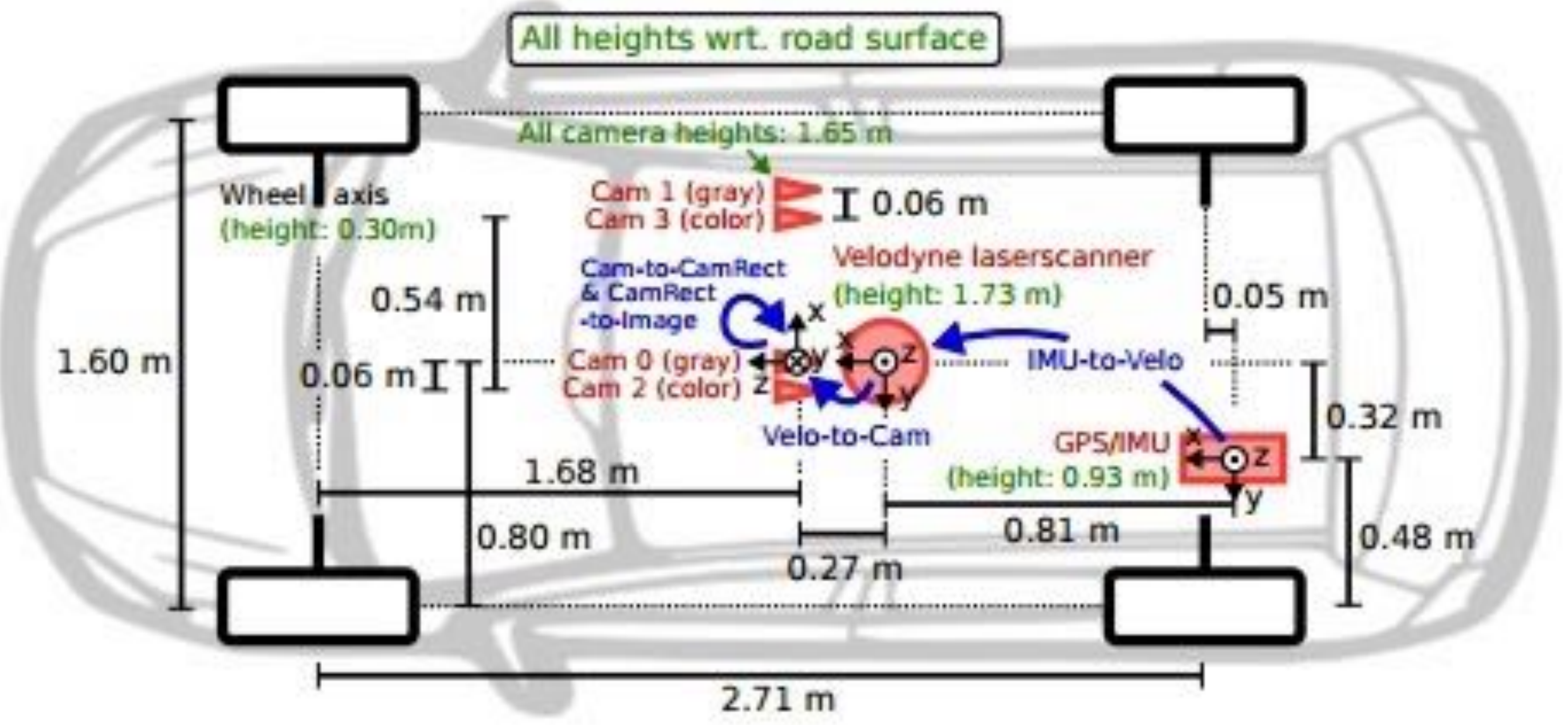
Question to start

Why are we doing lane finding ?

What are we going to do after lane finding?

How are we going to achieve it ?

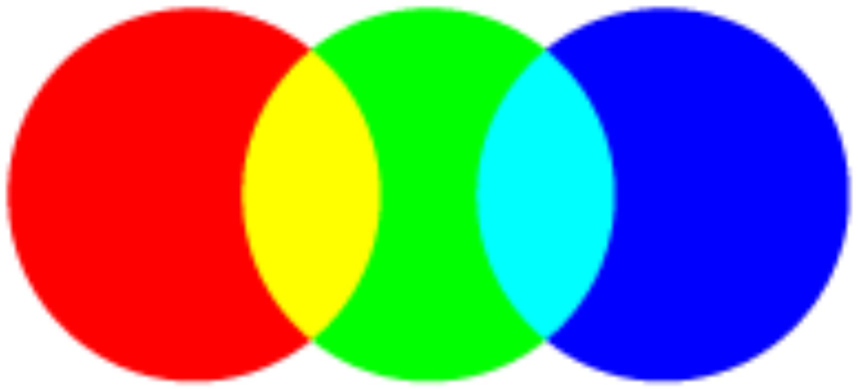
Sensors in Self-driving car

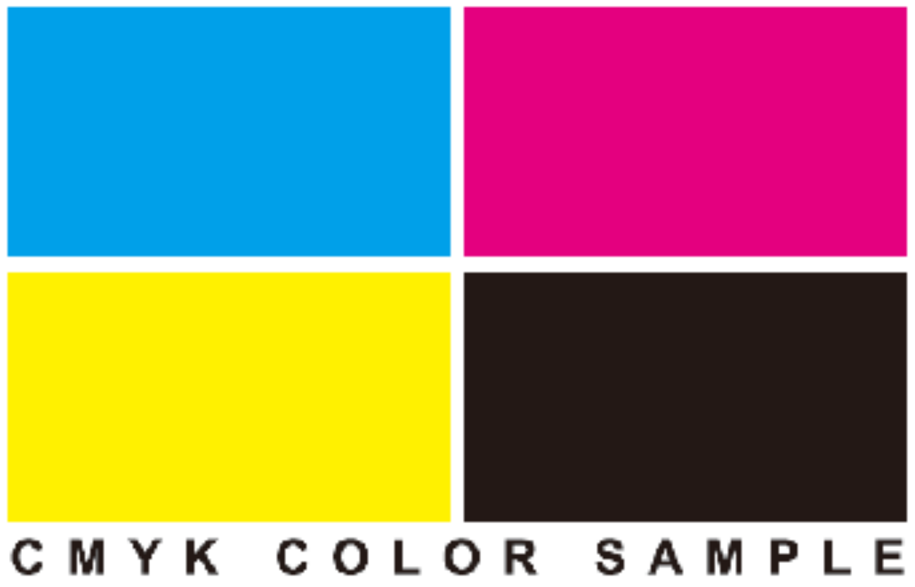


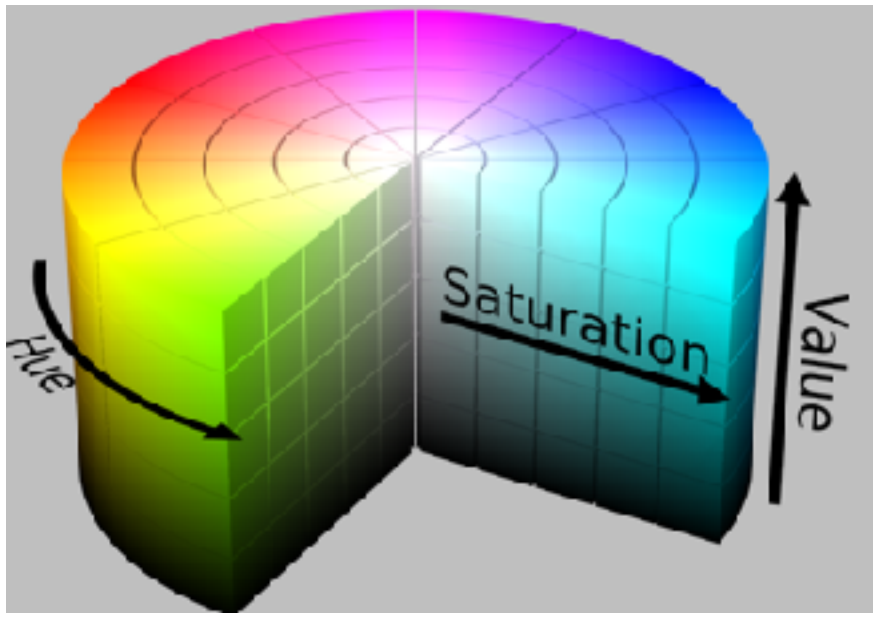
The Difficult Part of Building Driverless Cars

