2012 IEEE International Symposium on Multimedia

Video-based Lane Detection using a Fast Vanishing Point Estimation Method

Burak Benligiray[1], Cihan Topal[1], Cuneyt Akinlar[2]
Department of Electrical and Electronics Engineering[1], Department of Computer Engineering[2]
Anadolu University
Eskisehir, Turkey
e-mail: {burakbenligiray, cihant, cakinlar}@anadolu.edu.tr

***Abstract*— Lane detection algorithms constitute a basis for intelligent vehicle systems such as lane tracking and involuntary lane departure detection. In this paper, we propose a simple and video-based lane detection algorithm that uses a fast vanishing point estimation method. The first step of the algorithm is to extract and validate the line segments from the image with a recently proposed line detection algorithm. In the next step, an angle based elimination of line segments is done according to the perspective characteristics of lane markings. This basic operation removes many line segments that belong to irrelevant details on the scene and greatly reduces the number of features to be processed afterwards. Remaining line segments are extrapolated and superimposed to detect the image location where majority of the linear edge features converge. The location found by this efficient operation is assumed to be the vanishing point. Subsequently, an orientation-based removal is done by eliminating the line segments whose extensions do not intersect the vanishing point. The final step is clustering the remaining line segments such that each cluster represents a lane marking or a boundary of the road (i.e. sidewalks, barriers or shoulders). The properties of the line segments that constitute the clusters are fused to represent each cluster with a single line. The nearest two clusters to the vehicle are chosen as the lines that bound the lane that is being driven on. The proposed algorithm works in an average of 12 milliseconds for each frame with 640×480 resolution on a 2.20 GHz Intel CPU. This performance metric shows that the algorithm can be deployed on minimal hardware and still provide real-time performance.**

Keywords-intelligent vehicle systems; lane detection; lane tracking; image processing

I. INTRODUCTION

Intelligent vehicle systems aim to assist the driver, or to control the vehicle autonomously. The main motivation for implementing assistive systems is to improve driving safety, and thus prevent traffic accidents caused by driver inattention or incompetence. One of the most studied topics among intelligent vehicle systems is lane detection. Lane detection is the extraction of road lane markings, with the purpose of using the obtained data with other intelligent vehicle systems (e.g. lane departure detection).

In this study, we propose a linear model lane detection algorithm that uses a single monocular image. Since lane detection algorithms are meant to be run on embedded systems, the speed and simplicity of the algorithm is of utmost importance for a fast response rate to be achieved.

II. RELATED WORK

Lane detection studies can be classified into two groups with respect to their working strategies: feature based and model based methods. Feature based methods usually extract edges or gradient vectors from the image and apply elimination with respect to predefined attributes (e.g. orientation, location) [1-4]. After the elimination step, remaining features are used to reconstruct the lane markings. The second group, i.e. model based methods, define a mathematical lane model which is composed of lines or curves, and fit this model on the input image [5-8].

Lane markings lie parallel to each other. Consequently, from the perspective of the driver, these lane markings converge at a point, which is referred to as the vanishing point. Wang and Chen [1], Felisa and Zani [9] and Coskun et al. [10] apply inverse perspective mapping (IPM) to project the input image to an image space where the lane features lie vertical to the x axis. With this operation, detection and classification of the line features become easier. Instead of employing IPM, Jung and Otsuka et al. [2], Kelber [7] and Schreiber et al. [11] detect the features that are oriented towards the vanishing point. The lane markings in the near and the far field may not be parallel to each other due to curves and slopes on the road. In this case, the lane markings in the far field will converge at a point different than the vanishing point (one can say that they create another vanishing point). Since a linear road model is still adequate for many intelligent vehicle systems such as lane deviation detection, Schreiber et al. and Fardi et al. use only a linear road model, neglecting the curves on the far field [11-12], as a linear road model is still adequate for many intelligent vehicle systems such as lane deviation detection. Jung and Kelber apply a hybrid approach, by using a linear model for near field and extending their model according to the curves in the far field [7].

