Lane Line Detection for Vehicles

K. Sai Venkata Sri Supriya
Electronics and Communication Engineering
R.V.R & J.C. College of Engineering
Guntur, India

Md. Ayesha Begum
Electronics and Communication Engineering
R.V.R & J.C. College of Engineering
Guntur, India

M. Lakshma Naik
Electronics and Communication Engineering
R.V.R & J.C. College of Engineering
Guntur, India

K. Roopeshnadh
Electronics and Communication Engineering
R.V.R & J.C. College of Engineering
Guntur, India

***Abstract*— Lane line detection is critical section in modern vehicles to ensure driver safety and reduce accidents. The lane line detection for the vehicle camera should be highly accurate and fast to avoid any accidents. In order to meet such requirements, a detection algorithm based on the combined adaptive threshold and color threshold for accurate edge detection, sliding window and polynomial fitting for the interest model is proposed. This algorithm uses OpenCV in python and has high accuracy, high speed and good robustness. The proposed lane detection system can be applied on both curved and straight roads.**

Keywords*— *OpenCV, edge detection, color thresholding, lane line detection, region of interest, perspective transformation

I. INTRODUCTION

With the rapid development of society, automobiles have become one of the transportation tools for people. As number of vehicles are increasing day by day, the number of car accidents are increasing every year. Crossing the lane without following proper rules or careless driving are the root causes of most of the accidents. Lane discipline is crucial to road safety for drivers and pedestrians alike. So, Lane line identification for modern vehicles became important. One of the main technologies involved is computer vision which has been widely used in many applications. The lane line images are affected by light, wear, vehicle shade and tree shadow, it is still a big challenge to accurately detect the lane line. Therefore, a fast detection algorithm for lane line pixels based on combined adaptive threshold and color threshold is proposed. First, the algorithm uses adaptive threshold which detects edges even in uneven lighting conditions. The second basis is the color characteristics of the lane line. Suitable color space model is selected. Then adaptive threshold and color threshold are combined to detect a stable lane line. Region of interest is selected and perspective transformation is applied and then sliding window is used to determine the lane line pixels and polynomial fitting is done. Once the polynomial is fitted, curvature of the lane and positional information of the vehicle from the center are found. Then the detected lane boundaries are unwarped back onto the original image.

II. PRE TREATMENT

A. *Camera Calibration*

Images taken by camera are prone to distortions due to the nature of photographic lenses. In order to correctly detect the lane lines in the image, we first need to correct these distortions. There are two main types of camera distortions, radial distortion and tangential distortion. Radial distortion is caused by the fact that when light passes through the camera lens, the light at the edge will be bended more or less, so there will be distortion in the imaging of objects at the edge. Tangential distortion is mainly caused by the fact that the lens is not parallel to the imaging film or sensor. Therefore, it is necessary to calibrate the vehicle camera before lane line detection.

Camera parameters are intrinsic, extrinsic, and distortion coefficients. Extrinsic parameters define the location and orientation of the camera with respect to the world frame. They are rotation and translation vectors. Intrinsic parameters allow a mapping between 3D camera coordinates into 2D pixel coordinates in the image frame. These parameters include focal length, optical center, also known as the principal point, and skew coefficient. These parameters can be found by using known points in the real world and their projections on the image plane. Camera matrix is combination of intrinsic and extrinsic parameters and is unique for each camera.

Generally, the distortion is corrected by 5 parameters which are called distortion coefficients, that is D=(k1,k2,k3,p1,p2). To get the distortion coefficients, a chess board is used because of its regular and high contrast pattern, by which the parameters of camera can be easily calculated. In general, it is necessary to take more pictures of the checkerboard diagrams at different view angles, find the corner coordinates of each checkerboard diagram and its corresponding undistorted corner coordinates, and calculate the parameters for correction.
k1, k2, k3 represents the radial distortion parameters; p1, p2 represents tangential distortion parameters; r represents the distance from the corrected coordinates to the center of the image. Camera parameters are obtained using OpenCV functions findchessboardcorners(),drawchessboardcorners()and giving these points to function calibratecamera(). It gives camera matrix and distortion coefficients.