Taken from Wall Street Journal

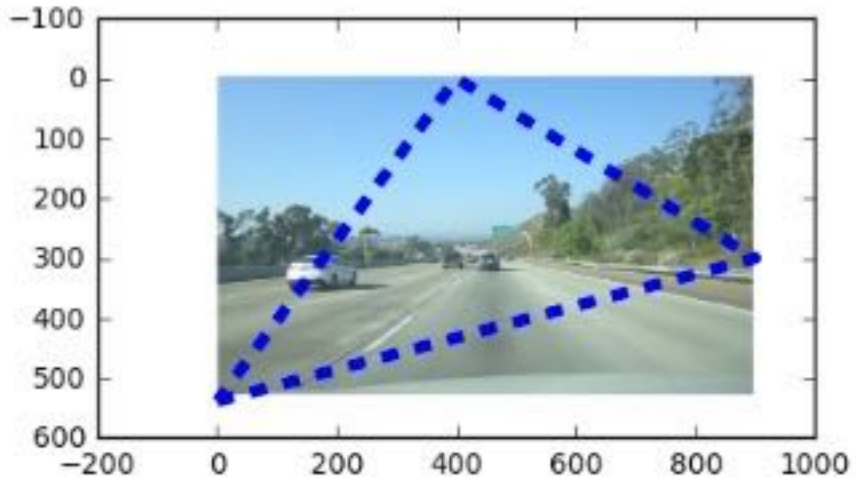
Most Common Color Selection

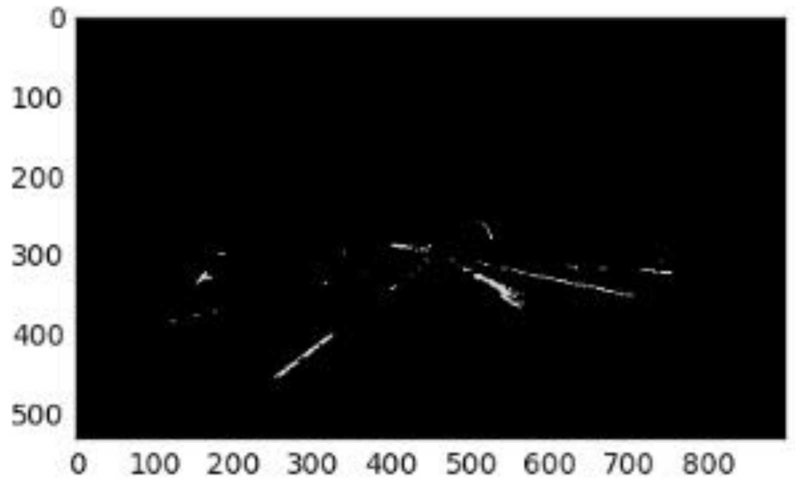


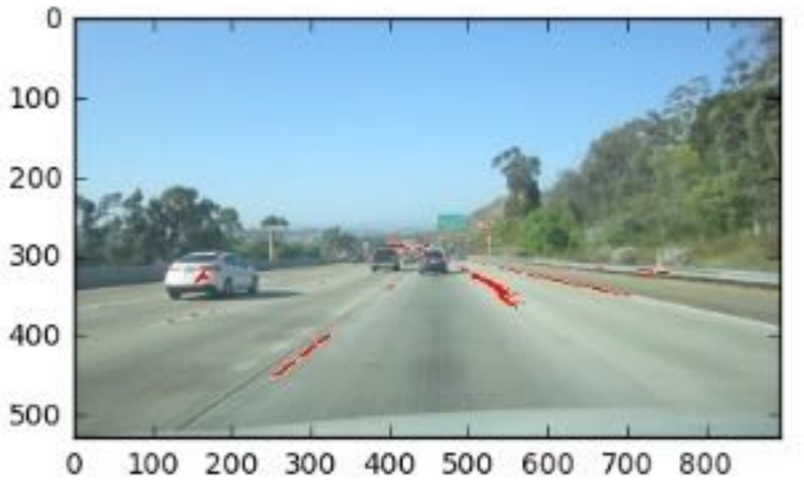




Color Region Mixed







Color Selection

I want to detect a specific color say, **black**, from a front face camera

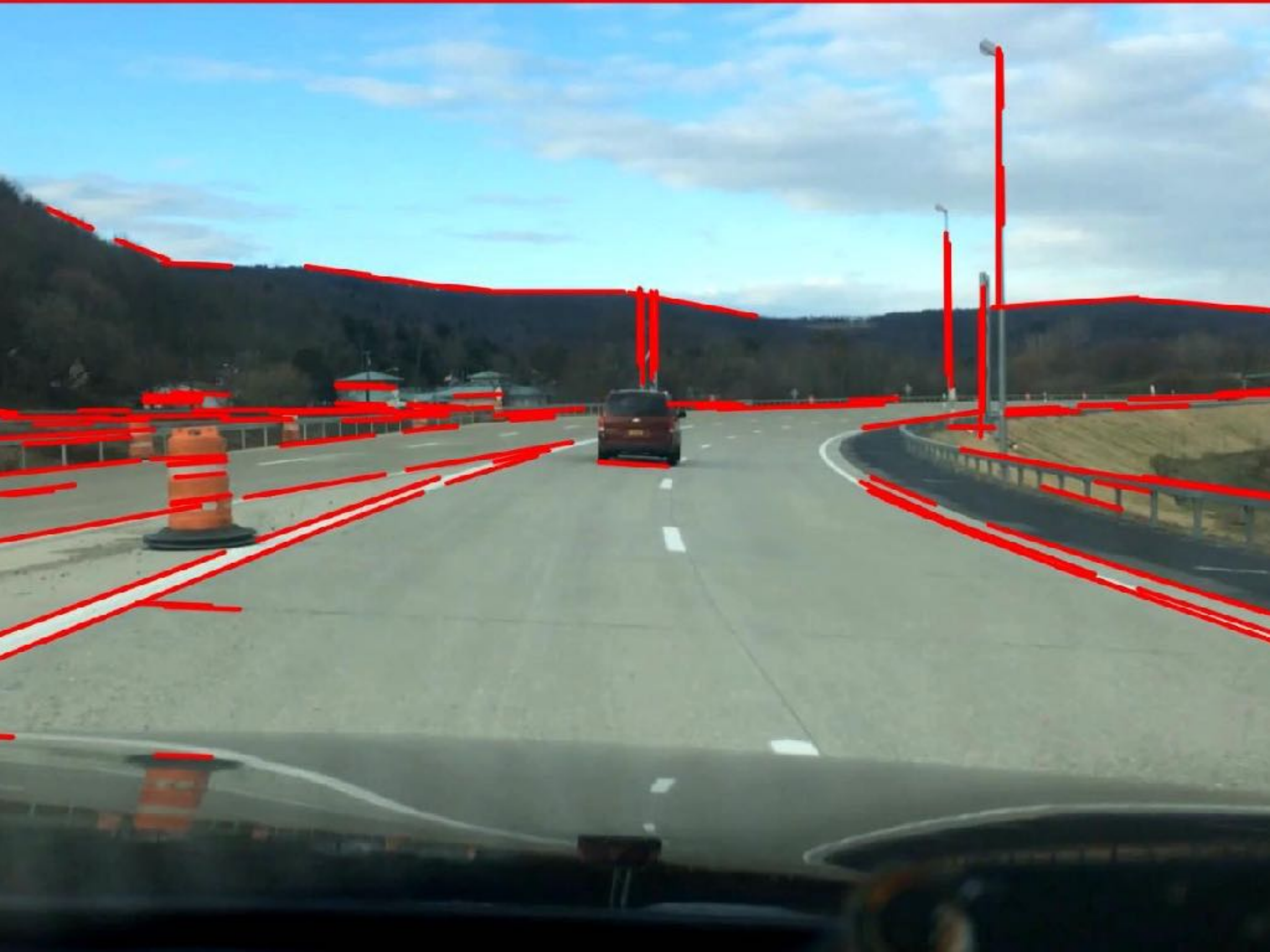
RGB: 0, 0, 0

CMYK: 0, 0, 0, 100

HSV: ??, ??, ??

Threshold

```
red_threshold = 200
green_threshold = 200
blue_threshold = 200
```



Here’s what we know so far

Color Region

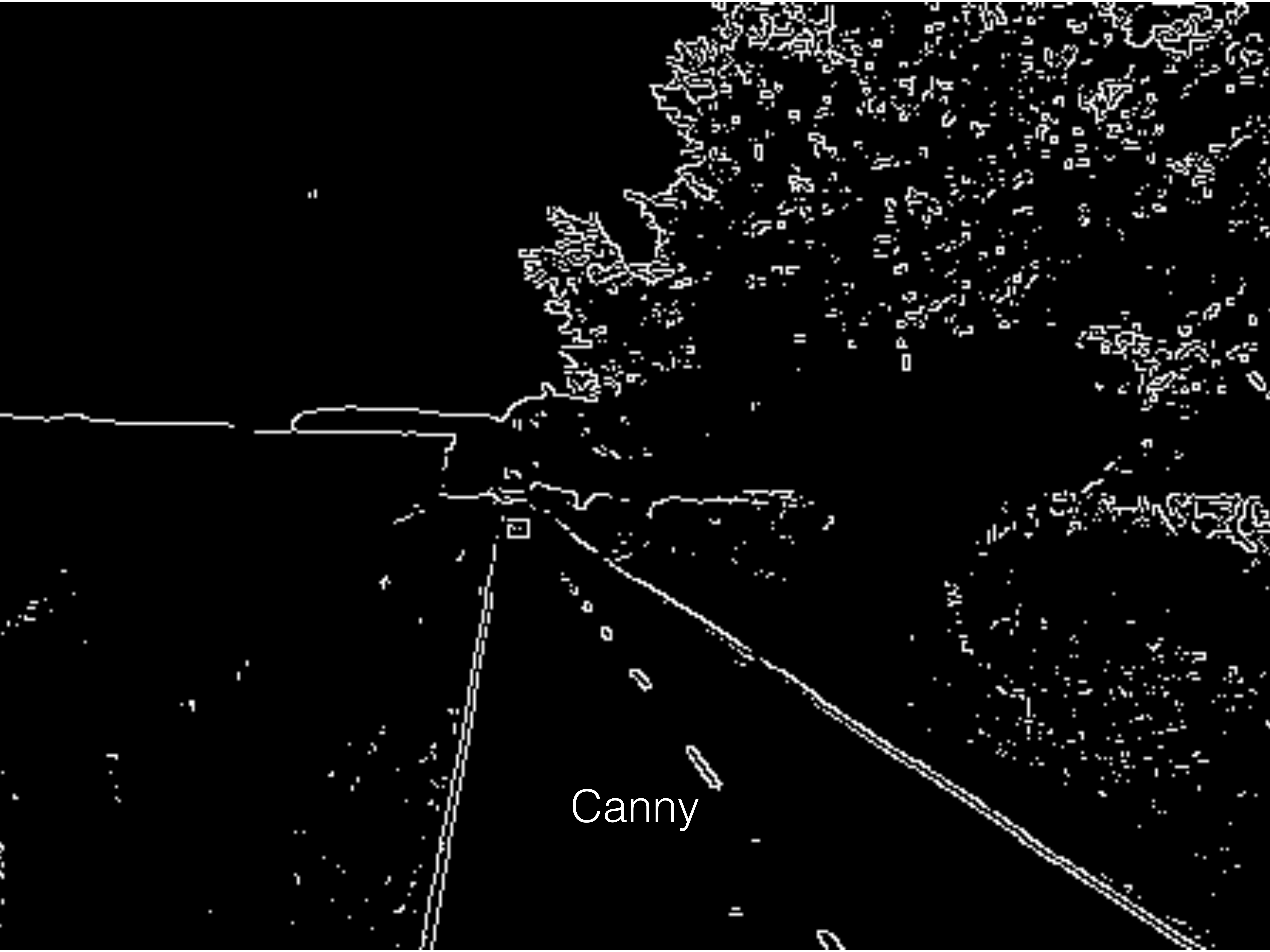
We did triangulation , mask the threshold, find region and mask the selection. Manual process concept.