As mentioned above, the position of the vanishing point is commonly utilized in lane detection. Studies that use IPM assume a constant single vanishing point, depending on the calibration of the camera. This approach will only be valid when there is no slope or unevenness on the road. Intersection of the two nearest lines to the vehicle has been used as the vanishing point [6-7, 11, 13]. This method will be very susceptible to occluded lanes and skid marks on the road. A more robust method of vanishing point detection is needed if one is to use the position of the vanishing point as the main tool of feature selection. We will propose such a method in Section 3.C.

978-0-7695-4875-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ISM.2012.70

As the feature detection method, edge detection algorithms, custom filters that expose specific types of lane markings or various machine learning methods are used in the literature. Nearly all studies use low-level features (mostly edge information). However, Felisa and Zani [9] and Tsai et al. [14] form edge segments as soon as edge detection is done. This approach is viable for lane detection, as the lanes to be detected are characterized by their contours.

III. Proposed Method

The algorithm can be briefly summarized as line segment detection, angle based elimination, vanishing point estimation, orientation based elimination, and finally lane detection using the remaining line segments. Each step will be discussed under the relevant subsection.

A. Line Segment Detection

The first step of the algorithm is extracting the line segments from the image. Hough Transform (HT) is a very common method used for line detection in the literature [5, 15-16]. However, HT is computationally intensive and does not extract the lines in segment form. Therefore, we employ EDLines, a recently proposed line segment detection algorithm [17]. EDLines runs very fast and is capable of detecting the lines in segment form without the need for any further processing. In Figure 1, an example image from the Caltech dataset [16] is shown with the line segments detected by EDLines overlaid on the image in green and blue colors. Green lines are located on the left half of the image and are used for left-lane detection; the blue lines are located on the right half of the image and are used for right-lane detection.

B. Angle Based Line Segment Elimination

After line segments are extracted from the image, two elimination steps are performed to remove the irrelevant line segments. Angle based line segment elimination is the first one of these elimination steps. As mentioned in the previous section, the line segments are divided into two subsets, namely left and right candidate sets. Separate threshold values are used to eliminate the elements of these two sets.

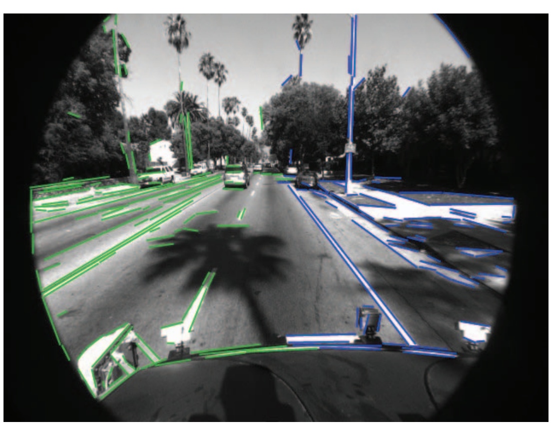
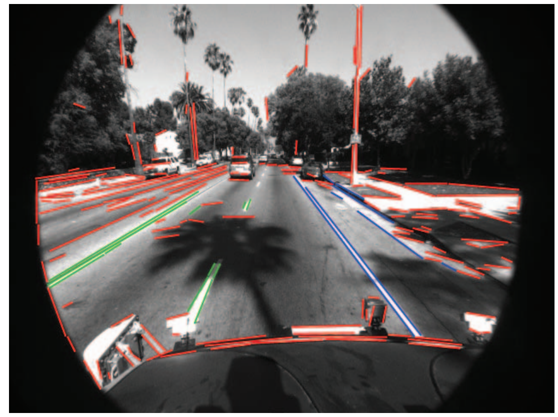
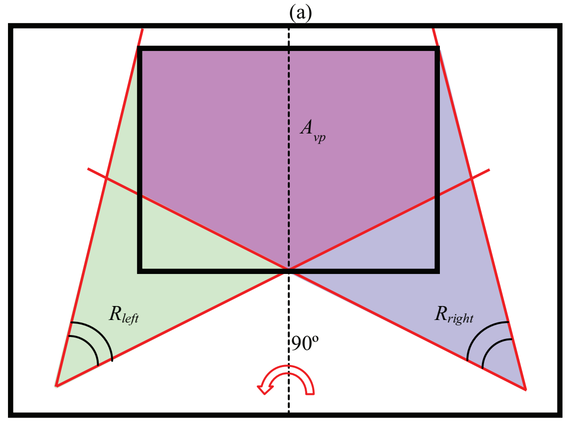


Figure 1. EDLines result for the test image. Green and blue lines correspond to left and right candidate sets, respectively.