B. *Undistortion*

Once we get the distortion coefficients, they are used to correct the distortion or undistort image by using OpenCV function undistort() instead of using the equations. The image to be undistorted is given to the undistort() function and we get undistorted image. In this way radial and tangential distortions are removed from the image captured.

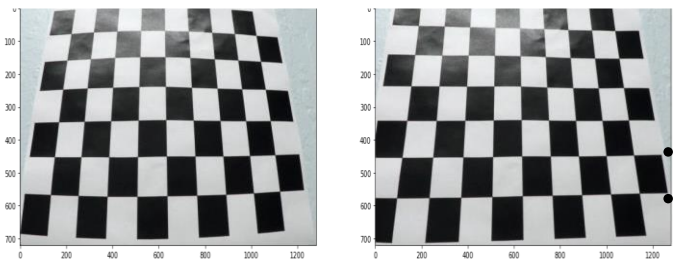

Fig. 1. Distorted and undistorted chess board images

III. IMAGE PROCESSING

A. *Edge Detection*

In Global thresholding, the thresholded value is global i.e., it is same for all the pixels of the image. So, a single threshold value is not suitable in the case of variable lighting conditions. To overcome this adaptive thresholding is used. Adaptive thresholding is the method where threshold value is found for different regions. So, edge detection is done using adaptive thresholding. In OpenCV, adaptive threshold operation on an image can be performed using the method cv2.adaptiveThreshold().We get binary image with edges highlighted.

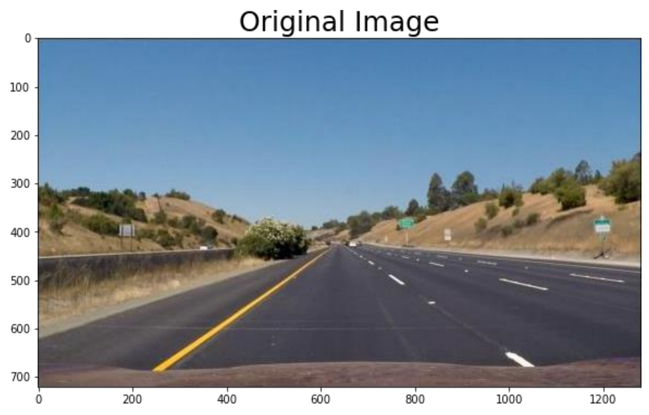

Fig. 2. Original image

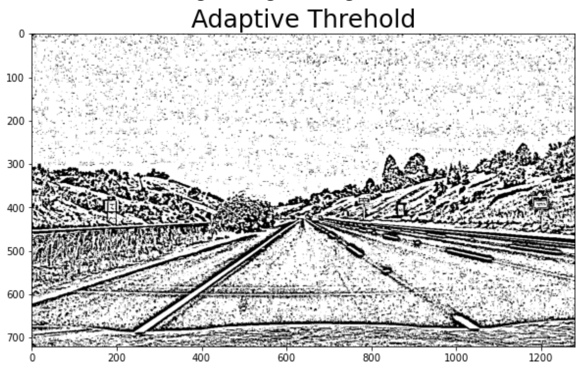

Fig.3. Adaptive thresholded image

B. *Color Threshold*

Color spaces are very useful tool to analyze images. There are various color spaces models that can be used to define the colors in an image like RGB, HLS or HSV, LAB etc. Different color spaces and color channels are tried and the best suited color space for the application to detect yellow and white lines is found to be HLS Color space. HLS means Hue, Saturation and luminance/brightness, which are particularly useful for identifying contrast in images. Hue is the different colors; Saturation is how intense the color is and value is the brightness value. OpenCV function is used for color thresholding. Once the correct color space is known, thresholding is applied to detect the edges.

For H- channel, Threshold is applied in the range of 10 to 40.

For L- channel, Threshold is applied in the range of 200 to 255.

For S- channel, Threshold is applied in the range of 100 to 255.