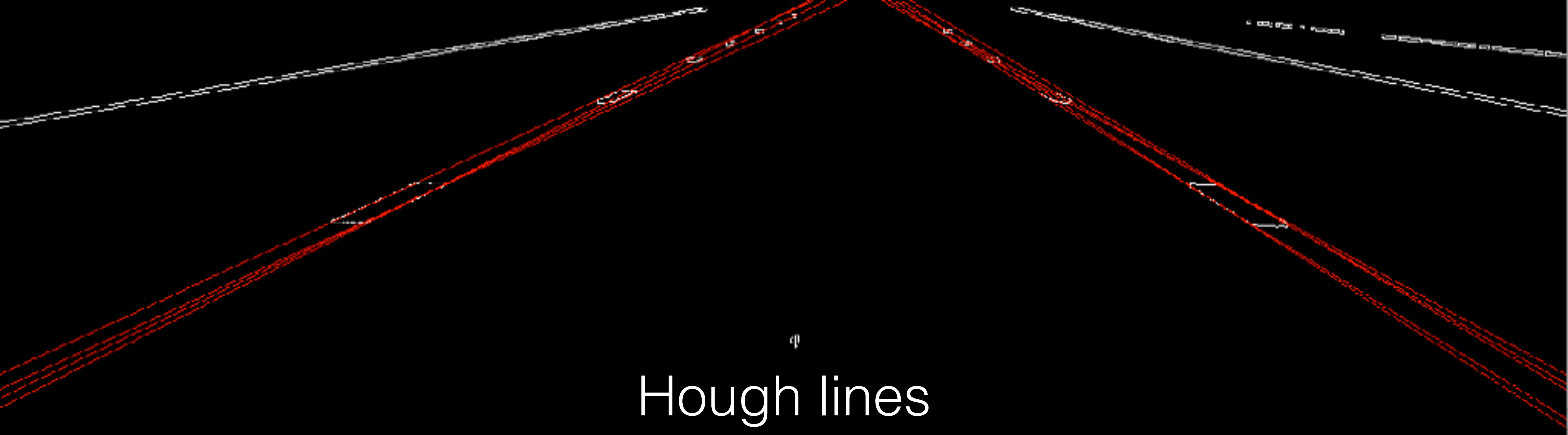
What is **`cv2.Canny ?`**

cv2.Canny(gray, low_threshold, high_threshold)

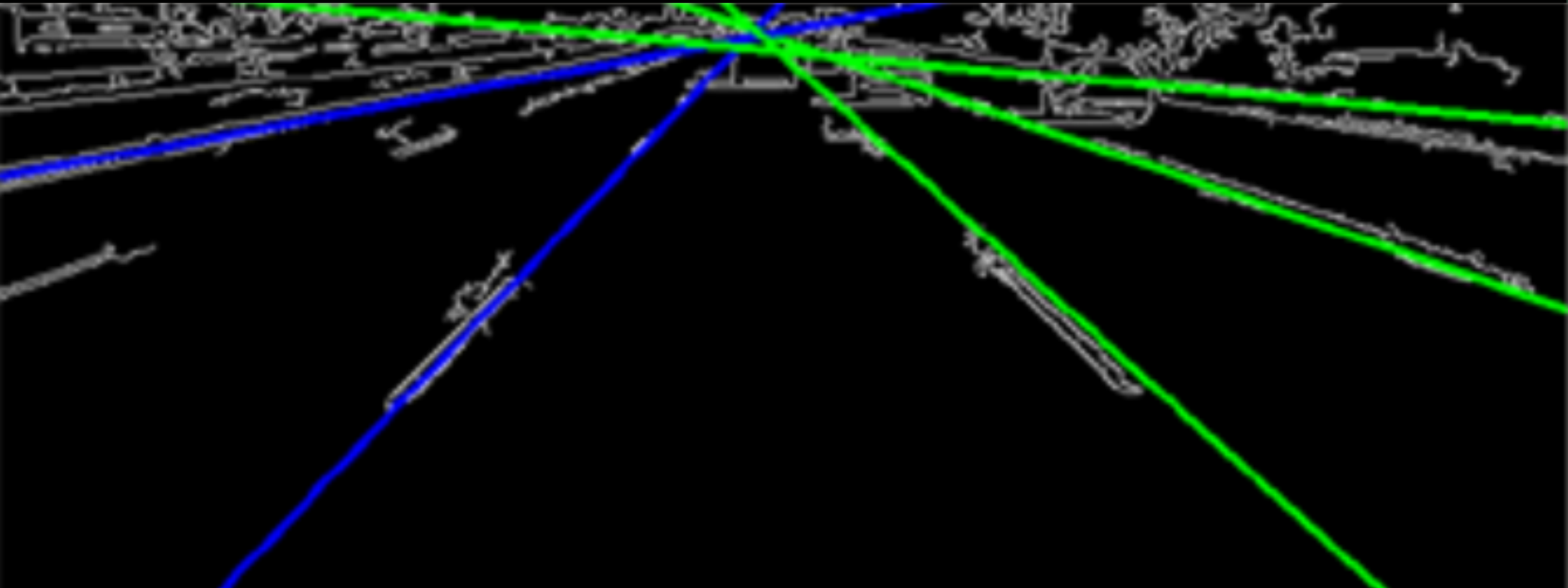
What is **`cv2.HoughLines ?`**

cv2.HoughLinesP(edges, rho, theta, threshold, np.array([]),
min_line_length, max_line_gap)


Canny



Hough lines



Here’s what we know so far

Color Region

We did triangulation , mask the threshold, find region and mask the selection. Manual process concept.

Canny Edge

Canny detection goal is to identify the boundaries of an object in a image. Convert to grayscale then take the gradient. After that, trace out the edges with strongest gradient.

Hough Lines

The Hough transform is a feature extraction technique used in image analysis, computer vision, and image processing The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure.

Steps For Finding Lanes

1. Determine the Region of Interest (ROI) by slicing the image (focus the middle part)

2. Grayscale the ROI

3. Equalized the grayscaled ROI with **`cv2.equalizeHist`**

4. Apply Gaussian blur to (3)

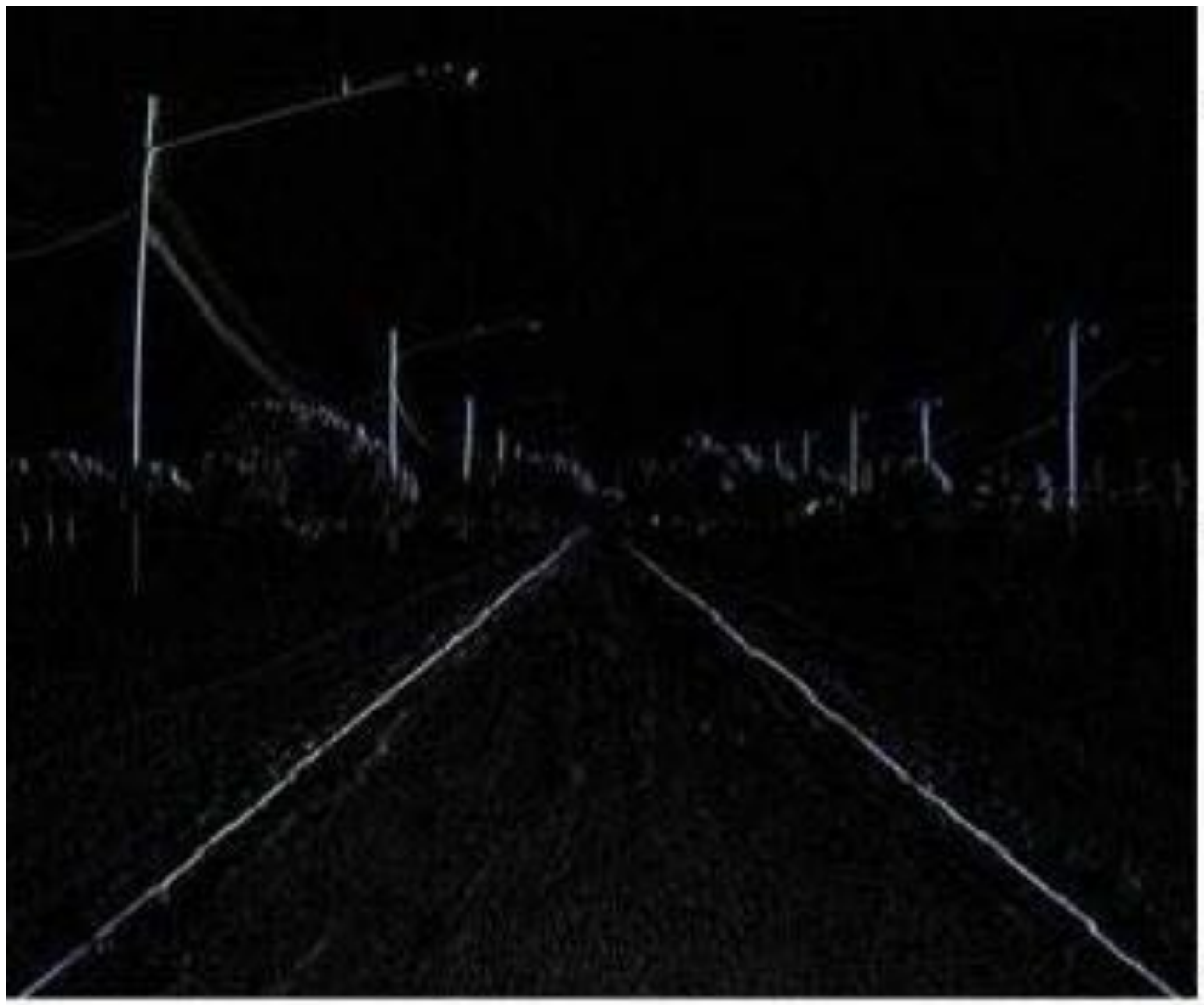
5. Threshold (4) by using **`cv2.adaptiveThreshold`**

