A Simple and Efficient Lane Detection using Clustering and Weighted Regression

Rudra N. Hota, Shahanaz Syed, Subhadip Bandyopadhyay, P. Radhakrishna

SET Labs, Infosys Technologies Ltd.
Hyderabad, 500032, India.
{rudranarayan_hota, shahanaz_syed, subhadip_b and radhakrishna_p}@infosys.com

Abstract

Developing a vision based, efficient and automatic lane detection system from a moving vehicle is a challenging task mainly due to poor quality of lane markings, occlusion created by traffic, complex road geometry and nevertheless, the non-existence of unique models. The focus of this paper is to detect line or curve like segments from a video image taken from moving vehicle and merge them to detect road lane marks. In this work, we propose a method of applying weighted regression to fit a curve in place of traditional line grouping and fitting curve separately. The proposed vision based solution is simple but computationally efficient and hence works in real time on the image sequence captured by camera mounted in the vehicle.

1 Introduction

The application of Computer Vision systems has boosted the advances in automation systems in various domains such as transport, surveillance, medical and retail. For example, lane mark detection, on-road obstacle detection, traffic monitoring system and their integrations for automatic vehicle guidance and safety systems are some of the demanding applications in the field of transport. In the recent years most of the on-road accidents and mishaps are caused due to speeding and rash driving. As the traffic ahead is not known, the chance of accident is high. With the new generation of large number of heavy vehicles on the roads and most drivers changing lanes as often as possible, thence the rise of accident rate. Thus, there is a need of driver aid system to perform emergency maneuvers or alarm generation to increase the safety of automobiles. To avoid a vehicles straying out of lane, interest in lane departure warning and tracking systems has increased a lot among the research community as well as the product groups. These systems also have widespread applications in intelligent transport systems and robotics [1]. Researchers have made some attempts in vision based lane detection and road boundary detection to warn or assist in vehicle guidance. However, it is clear that such systems are mainly affected by shadows, bad weather such as fog, rain, multiple source of lights, instability in camera view and by varied illumination (e.g change of light source from broad day to evening and late night). As the field of view of the camera has to deal with road perspective, the points that are close to the top of the image has a bigger uncertainty than the bottom ones. In many cases the uneven curvature at the far end makes it difficult to estimate lane marks properly and line detection and their merging works poorly under such cases.

Although systems with use of sensors or special markers can be embedded in the road to automatically guide vehicles, such systems are costlier as investment in road infrastructure and modification of automobiles is high. By using vision technology to detect lane markings on existing road infrastructure, it is possible to produce a viable autonomous vehicle or driver aid system. Computer vision based approach becomes particularly important in traffic applications mainly due to their fast response, easy installation, operation and maintenance. This kind of algorithms mainly has two sub problems. First one is the detection of the lane mark in static frames and second one is tracking them in video sequence. In this work, our focus is on lane mark detection.

The literature on automatic lane detection can be categorized into lane-region detection, model driven and feature driven approaches [6]. In lane-region detection, representation by color and texture features for road and/or non-road segments by using stochastic pattern recognition techniques followed by segmentation are popular. In [4] and [2], authors have tried to apply color segmentation and threshold on enhanced color channels to detect lane marks. However, color consistency is still very challenging due to variation in illuminations and road visibility. The aim in model-driven approaches is to match a deformable template (snakes and splines) through scene characteristic to

15^{th} International Conference on Management of Data
COMAD 2009, Mysore, India, December 9–12, 2009

model road segments. Model driven approaches provides good results in lane detection application but they require heavy computation. Feature driven approaches (local feature extraction, supervised methods for classification of lane marks [7]) faces many challenges such as noise effect, blur and irrelevant feature structure but they do have flexibility of local and global feature representation for near and far end curved lane marks and hence seems promising in lane mark analysis and tracking.