C. *Combined Thresholds*

Both adaptive thresholding and HLS color space thresholding are combined to get a stable lane line detection. The binary images from adaptive thresholding and color thresholding are combined using bitwise ‘and’ operator.

D. *Region of Interest*

The irrelevant information can interfere with the accurate extraction of lane lines. Therefore, it is necessary to filter out the interference information and extract the effective lane area of interest or region of interest. Region of interest (ROI) is the subset of original image. In extracting the region of Interest, first vertices of the required region of interest are determined. Then a function called cv2.fillPoly() is used. This function generally is used to draw filled polygons. Cv2.fillPoly() function takes three parameters as its input. They are image on which we want to draw the filled polygon, vertices of the polygon and color of the polygon. Once the polygon is drawn, bitwise 'and' operation is performed on the combined thresholded image and filled polygon image. To perform bitwise operation OpenCV has a function cv2.bitwise and(). After performing all the above operations, we only get the edges detected in the required region of interest (ROI) or effective lane area of interest.

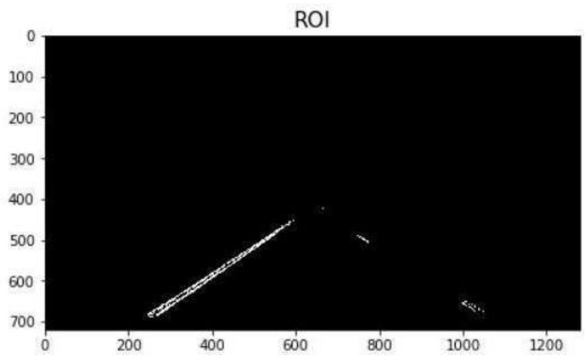

Fig.4. Region of Interest

E. *Perspective Transformation*

The In-perspective transformation, we change the perspective of a given image for getting better insights into the required information. In lane line images, the lane lines appear to be intersecting but they are actually not. So, we obtain thetop view of the lane line image to make the calculations easy.In perspective transformation, we need to provide the points on the image from which we want to gather information by changing perspective. We also need to provide the points to which we want to display our image. Then we get the perspective transform from the two given sets of points and wrap with the original image.cv2.getPerspectiveTransform() and then cv2.warpPerspective() are used from OpenCV. Cv2.getPerspectiveTransform() takes the source and destination points as inputs. Source points are the coordinates of vertices in the source image and destination points are the corresponding vertices in the destination image. cv2.getPerspectiveTransform() was used which gives transition matrix as its output. Then cv2.warpPerspective() was used which takes transition matrix and image shape as inputs and gives the transformed image.

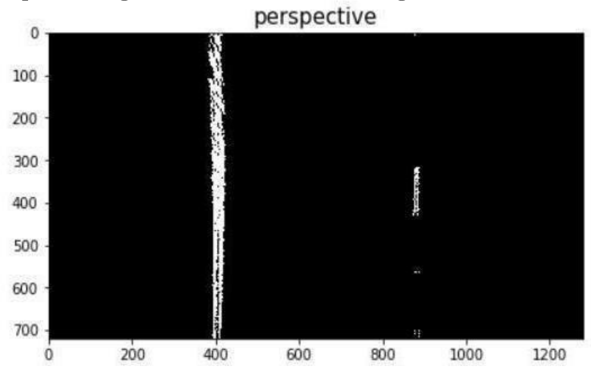

Fig.5. Perspective Transform

F. Sliding window and Polynomial Fitting

After applying all the above steps to a road image, we get a binary image where the lane lines are clearly seen. However, we still need to decide which pixels are part of the lines on the road and also which belong to the left line and which belong to the right line. Plotting a histogram of the image is one solution for this. Histogram is taken along all the columns in the lower half of the image. The two peaks in this histogram can be used as starting points for lane lines. Left peak tells left lane starting point and right peak tells right lane starting point. From that points, we use a sliding window to detect complete lines.

Sliding window Algorithm:

First non-zero pixels are found in the entire image image. Next, a sliding window is defined and non zero pixels inside the window are found.

Once the nonzero pixels exceed the given range, the window is taken centered at mean of nonzero pixels count. In this way sliding window is moved in top direction to find lane pixels.