6. Skeletonize (5) by using skimage.morphology.skeletonize

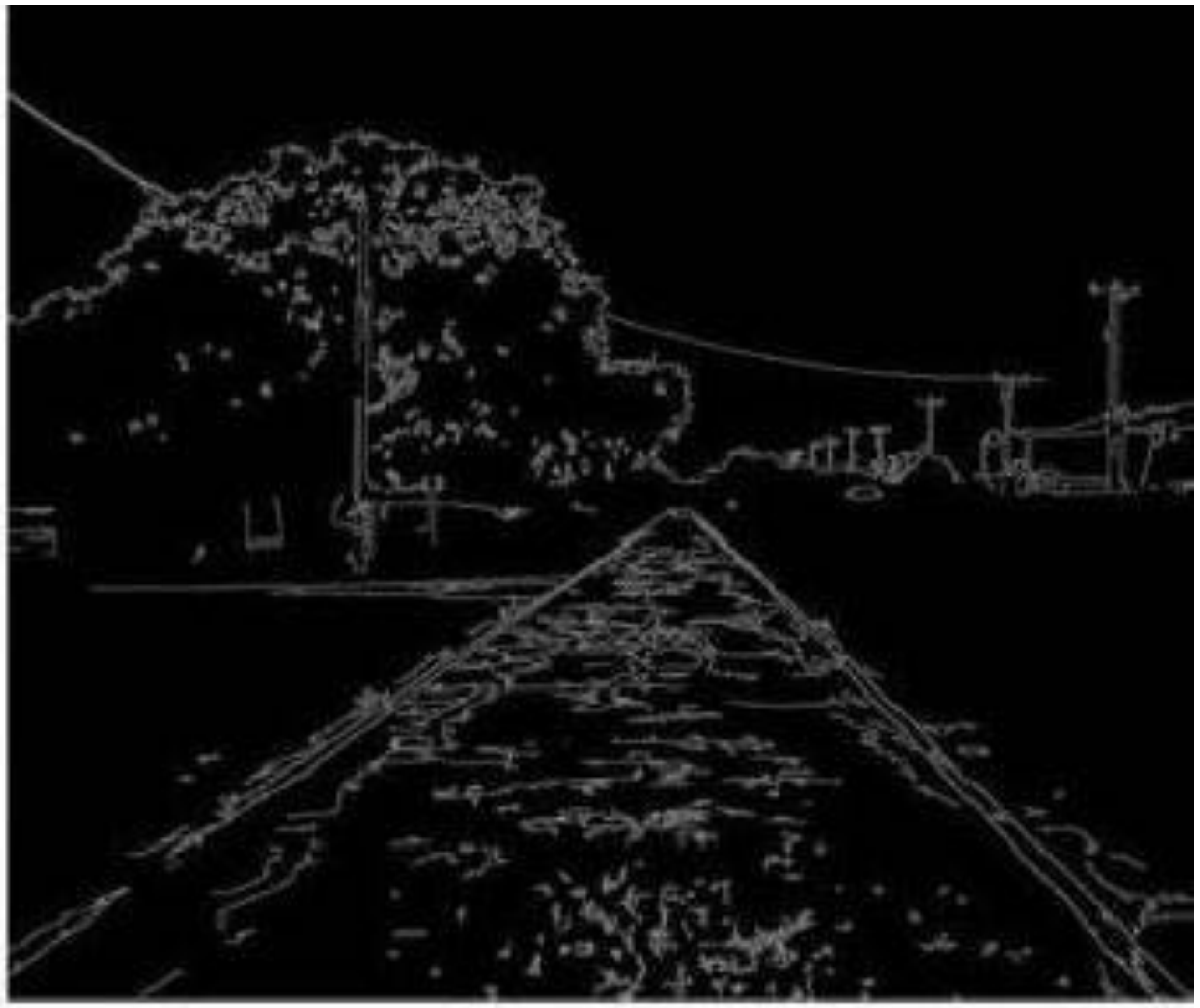
7. Apply the **`cv2.HoughLines`** on (6)

But wait, what about

`Sobel, Harris,`
`and Probabilistic Hough?`

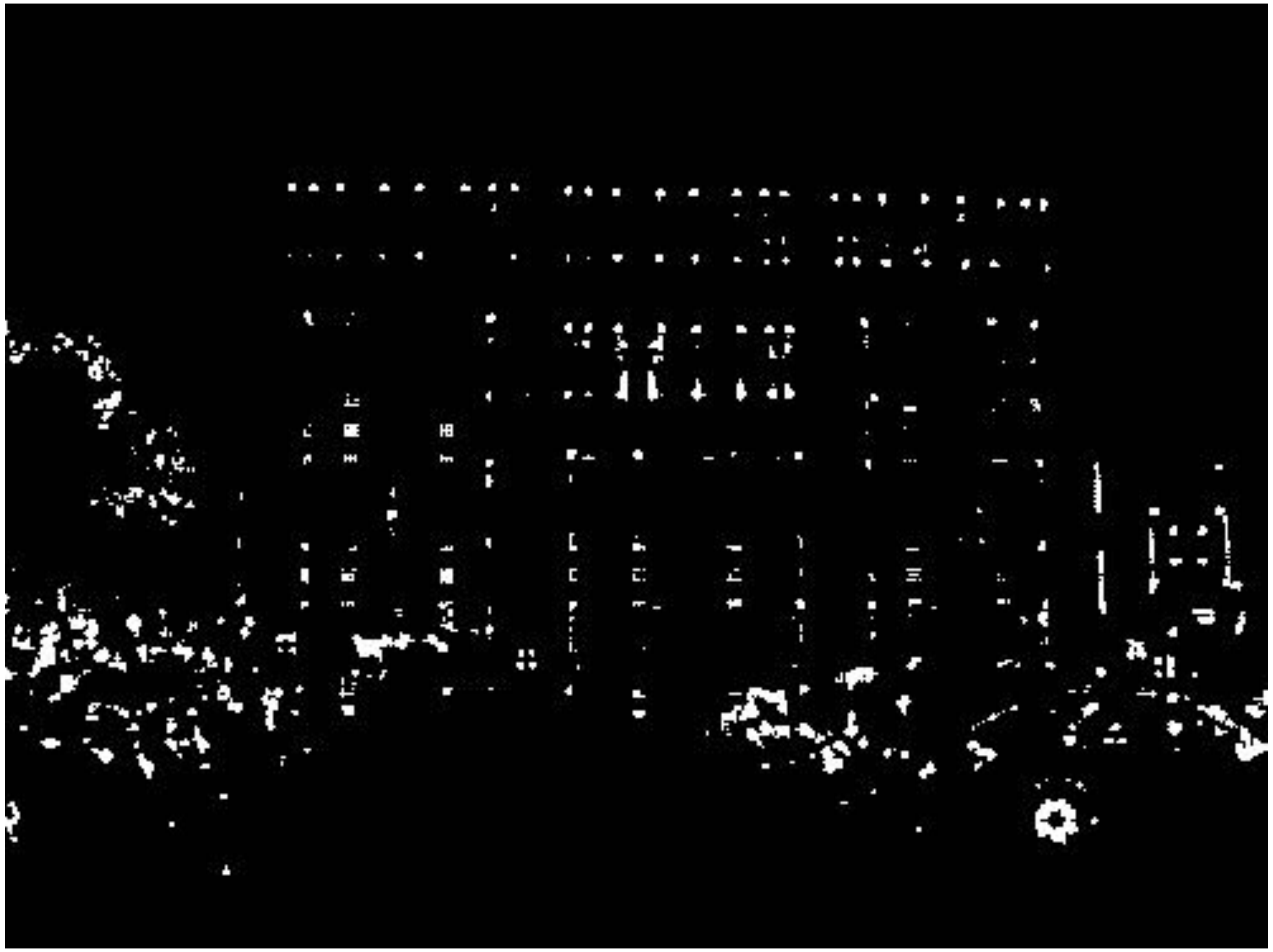


Sobel Edge Detection



Canny Edge Detection

Sobel detection refers to computing the gradient magnitude of an image using 3x3 filters. Where "gradient magnitude" is, for each a pixel, a number giving the greatest rate of change in light intensity in the direction where intensity is changing fastest.

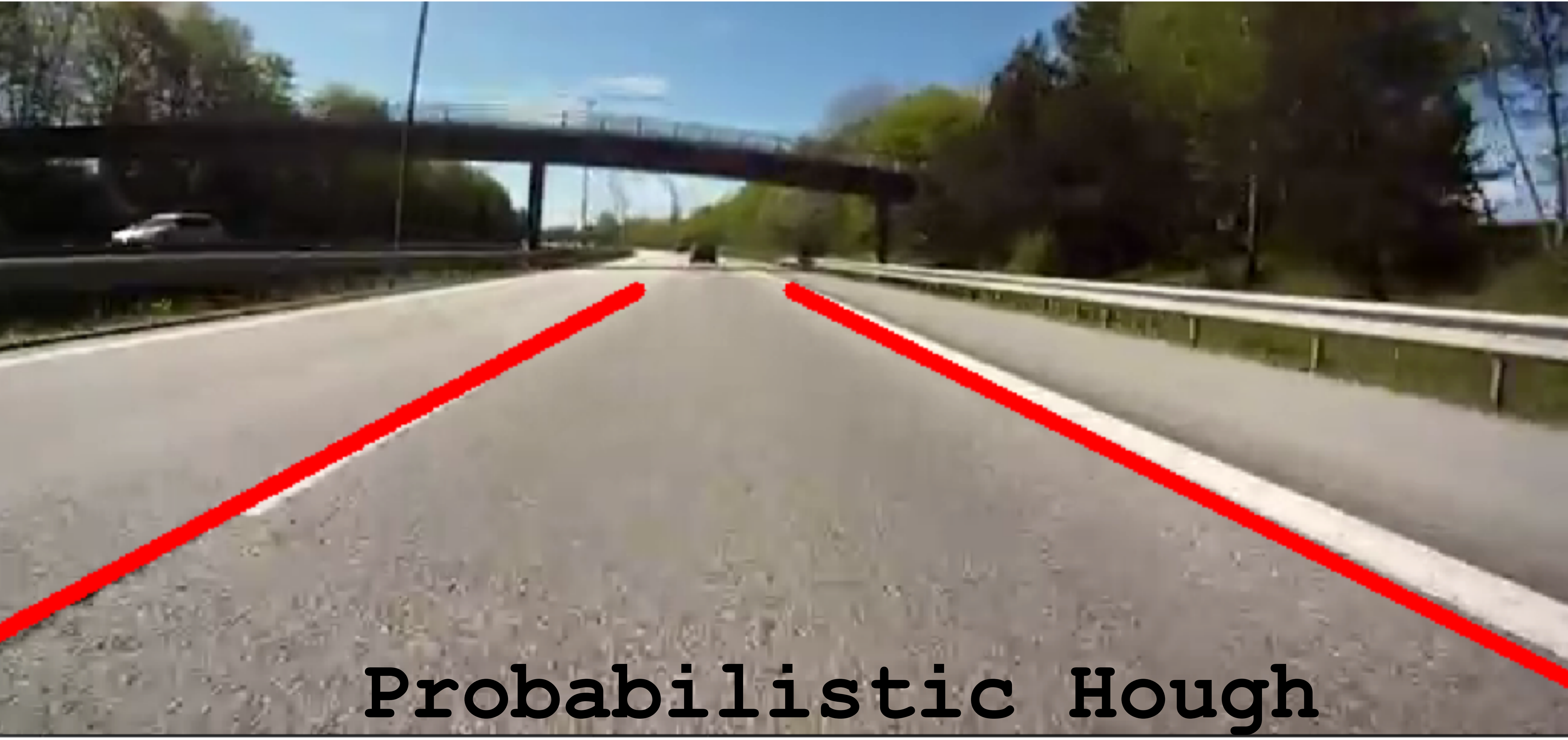


Harris

Harris Corner Detection looks for corners because corners are translation invariant and rotation invariant while distinguishable, unlike edges. These properties make corners good feature candidates.