In the view of above challenges and industrial demand, we developed a feature driven methodology for real-time lane detection for automotive vehicles. The block diagram in figure 1 explains the system overview of the method for Lane detection. The basic input is a video file taken from camera mounted on a moving vehicle. The flow of the method is depicted in this diagram. The major components in this approach includes video capture and frame extraction, line/curve like mark detection, spurious segment removal, line merging and fitting curves. A brief description of these components is given below:

- **Input frame extraction:**From the video output of color video camera, the sequence of frames has been extracted and fed to the system for further processing. Here we divide the image frame into two parts: near view and farther view. In near view the lane marks are normally line like structure and in farther view the marks are mostly curve like. So we divide the image into two parts and process them separately.

- **Line detection:** In general, lane marks are line and curve like structures. The disconnected lane marks are identified with line detection algorithm. Though, line detection method is able to detect small line like segments in the lane curve, they are not always continuous in space and slope. Hence we further process these line segments and merge them for continuous lane mark detection.

- **Spurious line removal and line merging:** The spurious lines are removed and merging of lines is done by grouping them according to their similarity in slope and spatial closeness. The spurious lines are removed, based on hypotheses as follows: (i) The lane markings should be somewhat parallel to each other and they should merge at vanishing Point. (ii) Starting point of the left and right lane mark should be in 3rd and 4th quadrant respectively for a given image divided into four quadrants. (iii) The slope of the line in the near sight should be approximately vertical and continuous in the latter part.

For effective detection, we use image enhancement followed by gradient operator for edge detection as the first step. In the second step we use probabilistic Hough transform for line like structure detection. Our unique contribution in this work is to merge various line segments by using agglomerative hierarchical clustering and apply weighted regression via least square fit for fitting these segments into a lane curve. The rest of the paper is organized as follows: in section 2 we briefly describe the literature work and present issues. In section 3, we present our approach and features of lane detection, followed by observation and results in section 4. We conclude in section 5.

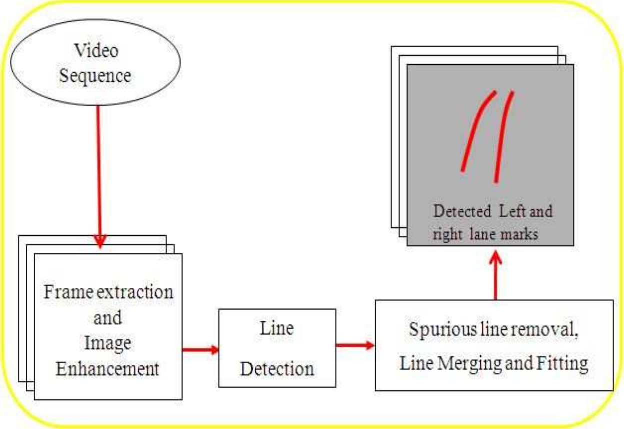

Video Sequence
Detected Left and right lane marks
Frame extraction and Image Enhancement
Line Detection
Spurious line removal, Line Merging and Fitting

Figure 1: Block diagram of Lane detection

2 Related work

In literature several methods have been presented for lane detection problem. Here we describe works related to feature (e.g., edge, gradient) based approaches for automatic lane detection and their issues.

The practical issues in gradient/edge based automatic lane detection includes the following:

- Low detection rate due to low visibility.
- Steep curve in near and far end of image frame.
- Low detection due to occlusion (in case of heavy traffic) and fast change in Field of View (FOV).
- Disconnected lane marks (some times colored painting and arrow marks).
- Missing lane mark at initial stage of the frame creates problem for tracker based lane detection method.