(a)



(b)

Figure 2. (a) Selected and eliminated line segments according to their angles. Eliminated line segments are indicated in red color. (b) Illustration of angle based line segment elimination.

The line segment survives if its angle is in a certain range, i.e. R_{left} and R_{right} for lines in left and right subsets, respectively (refer to Figure 2.b). We have chosen these parameters to be R_{left} = (30°, 75°) and R_{right} = (105°, 150°). This step eliminates majority of the lines and in result, greatly reduces the processing time of the following steps. Let us investigate how angle based elimination affects the most critical step of our algorithm, vanishing point estimation. We are expecting to use the remaining lines after this elimination to estimate the position of the vanishing point. In Figure 2.a, the area where the vanishing point will be located after the elimination is highlighted in purple (A_{vp}) as the intersection area of the angle ranges R_{left} and R_{right}. We chose the elimination criteria such that A_{vp} contains all possible vanishing point locations for this specific application.

Figure 2.a shows the eliminated line segments in red, line segments selected for the left lane detection in green, and the line segments selected for the right lane detection in blue. Please note that line segments that satisfy the angle criteria are retained regardless of their location.

C. Vanishing Point Estimation

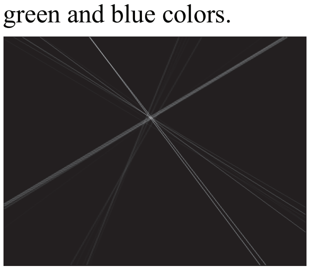
The previous step of the algorithm eliminated the majority of the redundant line segments. The next step of the algorithm aims to choose the line segments that define the lane markings. In this regard, we use the assumptions that the line segments extracted from the lane markings converge at the vanishing point, and the line segments whose extensions intersect the vanishing point are likely to define a lane division or a road boundary.

We utilize the assumption that the lane markings converge at the vanishing point to estimate the location of the vanishing point. After the elimination at the previous step, dominant parallel line segments at the scene will belong to lane markings and road boundaries. The point where the remaining line segments converge can be estimated as the vanishing point. The ordinary method to find this point would be exhaustively calculating the perpendicular distance from each pixel to all lines the line segments belong to, and choosing the pixel which most lines intersect. Since such an inefficient implementation is not applicable in a real-time system; a computationally efficient method is needed to find the dominant intersection point of the extended line segments.

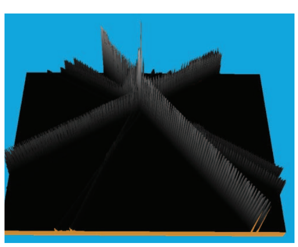
As the solution, we extrapolated the line segments along their directions in an image plane with the size of the road image. During the extrapolation, we apply a simple voting mechanism on pixels that the line segments intersect, weighted by the length of the segments. By this voting operation, we superimpose the discrete line segments to find the point where the majority of the lines converge. In Figure 3, we present the superimposed line extrapolation output for the test image where the vanishing point is clearly visible. Since this method needs to use each segment once, it is a linear time algorithm with *O(N)* complexity, where N is the number of detected line segments.

D. Orientation Based Line Segment Elimination

Once the vanishing point is detected, we simply compute the perpendicular distance between the extrapolated line segments and the vanishing point, and discard the line segments if the distance exceeds a certain threshold value (5 pixels for the experiments presented in this paper). Thus, the left and the right line segment sets are subjected to a further elimination according to their orientations. Figure 4 shows the vanishing point with a yellow cross and the remaining line segments on the left and right candidate subsets with green and blue colors.