First it computes the horizontal and vertical derivatives (edges) of an image, then it performs cross correlation on these edge images to highlight corners, and then it performs non-maximum suppression to get rid of the edge features.

Probabilistic Hough



Use line polar coordinate

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

where:

ρ (rho) = distance from origin to the line. [-max_dist to max_dist].

max_dist is the diagonal length of the image.

θ = angle from origin to the line. [-90° to 90°]

Probabilistic Hough

Algorithm:

1. Perform Corner or Edge Detection as input image
2. Rho range and Theta range creation
3. Setup Hough accumulator in 2D army with numbers equal from p and theta
4. Find the total points/pixels contributed for potential line lanes
5. Perform Peak finding by applying threshold

Quiz

Let's say your project manager asked; what are some very fast, less computation edge detection techniques?

A. Canny   B. Harris   C. Sobel   D. Marr-Hildreth

Functions

Some OpenCV functions (beyond those introduced in the lesson) that might be useful for this project are:

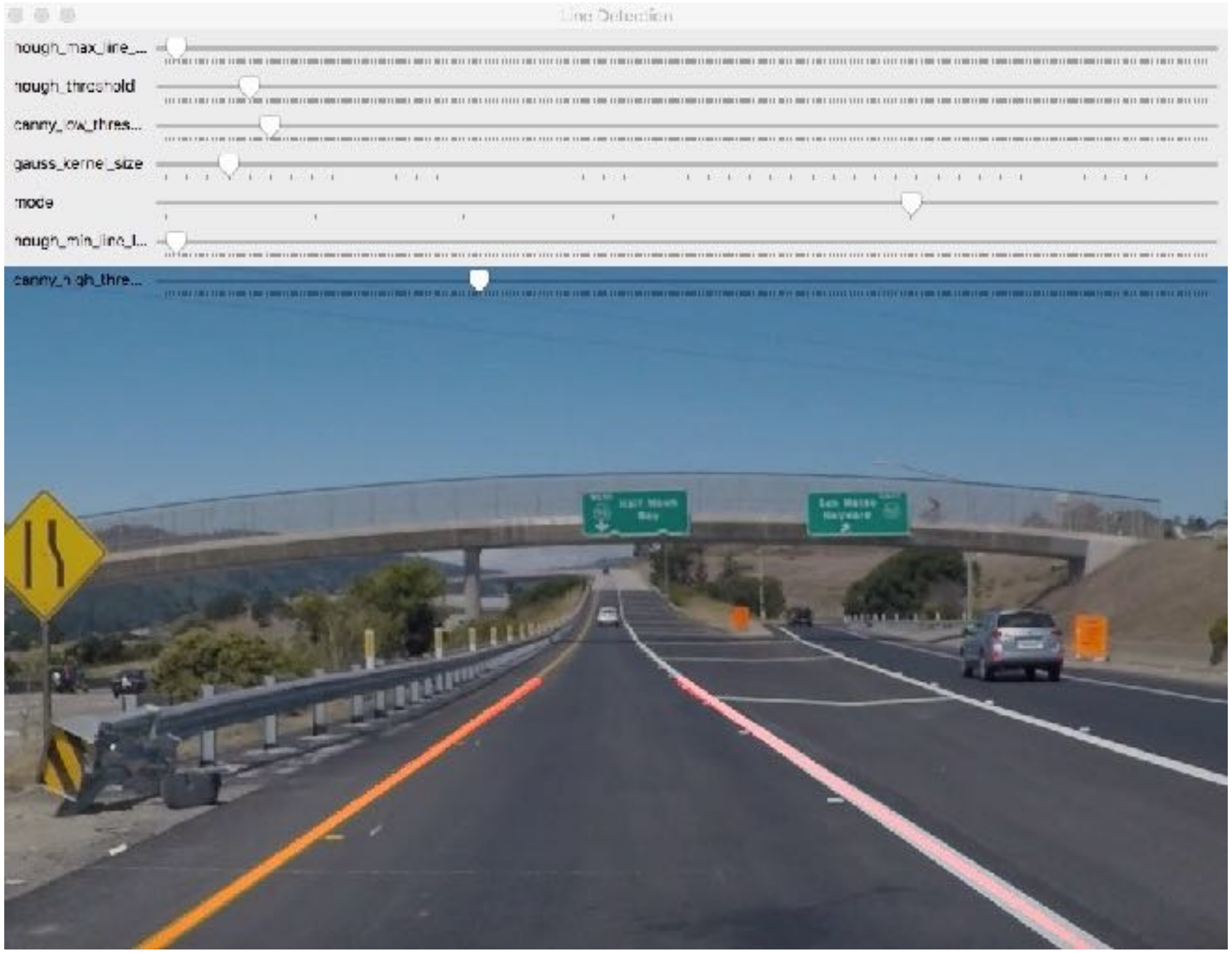
`cv2.inRange()` for color selection
`cv2.fillPoly()` for regions selection
`cv2.line()` to draw lines on an image given endpoints
`cv2.addWeighted()` to coadd / overlay two images
`cv2.cvtColor()` to grayscale or change color
`cv2.imwrite()` to output images to file
`cv2.bitwise_and()` to apply a mask to an image

Moving to the Project

I would recommend considering using the Probabilistic Hough Line Transform for our application. In OpenCV's Python API, it's implemented in the function, cv2.HoughLinesP.

Lane detection is an essential component of many intelligent vehicle applications, including Lane Following (LF), Lane Keeping Assistance (LKA), Lane Departure Warning (LDW), lateral control, Intelligent Cruise Control (ICC), Collision Warning (CW) and eventually autonomous vehicle guidance.

Let’s Prototype



hough_max_line_...
hough_threshold
canny_low_thres...
gauss_kernel_size
mode
hough_min_line_l...
canny_high_thre...

Here’s what we learn so far

Color Region

We did triangulation , mask the threshold, find region and mask the selection. Manual process concept.

Canny Edge

Canny detection goal is to identify the boundaries of an object in a image. Convert to grayscale then take the gradient. After that, trace out the edges with strongest gradient.

Hough Lines

The Hough transform is a feature extraction technique used in image analysis, computer vision, and image processing The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure.

Harris and Sobel

Harris Corner Detection looks for corners because corners are translation invariant and rotation invariant while distinguishable, unlike edges. Sobel detection refers to computing the gradient magnitude of an image

Question to start

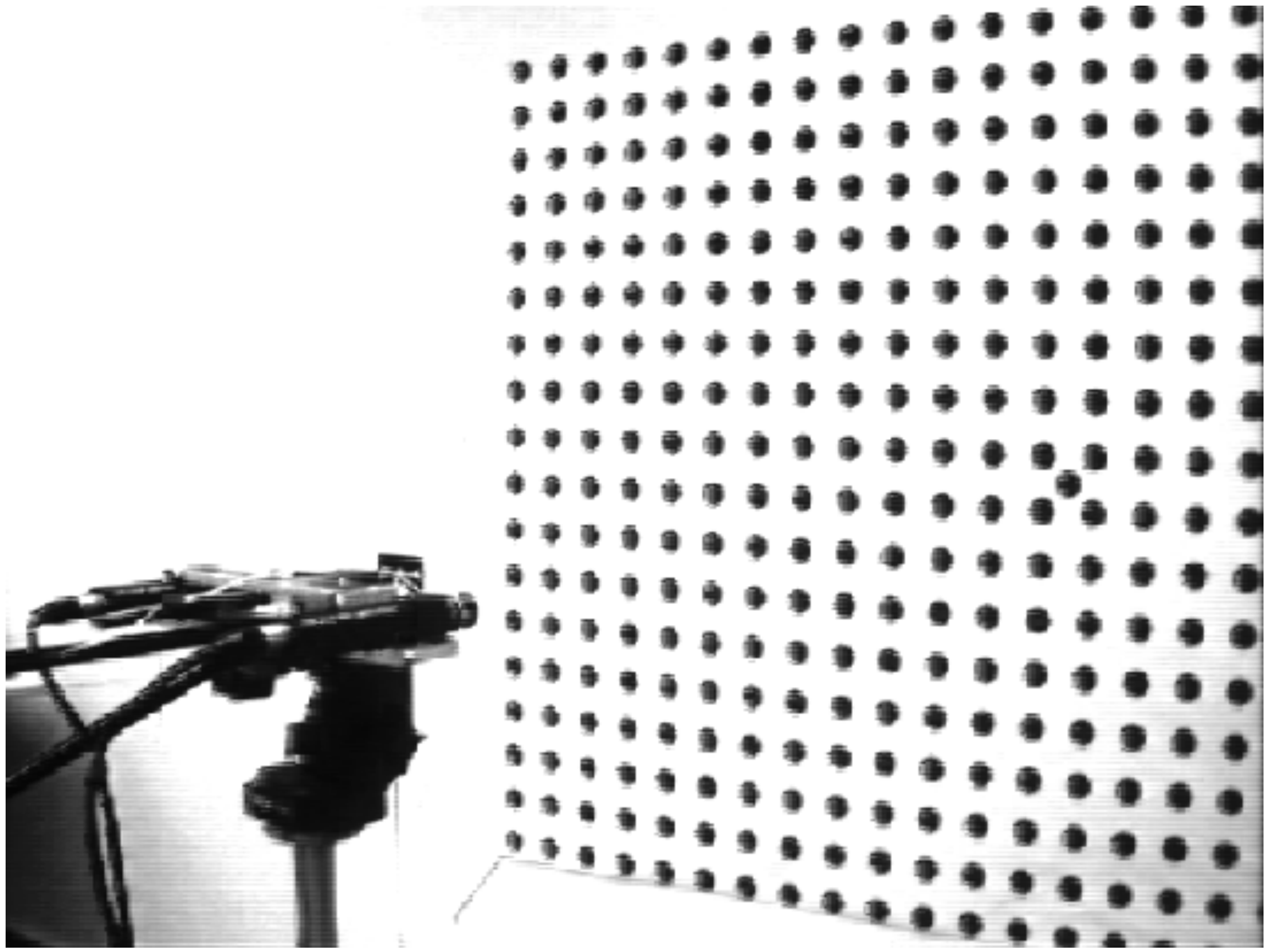
Why are we doing camera calibration ?

How do we do camera calibration ?

What are the steps to do camera calibration ?