Moreover, varied color mark, uneven structure at far end perspective camera view, simple pixel features, color value or road texture are not prominent for lane mark candidate features. Local gradient and edge component are more appropriate for lane mark hypothesis generation. A computationally faster algorithm for line like structure detection by gradient information extraction and randomized Hough transform is proposed in [1]. It performs well when the lanes are straight, but not quite suitable for detecting curving lanes. Other such similar feature like canny edge, steerable filters are attempted by [9] for lane mark estimation and tracking. In [7], authors have tried simple gray and color pixel value with supervised classifiers (ANN, SVM, Naive Bayes). In these works, the detected line segments are assembled into lane structures using merging and grouping them according to their spatial position and orientation. Least square fit [3], RANdom SAmple Consensus(RANSAC), Hyper parabola [12] or B-spline [11] fitting is the next step to estimate the probable position of aggregated lane in image frame. Inverse Perspective Mapping (IPM) along with particle filtering (for detecting lane structure with highest voting) is one of well suited techniques used for lane detection [10]. This technique failed to detect lane, when there are disconnected lines at the initial stage of the image. A robust lane-detection-and-tracking algorithm is reported to have good performance in dealing with challenging scenarios such as lane curvature, lane changes and splitting lanes in [7]. In this method a major case of miss detection is due to fast lane changes.

3 Lane Mark Detection

In this section, we describe our proposed method of detecting lanes, by extracting small line like structures of the lane using Sobel gradient operator and canny edge detection, probabilistic Hough Transform for line segment detection, and then clustering the line segments. We used clustering method to group the line segments with their slope and Y-intercept in two different stages: (i) clustering with slopes of line segments as input and (ii) clustering again with Y-intercepts of the grouped lines from previous step. This is mainly to get distinct distance between line segments with respect to their angle and positions separately, as they are values of different scales (which may hamper the distance computation for proper clustering). Secondly, this two stage separation helps us to compute weight according to Y-intercepts for each lines in their group and apply heteroscedastic regression to fit lane marks (line or curve fitting).

Most of the works in the literature, merging of the line segments is done by simple threshold or clustering on similar slope and position followed by curve fitting. In this work, we combine the line grouping and curve fitting in a single step in order to improve computational efficiency. This is carried out by the use of weighted least square regression. Figure 2 shows the flow of the proposed solution. The dotted line in this diagram shows the two stage clustering process, while the solid line shows the proposed method of weighted regression based on Y-intercepts.

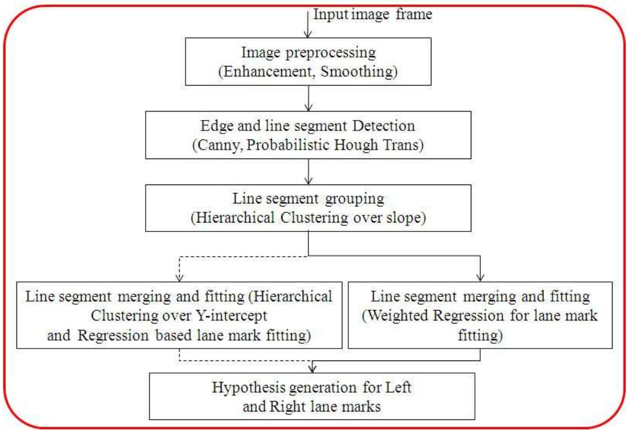

Input image frame
Image preprocessing (Enhancement, Smoothing)
Edge and line segment Detection (Canny, Probabilistic Hough Trans)
Line segment grouping (Hierarchical Clustering over slope)
Line segment merging and fitting (Hierarchical Clustering over Y-intercept and Regression based lane mark fitting)
Line segment merging and fitting (Weighted Regression for lane mark fitting)
Hypothesis generation for Left and Right lane marks

Figure 2: Flow diagram of Lane detection

The stepwise output for an image frame is shown in figure 3. The input image is smoothed by Gaussian smoothening method followed by gradient computation and edge detection. We applied probabilistic Hough transform on the detected edges to identify the line segments. In standard Hough transform the lines are detected by putting the edge points into similar rho (ρ) and theta (θ) (parameter of the line equation) values. The detected line segments are shown in red lines in the second block. We applied Probabilistic Hough transform to achieve computational advantages as compared to standard Hough transform with negligible performance degradation [8]. As the name suggests, probabilistic Hough transform detect lines from few of the randomly selected edge points, thus helps in faster computation than standard Hough transform technique.

