*Region Prediction Lane Detection Algorithm*

Yunfei Guo  
*Electrical and Computer Engineering   
Virginia Tech*Blacksburg, United State  
yunfei96@vt.edu

Xingyu Lu  
*Computer Science*  
*Virginia Tech*Blacksburg, United State  
toonoumi@vt.edu

Ryan Gerdes  
*Electrical and Computer Engineering   
Virginia Tech*Blacksburg, United State  
rgerdes@vt.edu

*Abstract - This paper describes the Region Prediction Lane Detection Algorithm. The algorithm is a vision-based lane detection algorithm implemented using C++ with OpenCV API. This algorithm can be used in any autonomous driving related research. The idea of this algorithm is using the lane detected in previous frame to highlight a region of interest and using the region of interest to predict where the lane might be in future frames. Compare with detecting lane in the entire image, using region of interest will dramatically decrease noise and increase processing speed.*

# Introduction

The two biggest challenge for lane detections are processing speed and noise. To solve these problems,  area prediction method is introduced.

## Predicting Lane

The prediction method is to use the lane detected in a previous frames to highlight a region of interest. The region of interest is the prediction of next frame, new Lane will be detected within region of interest. The region of interest is relatively small area in the original image, to process a smaller area will reduce noise and increase processing speed. The concept of region of interest is shown in figure 1.

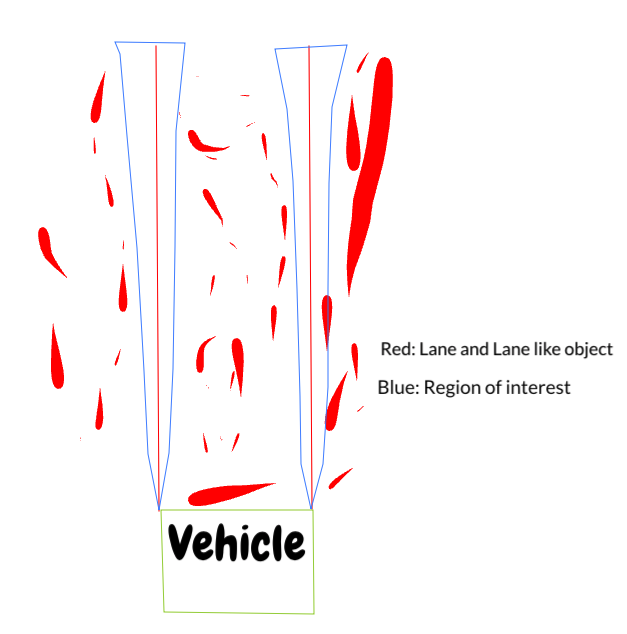


Fig. 1. Region of interest in prediction method

# Lane detection algorithm design

There are 4 parts in this lane detection algorithm. The first part is the predicting method used to predict where the new lane is. The second part is a multi-layer filter which filter out lane-like objects. The third part is fit a lane using the result of the filter. The fourth part is calculating region of interest.

## Predicting lane

For the first frame, the program use a very strict filter to filter out any lane-like object, fit a lane and calculate the region of interest. For the second frame and all consecutive frames, the program use a very loose filter to filter out any lane-like objects, fit a lane within the region of interest(ROI) and calculate the ROI for next frame. The ROI will be calculated by applying bitwise OR with ROI from past 5 frames. The calculated ROI will be used to predict the appearance of new lane. The flowchart of lane prediction method is shown in figure 2.