(a)



(b)

Figure 3. (a) Superimposed image of the extended line segments. (b) 3D illustration of the intersection point of candidate lines.

Figure 4. Selected and eliminated line segments according to their locations. Vanishing point is indicated with a yellow cross sign.

Red line segments in the image are eliminated because their extensions are distant from the vanishing point.

E. Detection of Lanes

After eliminating the line segments according to their angles and orientations, multiple groups of line segments that belong to lane markings remain. To define each lane marking, line segments that belong to the same lane marking are clustered together with respect to their angle values, with the purpose of finding a single line that represents a cluster of line segments.

For this clustering problem, we used single-linkage clustering. This is an iterative method that clusters two points which are closest at each step. We determined a 10° of angle difference as the stopping condition, which means that any two line segments which have more than 10° of angle difference cannot be clustered together. After clustering the line segments, we generate a single line from each cluster of line segments. As one point of this line, we calculate the centre of gravity of the line segments that constitute this cluster. Earlier, we have assumed that lane markings intersect the vanishing point. Therefore, the second point of the generated line is the vanishing point detected in Section 3.C. Figure 5 shows the four candidate lines.

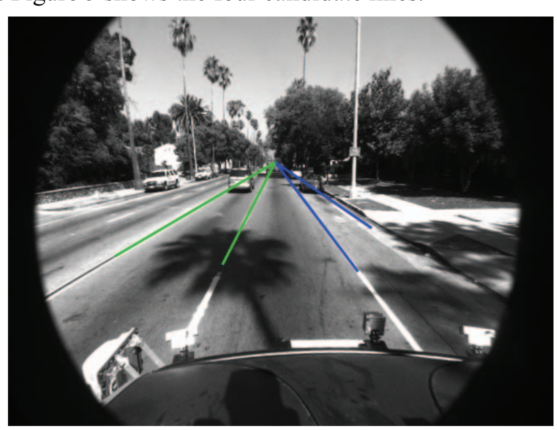


Figure 5. Lane candidates representing the clustered line segments.

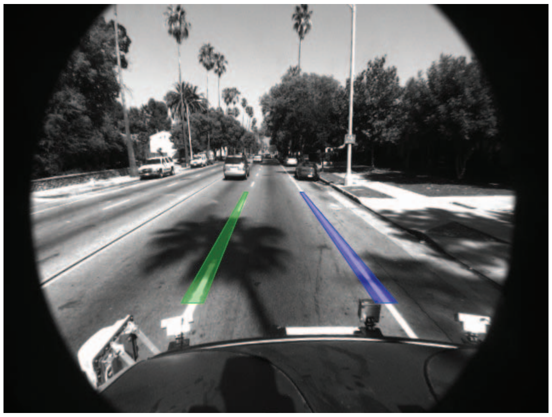


Figure 6. Right and left lanes selected among the lane candidates.

Having obtained several lines as lane candidates, two of these candidates should be chosen to represent the lanes. Here, we employ a very simple heuristic and choose the two lines that are closest to the left and right sides of the vehicle. Figure 6 shows the final detected left and right lane markings.

IV. Experimental Results

We tested our algorithm with the dataset presented in [16], which consists of 4 video sequences and 1225 frames in the size of 640x480. The average running time of our algorithm is 12 milliseconds on a 2.20 GHz Intel CPU. About 75% of the running time belongs to the line detection algorithm. Since we do not use temporal data to determine a region interest that the processing will be restricted to, running time for each frame is fairly stable.

Due to the fact that there is no established quantitative measure of lane detection validity, we manually evaluated the detection validity for each frame, similar to [1, 8, 10, 18]. The criteria for success were finding both lanes wherever possible and not giving any false positives. Accuracy rates for the separate sequences in the dataset are given in Table I. Please refer to the video on our website [19] for the results on Caltech Lanes Dataset [16].

TABLE I. Accuracy results for substituted datasets.