The next two images (image (iii) and (iv) in figure 3) are showing outputs from the clustering techniques used to group the lines in different sets. The computed slope and Y-intercept from each of the line segements are used for clustering them by agglomerative hierarchical clustering [5]. This method does not require the number of clusters as an input, but needs a termination condition. In line segment grouping for lane detection the number of group of lines is not known aprior, but we can put threshold in slope and Y-intercept to group them, hence we considered agglomerative hier-

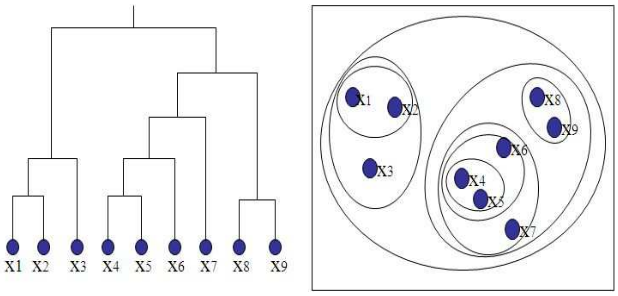
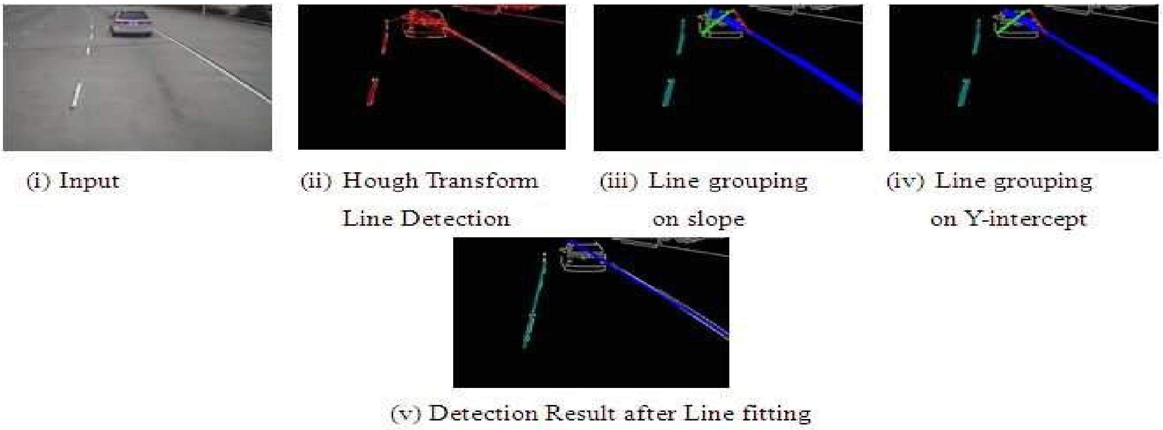


Figure 4: Dendrogram and Venn diagram representation of Agglomerative Hierarchical Clustering


(i) Input

(ii) Hough Transform Line Detection

(iii) Line grouping on slope

(iv) Line grouping on Y-intercept

(v) Detection Result after Line fitting

Figure 3: Step wise output of Lane detection

archical clustering approach. A brief algorithmic description of this method is given in table 1. Figure 4 shows the Dendrogram and graphical representation of the technique for nine samples. The process of clustering starts with many clusters where each sample represents one cluster and with different levels of clustering they get merged according to their closeness with other clusters. The closeness (or similarity) measure between any two clusters plays important role in resultant cluster formation. It can be observed from the Dendrogram and Venn diagram that how spatially nearer clusters are getting merged at different levels.