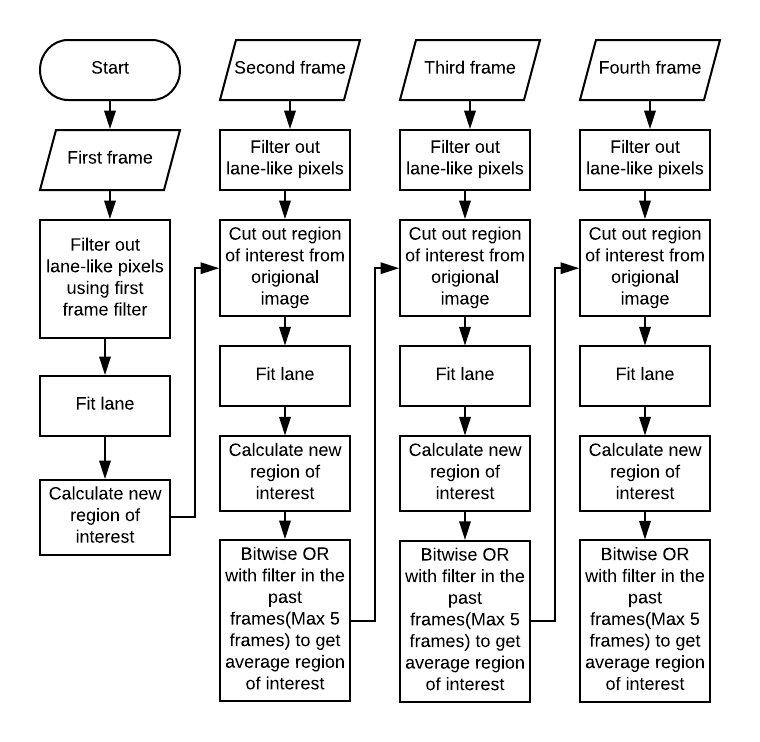


Fig.2. Flowchart for lane prediction method

## Filter design

There are two different filter designs. Each filter has a color-based detection layer and an edge-based detection layer.

Filter A is used for first frame detection, the result is filter out by applying bitwise AND between color and edge filter. It is a very strict filter that will eliminate as much noise as it can. But it will also lose some valuable information.

Filter B is used for detection in normal frames. The only difference is the result is filter out by applying bitwise OR between color and edge filter. It is a very lose filter that will filter out as much information as it can. But it will also filter out lots of noise.

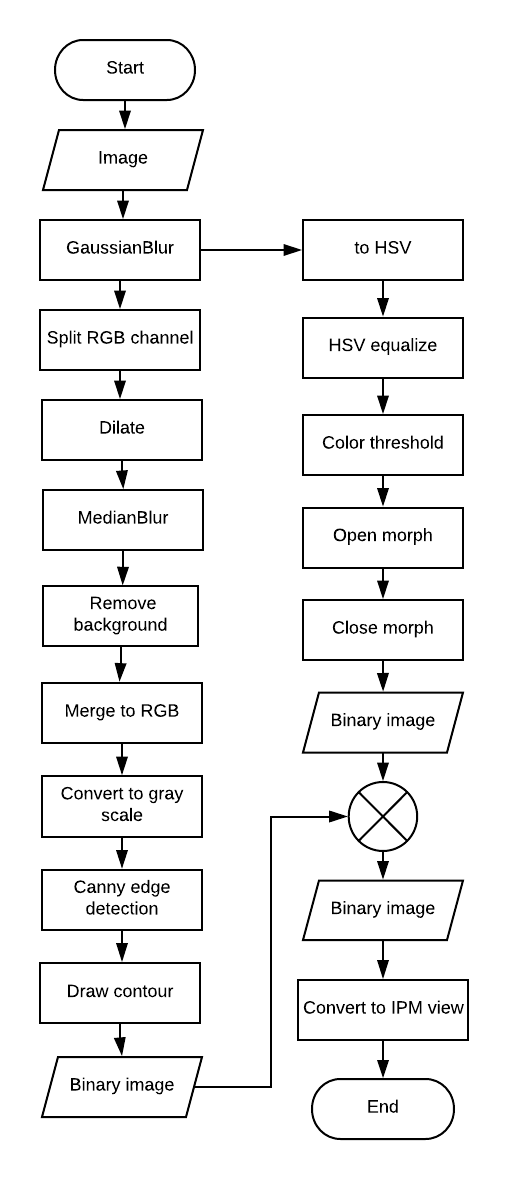
Before entering color-based or edge-based filter, there is a pre-process layer for the filter. In this layer, input image will go through Gaussian filter. It will blur the image to remove unnecessary texture. For example, texture of ground and grass.

In color-based filter, the blurred image is convert to HSV and equalized on all channel to eliminate the influence of light intensity. The equalized HSV image will go through a color threshold to get lane like objects. Open morphology and close morphology will be applied to result from previous layer to enhance the quality of detection. Open morphology removes small holes in the foreground and close morphology remove small holes in the background.