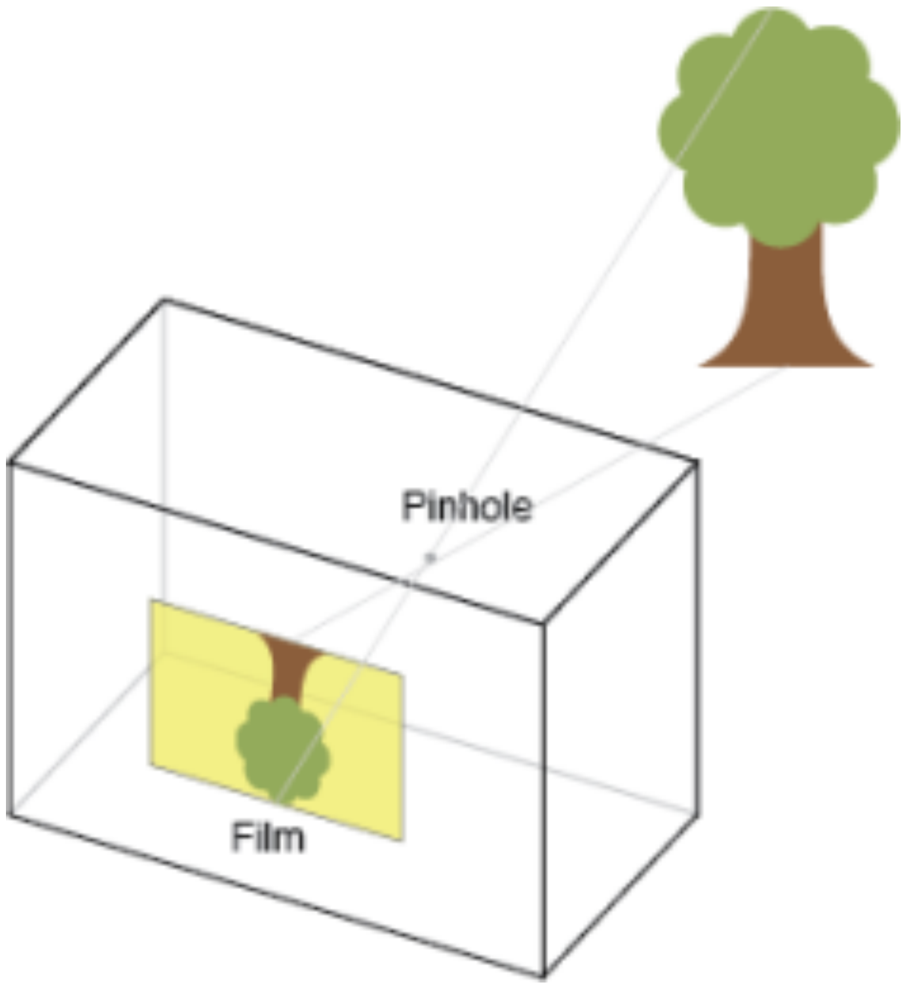
Calibration Concept

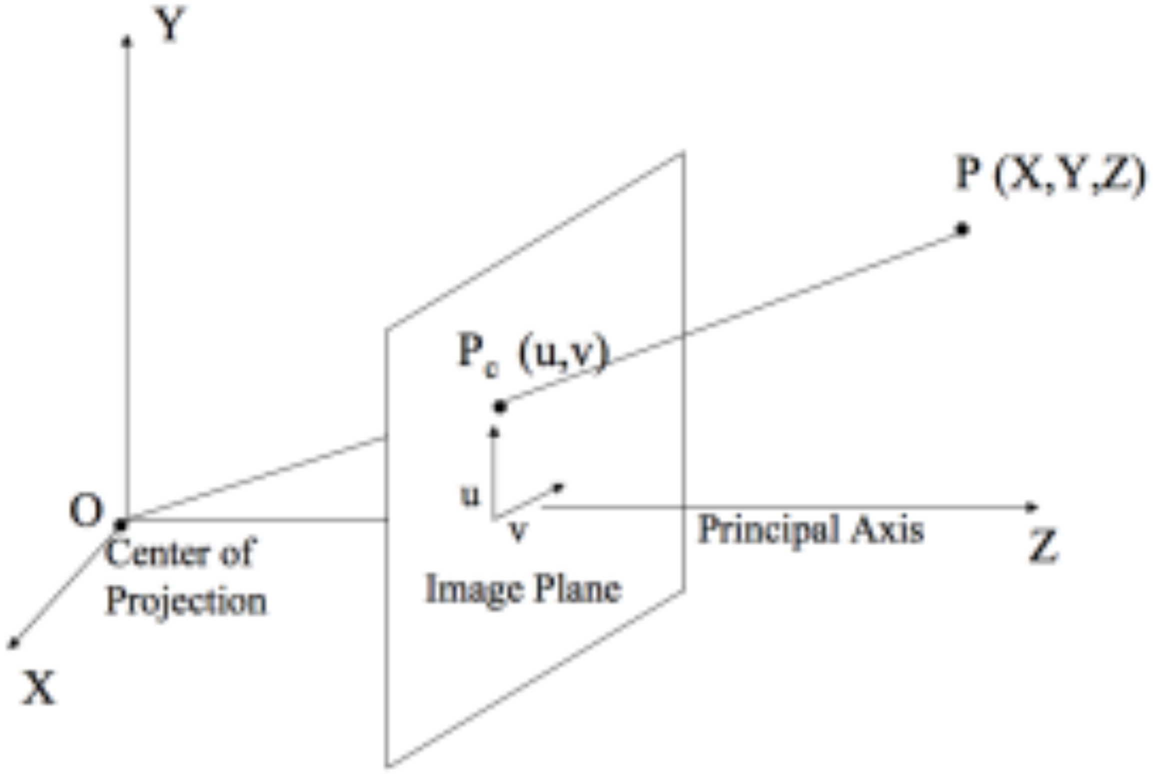
Our world is 3D so we need to extract number of metrics information from 2D images camera.



Majority of images captured in self-driving car are distorted. We need calibrate the camera in order to define the size, shape, field of view and distance of an 3D object.