Table 1: Agglomerative Clustering Algorithm for line segment grouping

```
Input: A Set of samples $X = \{x_1, x_2, x_3, \ldots, x_n\}$ of size n representing line segments.
T-Threshold to stop merging of sub-clusters.
Output: Clusters with maximum inter class distance greater than T.
1. Start with n disjoint clusters (NoofClsuters=n), each one representing one cluster.
2. Compute the similarity measure for each pair of clusters $C_i$ and $C_j$ [ $d(C_i, C_j)$ as in equation 1].
3. Find the most similar pair of clusters $C_i$ and $C_j$, in current clustering and merge them into one cluster if their similarity is less than equal to T and consider them as one cluster for further processing.
4. Decrease NoofClusters by 1.
5. Repeat step 2, 3 and 4 until there is no two clusters with closeness less than T or reached to single cluster.
6. Return clustered samples
```

We used average linkage to measure the cluster similarity. So the similarity between the clusters $d(C_i, C_j)$ in the algorithm is computed as the average distance from all the member of one cluster to all member of other cluster as given below.

$$d(C_i, C_j) = mean_{(x_1,x_2)\epsilon C_1 \times C_2} d(x_1, x_2) \qquad (1)$$

where $d(x_1, x_2)$ is euclidian distance between x_1 and x_2

The third image in figure 3, shows the output of the grouping done on slope and fourth image is output of grouped lines on Y-intercept. The clustering method is applied on the computed slope and Y-intercept for each line. Different threshold (T) values (for slope and Y-intercept) in agglomerative clustering are used to stop the clustering process of merging intermediate clusters to get line groups. Each group of lines after clustering is shown in different colors to demonstrate the effectiveness of the approach in line grouping for lane detection. The clustering on slope helps to find the set of line segments with similar angles with respect to x-axis. But clustering on Y-intercept ensures to discard the outside lines (e.g., lines parallel to the set, but far away from the group) which may be parallel to the group. One such case is shown in figure 5, where the closely placed lines are grouped and shown in green color.

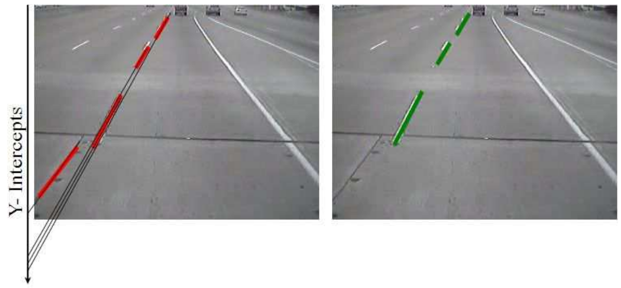


Figure 5: Clustered output based on Y-intercepts of line segments

The output of clustering gives sets of lines which are having similar slope and Y-intercepts. In practical case of lane detection these line segments are disconnected in nature. To get the actual lane marks these grouped lines need to be combined to one continuous mark. We used weighted regression technique to fit these lines, and compared with least square regression to show the viability of the former approach. Below we briefly described these regression techniques:

3.1 Least square Regression

Consider a multiple regression setup, where the i_{th} sample observation on p independent variables is given by $X_i = (X_{i1}, X_{i2}, X_{i3}, ..., X_{ip})$ and one dependent variable Y_i. For a random sample set of size n $\{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ..., (X_n, Y_n)\}$, The matrix representation of linear regression is given by:

$$Y = X^*\beta + \epsilon \tag{2}$$

where $Y = \begin{bmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{bmatrix}$, $X^* = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix}$,

$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ ... \\ \beta_p \end{bmatrix}$ is the unknown parameter and ϵ is the random error.