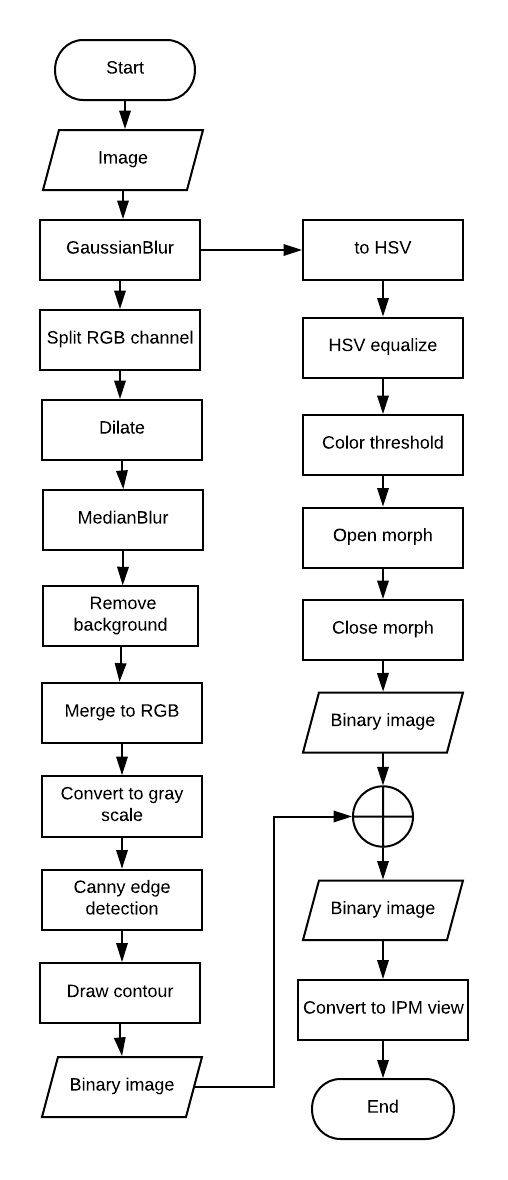
In edge-based filter, the first step is shadow removal. The blurred image is split to RGB channels. For each channel, the image will pass Dilate filter and Median Blur filter. Dilate filter will extract background and the Median Blur filter will blur the background to increase accuracy of background. The background will be subtracted from original image to remove shadow. The result will go through canny edge detector and contour will be drew on the image.

The result from color-based filter and edge-based filter will be combined using bitwise AND/OR to create a combined binary image. The image will be transformed into Inverse Perspective Mapping(IPM) view for easy lane fit and curvature calculation.

The flowchart of filter A and filter B are shown in figure 4 (a) and figure 4 (b).



(a)



(b)

Fig.4. (a) Filter A design flowchart. (b) Filter B design flowchart.

## Lane fitting

The binary image produced from filters are used to fit a lane. The demonstration of lane fitting is showed in figure 5. The region enclosed by red line are region of interest, new lane will be fit within the region. Black spots are all valid lane-like pixels. Lane-like pixels are result from the filter discussed in previous section. The medium point of lane-like pixels in each row is highlight in light blue. The region of interest are split into 8 sub-regions by 9 green lines. At further distance, there will be less valuable information, so the density of the line will decrease. When light blue lane-like pixel and the green lines intersect, Critical Points are formed. Connect these 8 Critical Points with dark blue lane give the result of lane fitting.

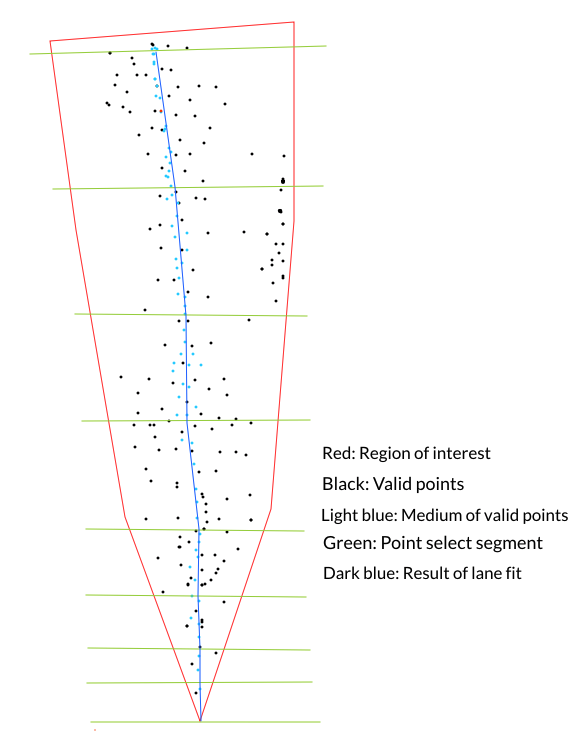


Fig.5. Demonstration of lane fitting.

## Calculating region of interest

If the detected lane touches the top and bottom, the region of interest will be calculated using existing 8 Critical Points. Each Critical Point has x and y coordinate based on the position in the image. By adding and subtracting certain value on x value of each critical points, 2 groups of Spread Lane Critical Points are formed. Connecting these Spread Lane Critical Points will form 2 Spread Lanes. The area enclosed by these 2 Spread Lanes and the boundaries of the image is region of interest.

If the detected is not touching the top or bottom of the image. ROI need to be extent towards both direction to produce better prediction result in next frame.

When detected lane is not touching the top of the image, the lane will be extended to the 80 pixels following the gradient of 2 Critical Points (p5 and p9). The coordinate of the Extend Critical Points following formula is used: , . This Extend Critical Point will be added to Critical Points. Extend Spread Lane will be calculated based on the new Critical Points. Connect the top of 2 Spread Lanes. The ROI will be area enclosed by these 2 Spread Lanes.

