

# Lane Detection

Lewis Collum



## CONTENTS

<b>1</b>	<b>Summary</b>	<b>1</b>
<b>2</b>	<b>Black and White Lane Mask from Raw Frame</b>	<b>2</b>
<b>3</b>	<b>Line Segment Coordinates from Lane Mask</b>	<b>3</b>
<b>4</b>	<b>Lane Angle Error from Line Segment Coordinates</b>	<b>4</b>

## 1 SUMMARY

The goal of the Lane Model is to detect a lane and to find the error between the vehicle's trajectory and the center of the lane. As the vehicle is driving, we will try to minimize this error, shown in figure 1, by steering the vehicle accordingly.

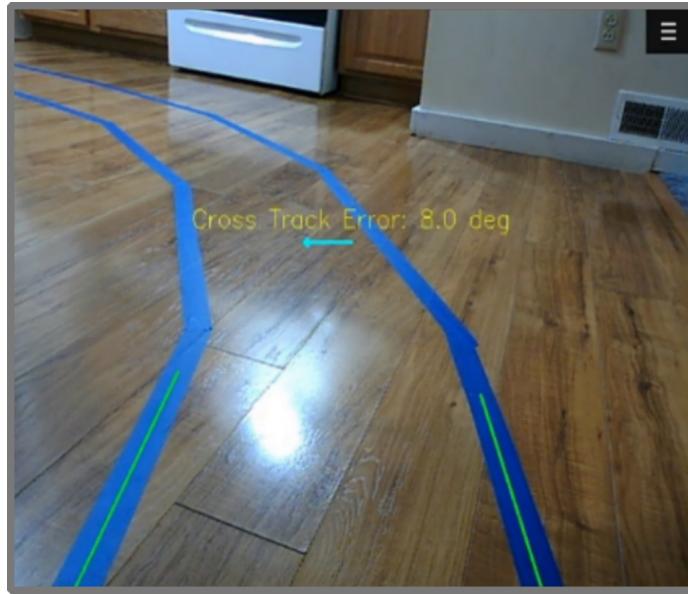


Fig. 1. Error generated by the lane model that will be used to steer the vehicle.

As stated before, the Frame Subject is responsible for fetching frames from the camera. These frames are unprocessed and in RGB format (see figure 2). Before extracting the lane lines, and subsequently, an error value, we need to send the frame from the Frame Subject through a series of filters.

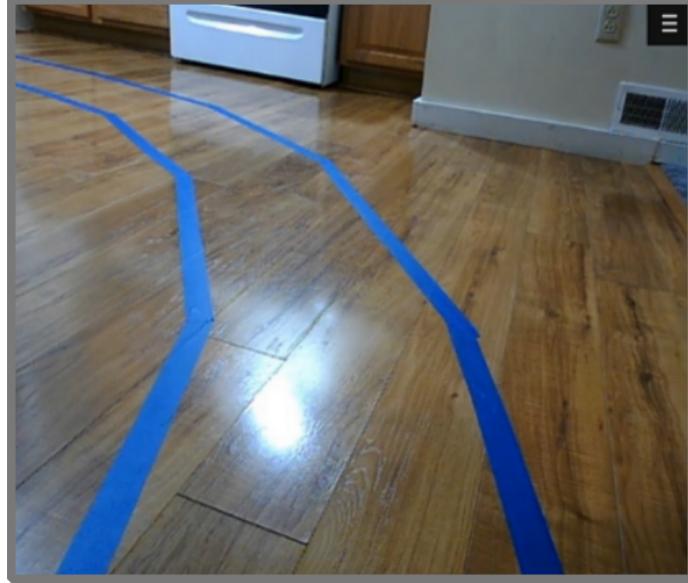


Fig. 2. The raw frame we pull from the Frame Subject.