The least square estimation of β under homoscedastic (equal variance of the error $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n)$) is

$$\hat{\beta} = (X^{*T}X^*)^{-1}(X^{*T}Y) \tag{3}$$

3.2 Weighted regression

One of the common assumptions in linear and nonlinear least squares regression is homoscedasticity. This assumption, however, clearly does not hold in many modeling application like line merging in lane mark detection. Lines farther away from the group has very less contribution to the group and hence it is not reasonable to assume that every observation should be treated equally. Weighted least square (for heteroscedastic regression) is the most appealing solution which is used to maximize the efficiency of parameter estimation. This is done by assigning each data point to its deserved degree of importance in the parameter estimation process. By including weight factors, the solution equation (2) changed to

$$\hat{\beta} = (X^{*T}\Omega X^*)^{-1}(X^{*T}\Omega Y) \tag{4}$$

where $\Omega = \begin{bmatrix} \omega_1 & 0 & 0 & \dots & 0 \\ 0 & \omega_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \omega_n \end{bmatrix}$ is a diagonal matrix consists of weighing factors (ω_i) for each of the sample point. In line merging for lane mark detection, points from each of the detected line segments are considered for regression based fitting and each of the points (samples) are weighted according to the line weighing scheme described in the next subsection.

3.3 Line segment Weighting strategy

To reduce the effect of spurious line segments which are far away from the group, each of the line is weighted with a weight factor. The weight factor is computed as per its distance from the other lines in the same set as given below.

$$\omega_i = \sum_{j=1}^{n} \exp^{-d(x_i, x_j)} \tag{5}$$

where ω_i weight for ith line segment in a group of n lines and $d(x_i, y_i)$ is the distance between i_{th} and j_{th} line.

4 Results and Observations

In this section, we discuss about the dataset used and compare the performance of lane detection by different grouping and regression method. We also describe the performance analysis in detecting right and left lane marks.

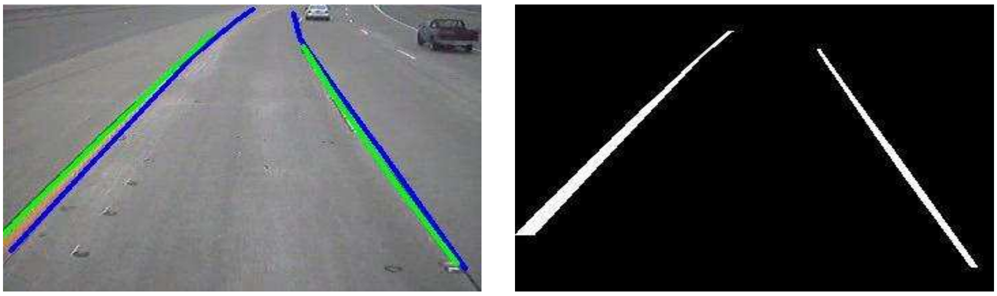
Datasets: We have conducted experiments for lane detection on publicly available datasets [7] and tested our approach on the selected 700 images from this video data. We extracted the image frames which are of size 240x352 pixels. We manually annotated each frame by marking left and right lane marks and used them to compare the performance obtained from the proposed approach. The strategy for comparing the annotation and detection marks is described below.

Performance Analysis: We considered proximity between annotated and detection marks for calculating the correct detections. The goodness of the detection is measured according to the normalized area between the detection and annotation lane marks (see figure 6). The green marks in figure 6(a) shows detections and blue lines represent annotations. The common region between annotation and detection of left and right lane marks are shown in figure 6(b). The correct detections are calculated by taking 30 (number of pixels in the common region) as the limit on this area.