Once we get all the pixel points of lanes, a second-degree polynomial curve is used to fit across them. NumPy polyfit function is used to create polynomial fit .The polynomial equation of second degree polynomial is: f(y)=Ay^2+By+C where A, B and C are the coefficients. The steps are repeated for left and right lane line separately.

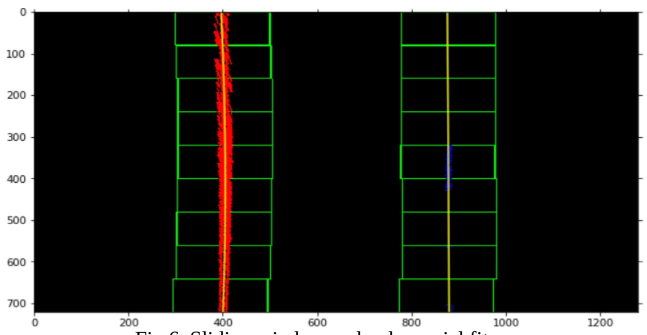

Fig.6. Sliding window and polynomial fit

G. Measure Curvature and Position

Once the polynomial is fitted through the lane lines, radius of curvature is calculated . To find the distance from the center, The average of the left and right lane is taken and then subtracted from the center. This distance is next multiplied by the xm_per_pix to convert it into meters. Finally, Cv2.putText() helps to put the radius of curvature and position of the vehicle along with direction on the image.

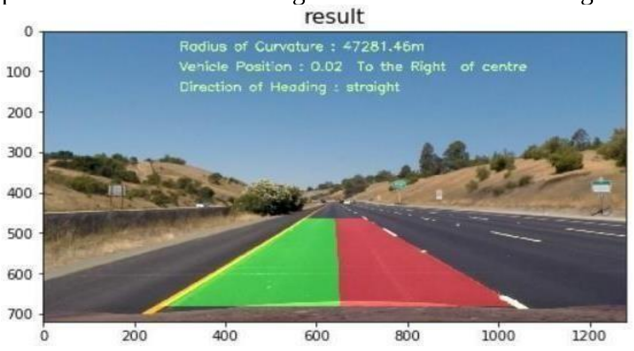
result
Radius of Curvature : 47281.46m
Vehicle Position : 0.02 To the Right of centre
Direction of Heading : straight

Fig.7. Resultant image

IV. CONCLUSION

In this paper, combined edge method and color filter are used to detect the lane line. This project is entirely based on image processing and entire implementation is done using specific algorithm. This test method effectively avoids the light, wear, car shade and trees shadow effects on the lane line image. Amount of calculation and robustness of algorithm are highly optimized, this method can be also applied to the safety assistant driving system, or autonomous vehicle system.

REFERENCES

[1] Sharath Srini "Lane detection using sobel operator and sliding window," IEEE Intelligent Vehicles Symposium, 2005

[2] MA Chao, XIE Mei. A method for lane detection based on color clustering[C]//Third International Conference on Knowledge Discovery and Data Mining. Phutet: IEEE [7] Computer Society, 2010: 200-203. K. Elissa, "Title of paper if known," unpublished.

[3] Yangzhe Wei proposed 'Detection of lane line based on Robert Operator' [C]//2017 IEEE3rd Information Technology and Mechatronics Engineering Conference (ITOEC2017),2017

[4] Want, J.S. and Knipling, R.R. Single-Vehicle Roadway Departure Cranshes: Problem Size Assessment and Statistical Description. National Highway Traffic Safety Administration Technical Report DTNH-22-91-C-03121.

[5] PENG Bo, CAI Xiao-yu, ZHANG You-jie, et al. Automatic detection of UAV video vehicles based on symmetric frame difference and block background modeling[J]. Journal of Southeast University(Natural Science Edition),2017,47(4):685-690.

[6] Y. Kutsuma, H. Yaguchi, and T. Hamamoto, "Design and implementation of smart image sensor for real-time lane detection", ITETechnical Report, vol.28, no.25, pp. 17-20(2004)