## 2 BLACK AND WHITE LANE MASK FROM RAW FRAME

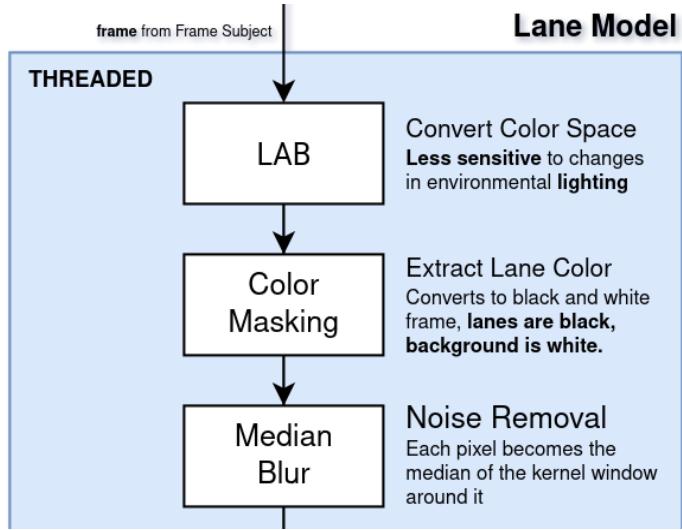


Fig. 3. Block diagram of the 3 steps we used to get a masked frame from the raw frame.

The incoming frame is in RGB format. The RGB channels are highly sensitive to changes in light, meaning that slight changes in environmental lighting, will change our model's ability to detect lane lines. As an antidote, we convert the frame to the LAB color space. The LAB color space separates the luminosity of the frame into its own channel, so the other two channels, which represent color, are less susceptible to changes in light. This allows us to tune our model to look for a specific color, that is mostly independent from lighting conditions.

We are using blue painter's tape as lane lines. So, we pass the LAB frame into a masking operation which creates a black and white frame, where the lane lines are masked out as black, and the background, is white.

The masked frame typically has noise. We use a median blur to reduce the noise. Median blur slides a small window across the frame and for each pixel, replaces it with the median of the pixel intensities for the window.

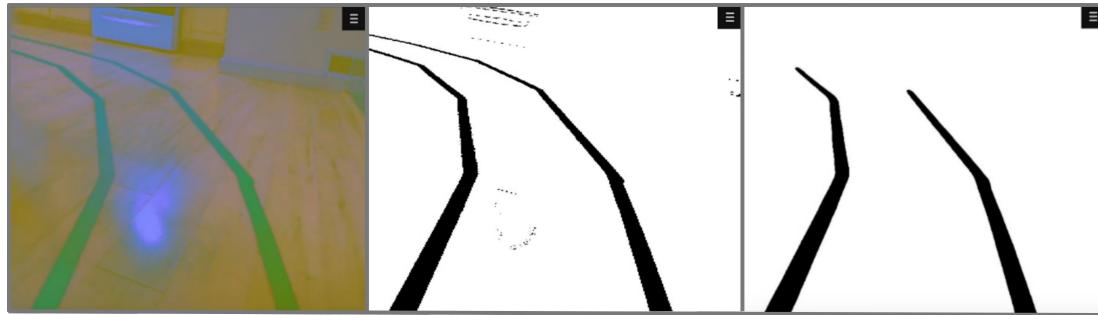


Fig. 4. Illustrates 3 steps to get a black and white mask of the lane lines, including, (left) converting the RGB frame to LAB, (middle) masking for the color of the lane lines, and (right) applying a median blur to reduce noise.

### 3 LINE SEGMENT COORDINATES FROM LANE MASK

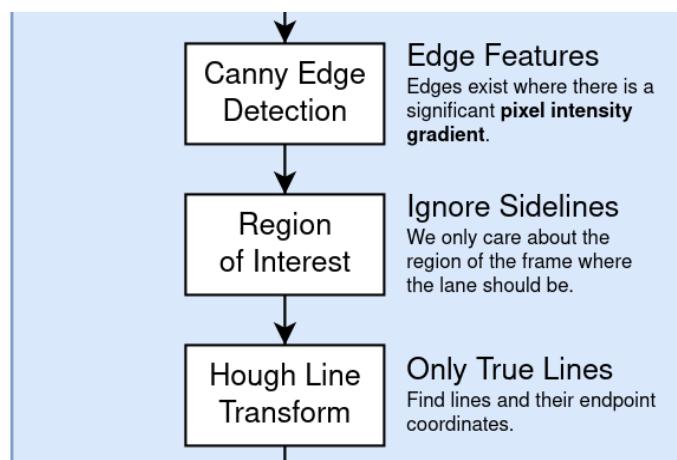


Fig. 5. Canny Edge Detection and Hough Line Transform are used to get lane line coordinates from the masked frame.