The Detection Rate (DR) is computed as below:

$$DR = \frac{Number\ of\ correctly\ detected\ lanes}{Total\ number\ of\ lanes\ in\ video\ sequence} \tag{6}$$

Detection Rates have been used for analysis of the results for four different approaches used for lane mark detection. Figure 7 shows the pictorial representation of the detection rates for different approaches. The x-axis shows four different experiments and y-axis represents corresponding detection rates. From the figure



(a) Annotated and Detected Lanes (b) Region within the lane marks

Figure 6: Detection and annotation lane mark comparison

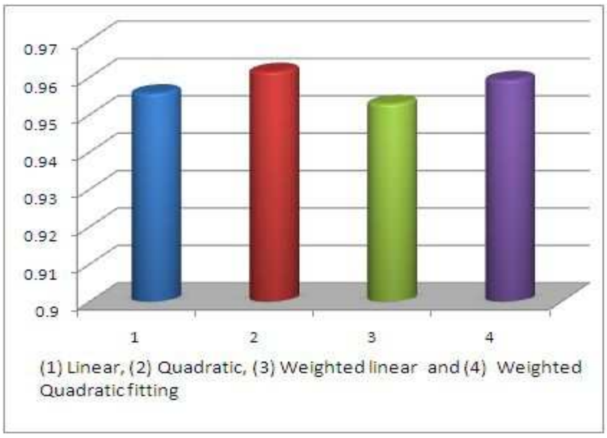


Figure 7: Lane marks Detection rate for different approaches

it is clearly observed that the detection rates for the Quadratic Regression is better than the linear regression both for clustering and weighted regression. Further, it is also observed similar performance for Lane mark detection using both weighted linear/quadratic regression and clustering followed by linear/quadratic regression.

Sl. no	Regression Method	Avg. frames per second (fps) processed
1	Linear	25.39
2	Quadratic	26.50
3	Weighted Linear	30.47
4	Weighted Quadratic	30.20

Table 2: Computational time comparison

The computational time for the weighted linear/quadratic regression is less when compared to linear/quadratic regression using clustering (Table 1). We achieved 30 fps (frames per seconds) processing by applying weighted regression.

Some of the selected frames with practical challenges along with detection results by the proposed approach are shown in figure 8. The input frames are shown in column 2. In 3rd and 4th column of this figure, the detection results of two different methods: Clustering followed by Regression and Weighted regression are presented. For frame 24 both the methods are performing quite well. Frame 48 shows a special case where the right lane mark is not continuous in the farther end. So there no supportive line segment for proper detection, still weighted regression is showing proper estimated lane mark. In next two frames two difficult cases namely occlusion and disconnected lane marks are shown. The lane marks in frame 550 are curved in nature and they are occluded by the vehicle in the front. In frame 302 the left lane marks are very small and disconnected also. In both the cases our proposed approach of weighted regression is performing better.

5 Conclusion

In this article, we described a computer vision based on-road Lane detection method. The efficacy of the method is shown on real data set in out-door scenarios. The proposed approach of weighted regression for lane mark fitting performed better than simple regression, both in terms of detection rate and computational efficiency. Our future work includes investigation and analysis of on road challenges in heavy traffic.

References

[1] S. Azali, T. Jason, H. Mohd, and H. Ahmad. Fast lane detection with randomized hough trasform. In *Proceeding of International Symposium on Information Technology 2008*, volume 4, pages 1–5, August 2008.

[2] J. Calderon, A. Obando, and D. Jaimes. Road detection algorithm for an autonomous ugv based on monocular vision. In *CERMA '07: Proceedings of the Electronics, Robotics and Automotive Mechanics Conference*, pages 253–259, Washington, DC, USA, 2007. IEEE Computer Society.