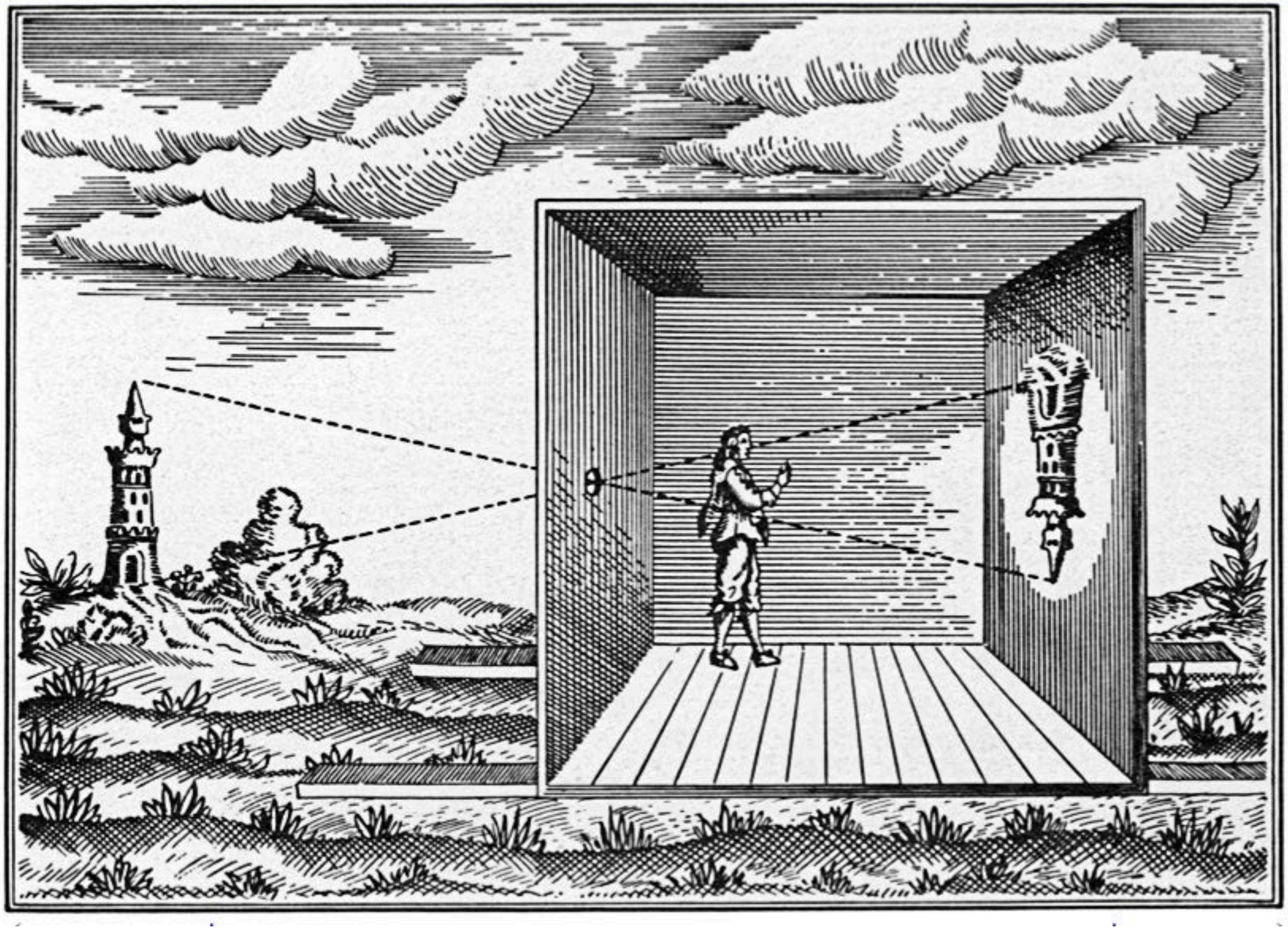
Pinhole Camera



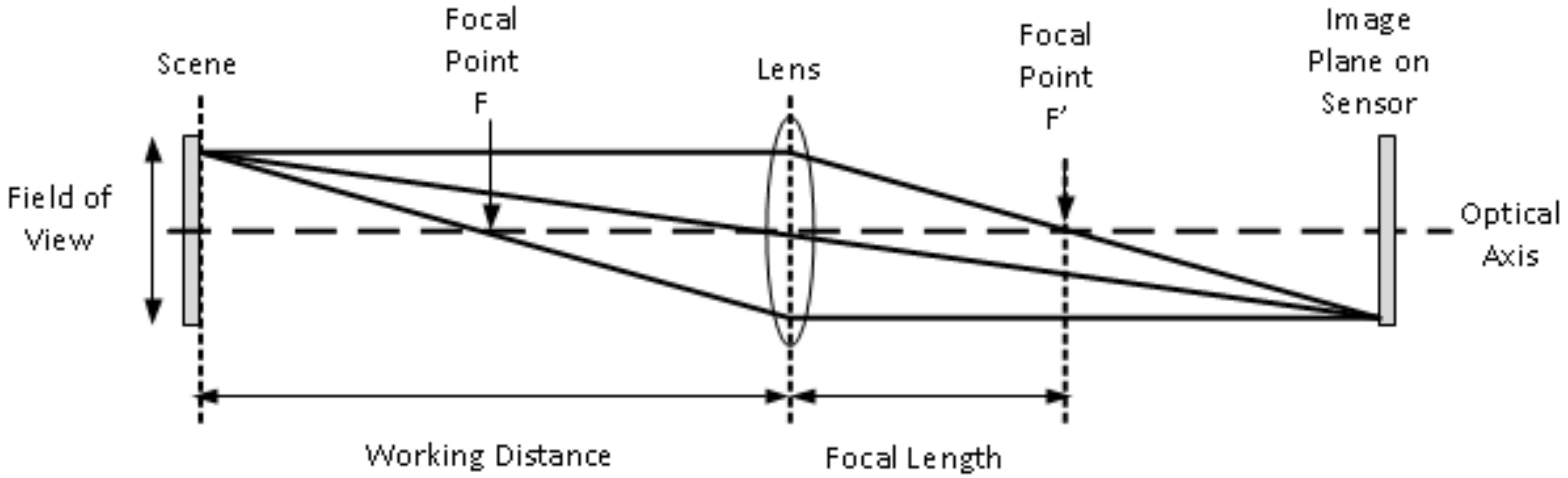


In industrial application, when you need to find a camera that suits your needs, you might overwhelmed with the concepts. By knowing Pinhole Camera, you should be able to grab the fundamental knowledge to pick right cameras for your self-driving car.

Camera Obscura



Quick Quiz

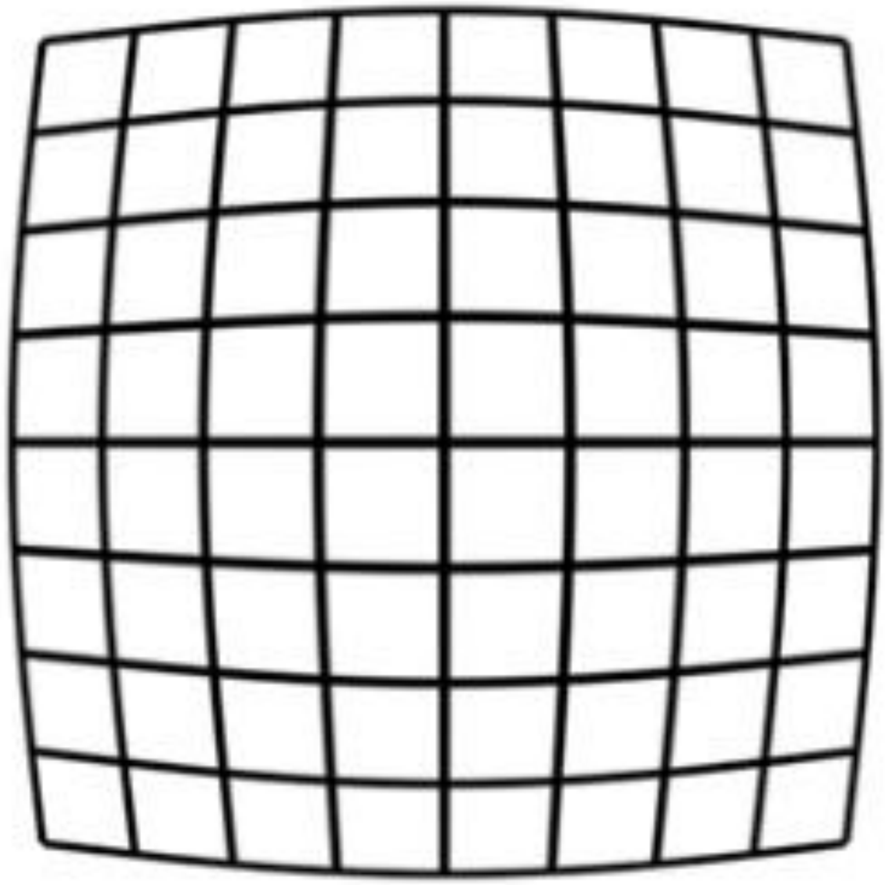


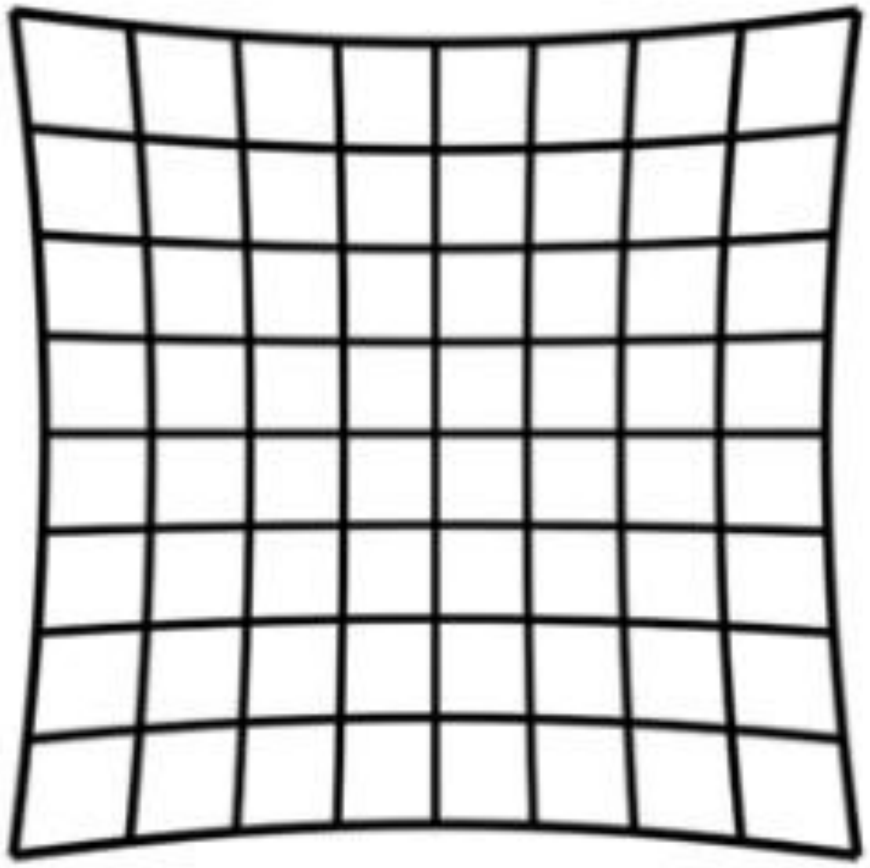
Who is actually giving image distortion ?