Canny Edge detection is done next, yielding edge features by looking for significant gradients in pixel intensity. Additionally, we know where in the frame the lane should be, so we simply ignore everything outside of the region that should contain the lane. This reduces the number of edge features which are not part of the lane, and parts of the lane which are not important yet.

Furthermore, we get line segments using the Hough Line transform, which returns the endpoint coordinates of edges that are specifically straight lines.

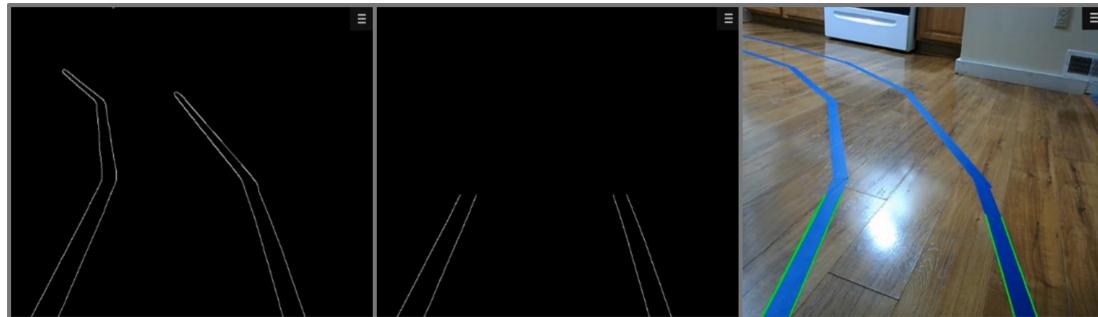


Fig. 6. Illustrates 3 steps to get lane line coordinates, including, (left) finding edge features with Canny Edge Detection, (middle) Applying a region-of-interest, and (right) using the Hough Line Transform to find straight line coordinates from the remaining edge features.

## 4 LANE ANGLE ERROR FROM LINE SEGMENT COORDINATES

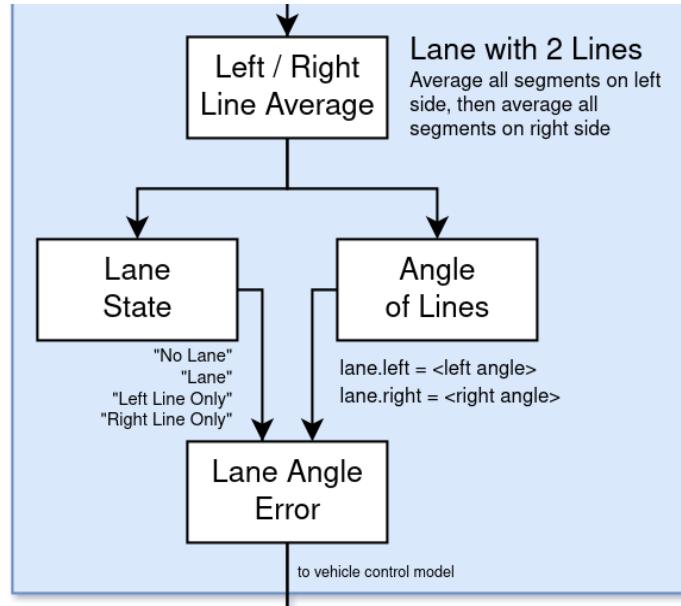


Fig. 7. The lane lines are average and used to detect the state of the lines (both, left, right or none) and the angle of the line, which we use to approximate lane error.

We need to get only two lines that represent the lane. So, we separate all segments with bottom intercepts on the left side of the frame from those on the right side of the frame. Then, we average the two groups of segments, to get the left and right lane lines.

When two lines are detected, the error between the vehicle's trajectory and the lane, we currently define as the difference between the angle of the bottom intercept of the left and right lines from the center of the frame. In our case, a positive error means the vehicle needs to steer left, and a negative error means the vehicle needs to steer right. If only one line is detected, we use a state machine which estimates the error between the line and a user-specified calibration angle. If no lines are detected the error does not change. The final error is illustrated in figure 1.