Dataset	Cordova 1	Cordova 2	Washington 1	Washington 2
Accuracy	98.8 %	98.3 %	91.4 %	95.3 %

V. Conclusion

We present a simple yet efficient algorithm for real-time lane detection. The proposed algorithm estimates the location of the vanishing point fast and accurately for every frame. The estimated vanishing point is used to determine the line segments that belong to the lane markings. Subsequently, lane markings are reconstructed using the line segments.

Many existing lane detection algorithms use the vanishing point for feature selection or image transformations and a linear lane detection algorithm to estimate a region of interest. Our study offers an alternative method to obtain this information in a computationally efficient manner.

References

[1] H. Wang and Q. Chen, "Real-time lane detection in various conditions and night cases," Proc. Intelligent Transportation Systems, pp 1226 – 1231, ITSC '06. IEEE, Toronto, ON, 2006,.

[2] Y. Otsuka, S. Muramatsu, H. Takenaga, Y. Kobayashi, and T. Monj, "Multitype lane markings recognition using local edge direction," in Intelligent Vehicle Symposium, 2002 IEEE.

[3] K. Y. Chiu and S. F. Lin, "Lane detection using color-based segmentation," Intelligent Vehicles Symposium, 2005 IEEE.

[4] Z. Kim, "Realtime lane tracking of curved local road," in Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE.

[5] Q. Chen and H. Wang, "A real-time lane detection algorithm based on a hyperbola-pair model," in Intelligent Vehicles Symposium, 2006 IEEE, Tokyo, 2006, pp. 510–515.

[6] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong and H. Chen, "A novel lane detection based on geometrical model and gabor filter," in Intelligent Vehicles Symposium (IV), 2010 IEEE.

[7] C. R. Jung and C. R. Kelber, "A lane departure warning system based on a linear-parabolic lane model," in Intelligent Vehicles Symposium, 2004 IEEE, 2004, Parma, pp. 891–895.

[8] R. Labayrade, J. Douret, and D. Aubert, "A multi-model lane detector that handles road singularities," in Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE, Toronto, ON, 2006.

[9] M. Felisa and P. Zani, "Robust monocular lane detection in urban environments," in Intelligent Vehicles Symposium, 2010 IEEE.

[10] F. Coskun, O.Tuncer, M. E. Karsligil and L. Guvenc, "Real time lane detection and tracking system evaluated in a hardware-in-the-loop simulator," in Intelligent Transportation Systems (ITSC), 2010.

[11] D. Schreiber, B. Alefs, and M. Clabian, "Single camera lane detection and tracking," in Intelligent Transportation Systems, 2005.

[12] B. Fardi, U. Scheunert, H. Cramer and G. Wanielik, "A new approach for lane departure identification," in Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, Columbus, OH, 2003, pp.100-105.

[13] L. Chuanjin, Q. Xiaohu, H. Xiyue, C. Yi and Z. Xin, "A monocular-vision-based driver assistance system for collision avoidance," in Intelligent Transportation Systems, 2003 IEEE.

[14] L.W. Tsai, J.W. Hsieh, C.H. Chuang and K.C. Fan, "Lane detection using directional random walks," in Intelligent Vehicles Symposium, 2008 IEEE, Eindhoven, 2008, pp. 303 - 306.

[15] T. Ogawa and K. Takagi, "Lane recognition using on-vehicle lidar," Proc. Intelligent Vehicles Symposium, pp. 540-545, Tokyo, 2006.

[16] M. Aly, "Real time detection of lane markings in urban streets," in Intelligent Vehicles Symposium, 2008 IEEE.

[17] C. Akinlar and C. Topal, "EDLines: Real-time line segment detection by Edge Drawing (ED)," Pattern Recognition Letters, vol.32, no.13, pp.1633-1642, October 2011.

[18] Wei Liu; Hongliang Zhang; Duan, B.; Huai Yuan; Hong Zhao; , "Vision-Based Real-Time Lane Marking Detection and Tracking," Intelligent Transportation Systems, 2008.

[19] http://ceng.anadolu.edu.tr/cv/LaneDetection/LaneDetection.htm