A. Camera Sensor

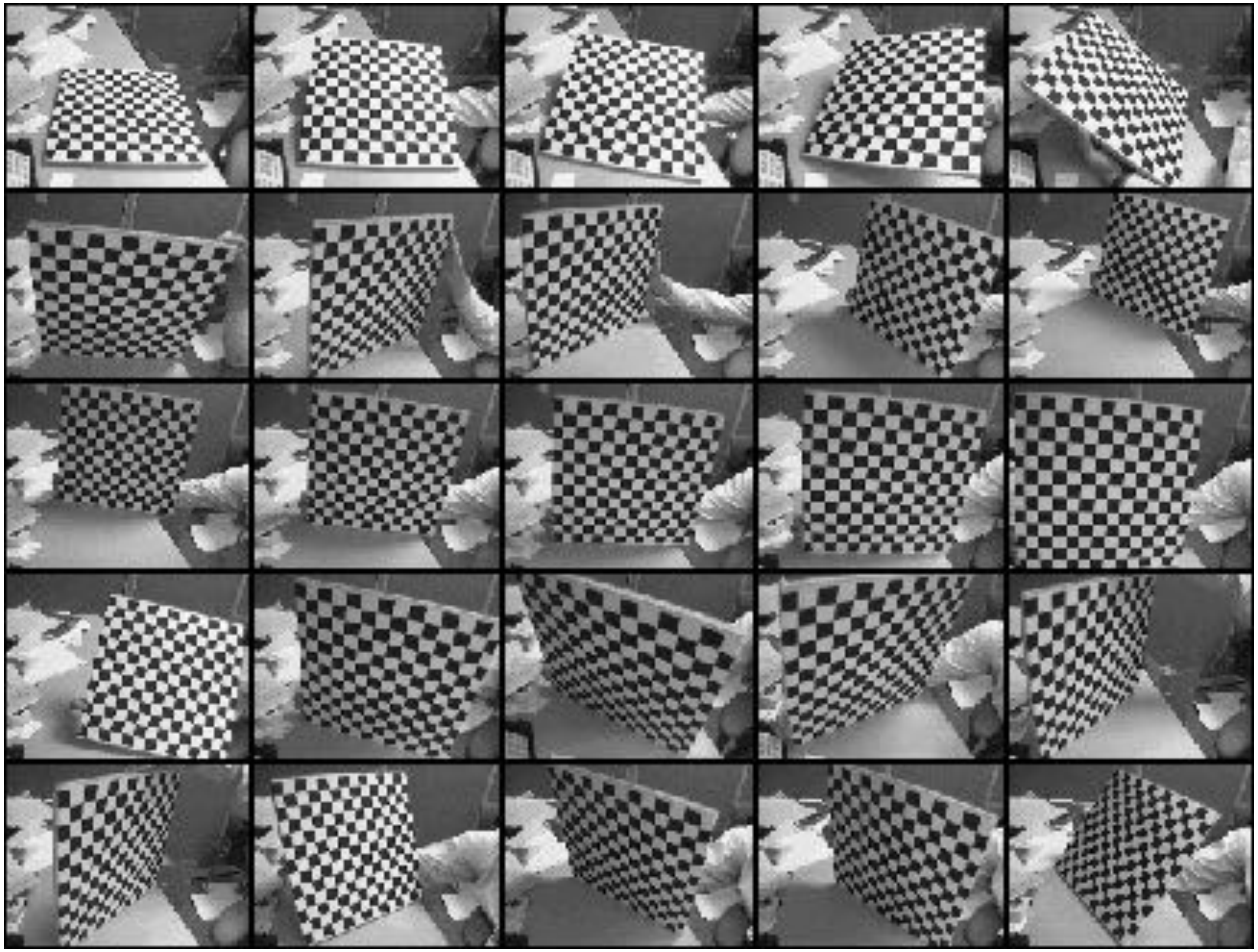
B. Camera Lens

Distortion

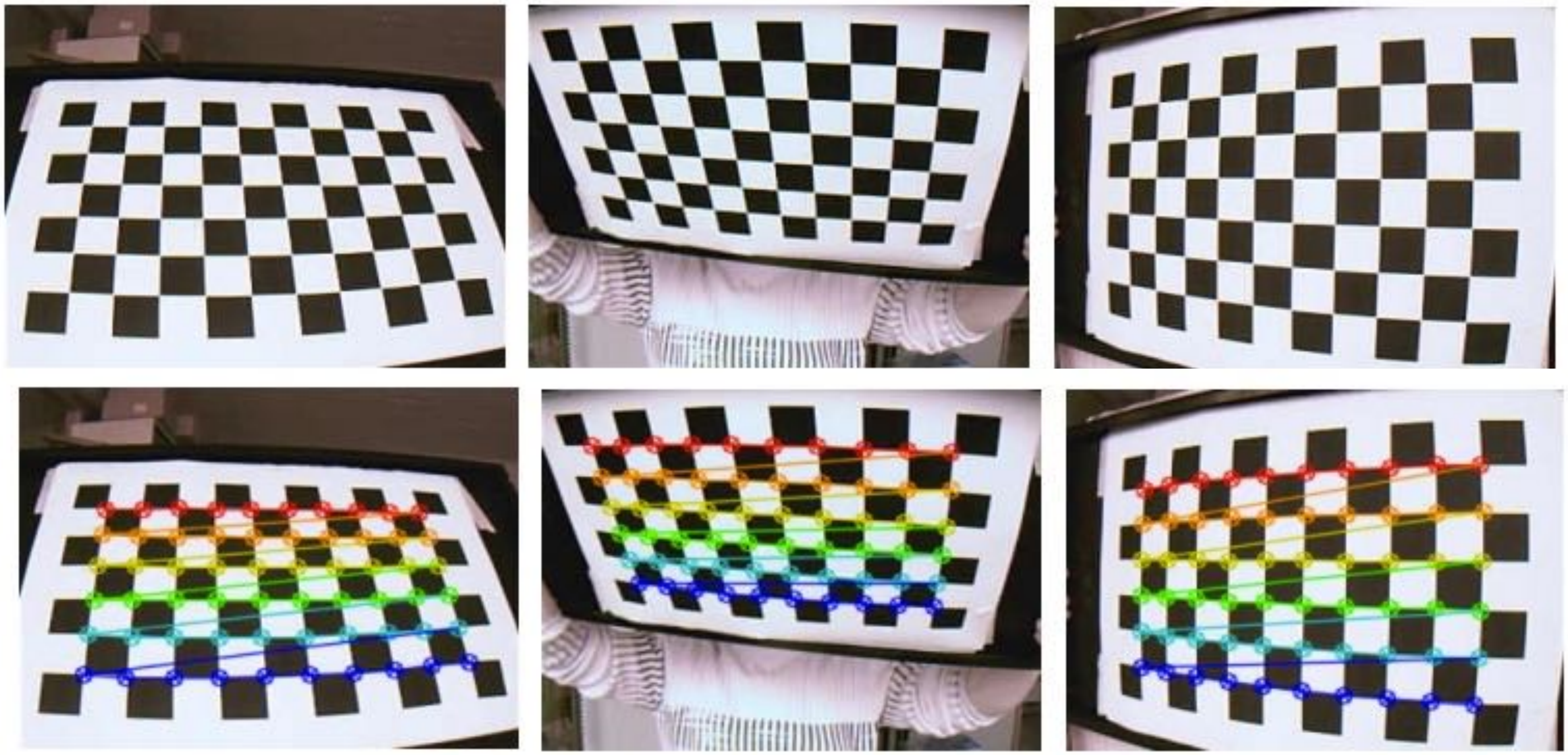

Barrel Distortion


Pincushion Distortion

Measuring Distortion

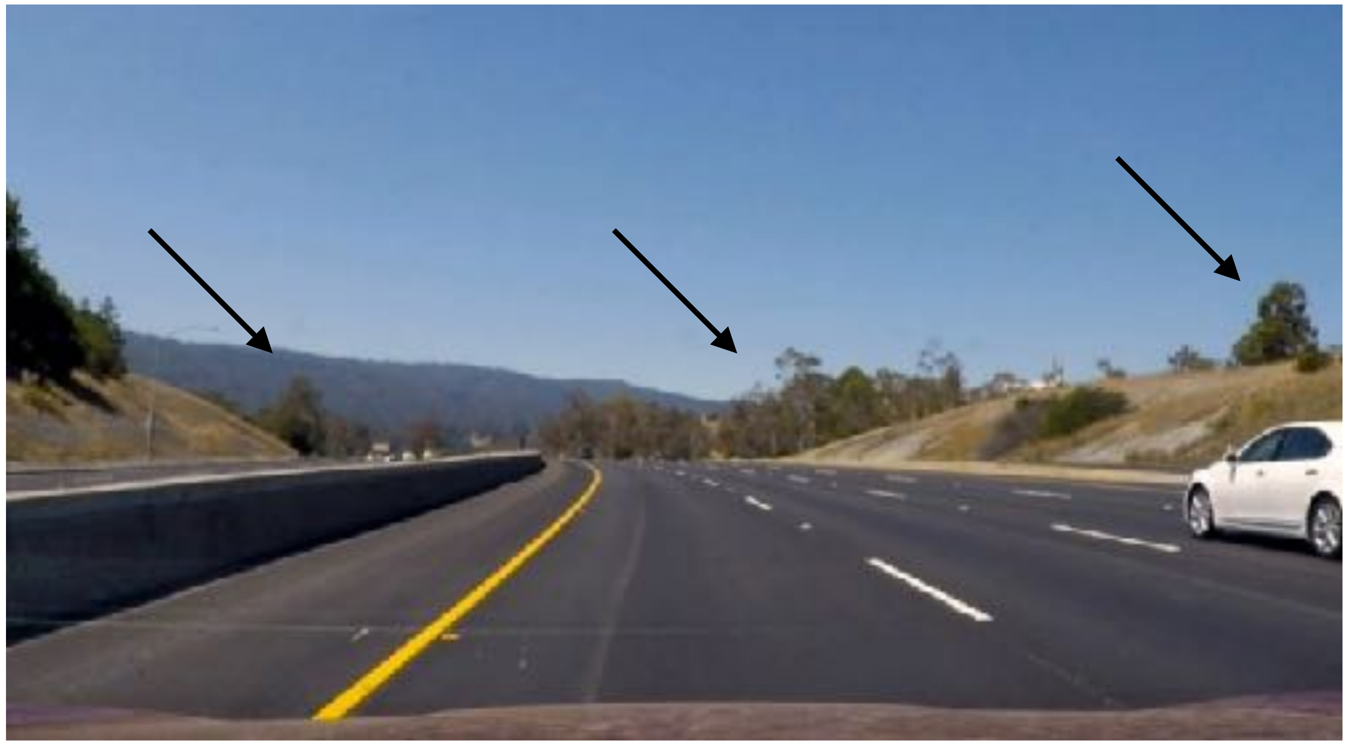


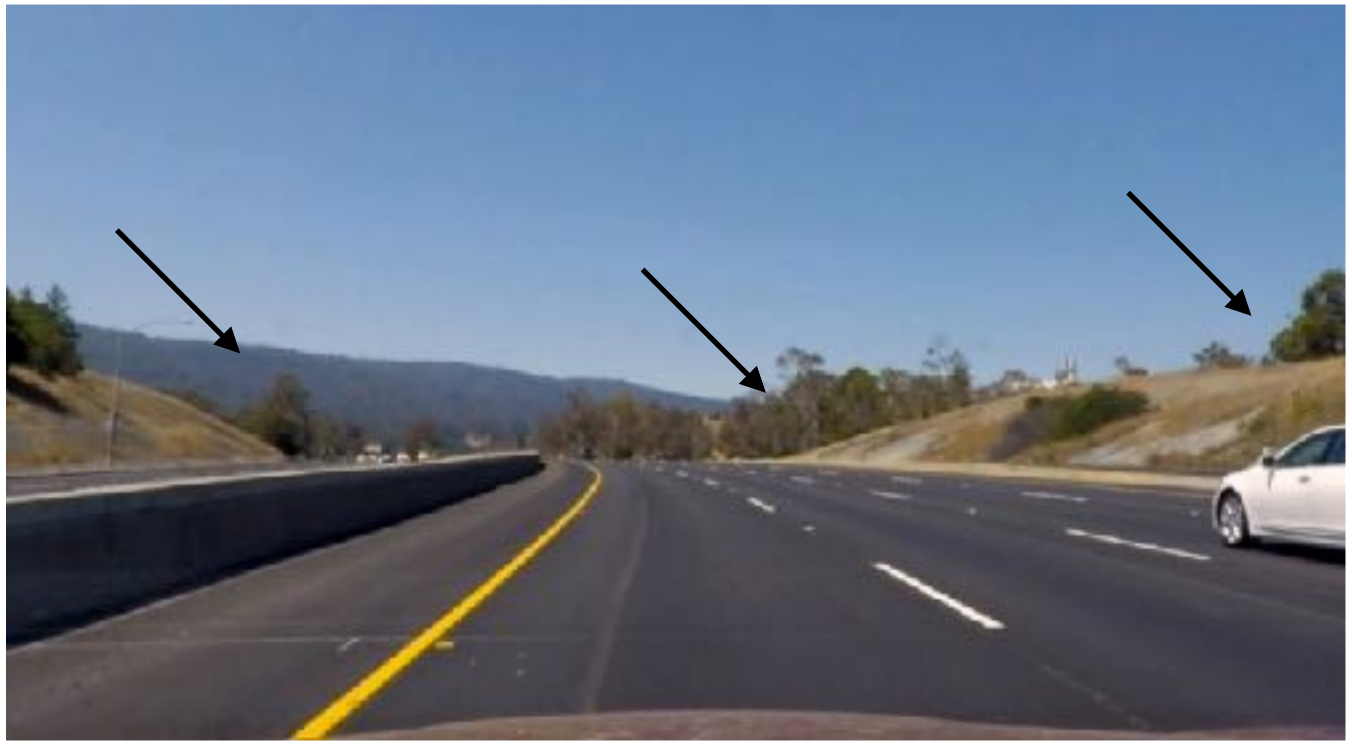
Distortion Correction



We will use 2D pattern checkerboard for 2D camera calibration

Results


Before


After

Application in Lane Curvature