When detected lane is not touching the bottom of the image, the lane will be extended to the bottom following the gradient of 2 Critical Points (p0 and p2). The coordinate of the Extend Critical Points following formula is used: , . This Extend Critical Point will replace the lowest Critical Points. ROI will be calculated based on the new Critical Points.

# Testing environment and result

The algorithm are tested using videos having resolution at 1280x720. There are 2 different test environments. The first environment is real road environment. A camera is mount on a vehicle running on a real road with traffic coming on both sides. Multiple lighting condition are included in the test. The second environment is robotic platform environment. A camera is mounted on a tank drive robot and the robot are running in a parking lot. Multiple lighting condition are included in the test.

## Real Road

Video for this test is taken on a sunny day at 3:00 pm. This test use video from a camera mount on a vehicle running on road around a mountain. Different lighting condition are engaged. The vehicle start in shade of tree, after a while, the road is half cover by the shade. At the end of the video, there is no shade on the road. The algorithm works well under all conditions. Figure 6 (a) is a demonstration for real road testing.

## Sunny

The video for this test is taken on a sunny day at 12:30pm. This test use video taken by a camera mount on a robot to verify the performance under intense lighting condition. The algorithm works well under intense lighting conditions. Figure 6 (b) is a demonstration for intense lighting condition.

## Cloudy

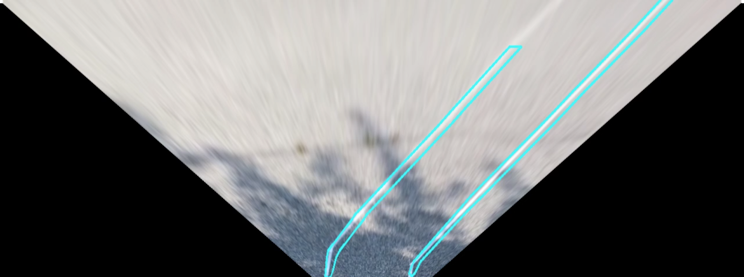
This video is taken at on cloudy day at 9 am. This test use video taken by a camera mount on a robot to verify the performance under *medium* lighting condition. The algorithm works well under medium lighting conditions. Figure 6 (c) is a demonstration for medium lighting condition.

## Low light

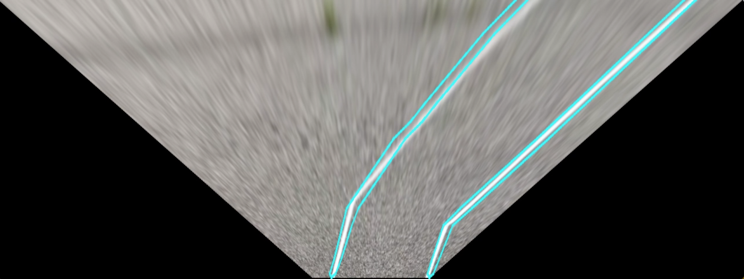
## This video is taken on a cloudy day at 8:30pm. This test use video taken by a camera mount on a robot to verify the performance under low lighting condition. The algorithm works well under low lighting conditions. Figure 6 (d) is a demonstration for low lighting condition.

## 

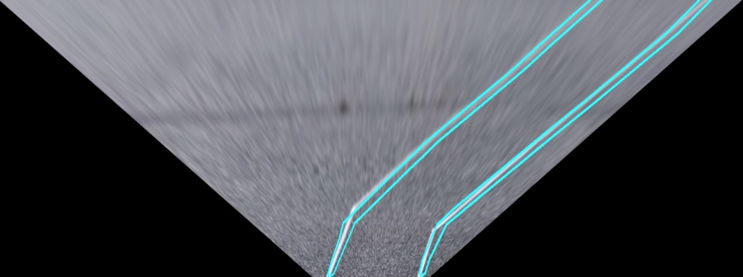
(a)



(b)



(c)



(d)

Fig.6. (a) Real Road test. (b) Sunny day test. (c) Cloudy day test (d) Low light test

##### Limitations

There are certain limitations for this algorithm.

The first frame detection is not perfect, especially under intense lighting condition. Some of the test videos need to be processed in order to detect lane on first frame.

When test in robotic platform, the shake of the robot will blur the video and decrease the accuracy of recognition.

When the robot/vehicle climbing up or down hills, the angle of view changed. The camera will facing up to the sky or down to the ground. When the proportion of lane and sky changes,  the algorithm will not work.

##### Conclusions

The algorithm works well when the region of interest is successfully selected. It also showed the ability to auto correct some wrong detection. It can handle various lighting conditions. The filter has a flexibility to add or remove any layer to improve performance.