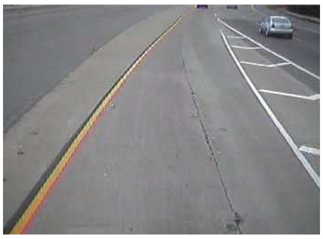
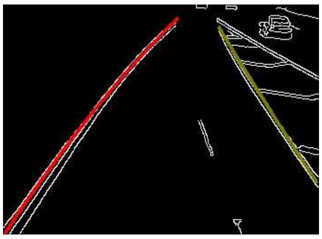
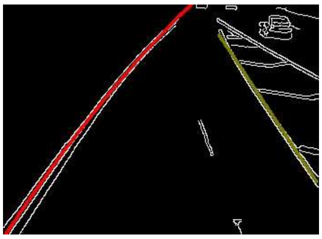
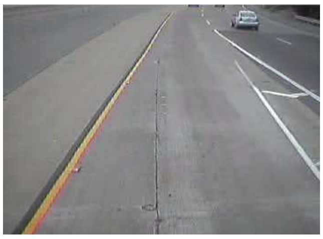
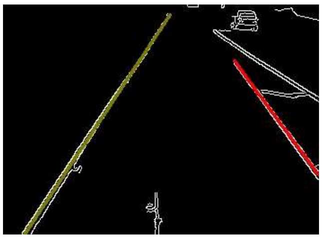
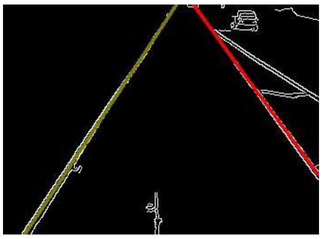
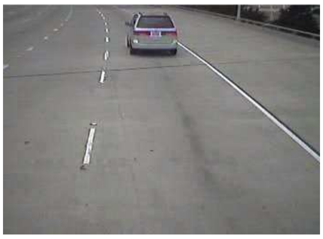
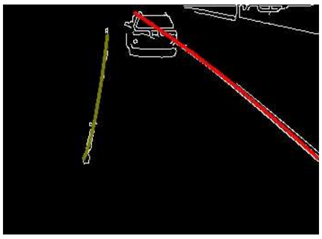
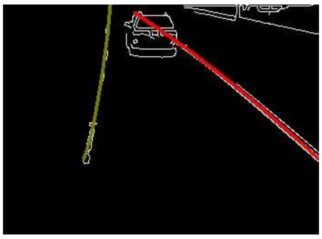
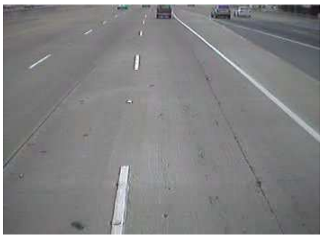
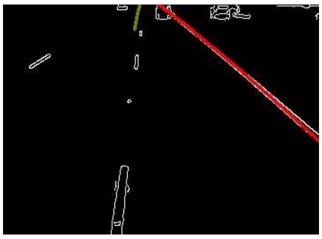
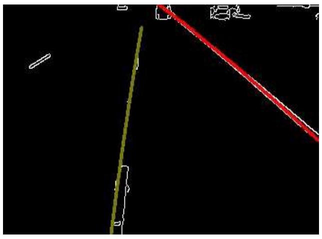
Frame No.	Input image	Clustering over Y-intercept and Regression	Weighted Regression
24			
48			
550			
302			

Figure 8: Detection Results

[3] Q. Chen and H. Wang. A real time lane detection algorithm based on a hyperbola-pair model. In *IEEE Intelligent Vehicles Symposium*, pages 510–515, 2006.

[4] K. Y. Chiu and S. F. Lin. Lane detection using color-based segmentation. *intelligent Vehicles Symposium*, (4):706–716, June 2005.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience Publication, 2000.

[6] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4):359–381, April 2003.

[7] Z. Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26, 2008.

[8] N. Kiryati, Y. Eldar, and A. Bruckstein. A probabilistic hough transform. 24(4):303–316, 1991.

[9] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006.

[10] S. Stephan, K. Sarath, A. Alen, and D. Gamini. Efficient lane detection and tracking in urban environments. In *3rd European conference on Mobile Robots*, September 2007.

[11] Y. Wang, E. Teoh, and D. Shen. Lane detection and tracking using b-snake. volume 22, pages 269–280, April 2004.

[12] W. Zhu, Q. Chen, and H. Wang. Lane detection in some complex conditions. In *IEEE International Conference on Intelligent Robots and Systems*, volume 4, pages 117–122, October 2006.