Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 4, Number 1, pages 65–68, January 2015

Multi-Lane Dection and Tracking using Dual Parabolic Model

Soon Kwon, Dajun Ding, Jongsu Yoo, Jekyo Jung and Sungho Jin
Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu, Republic of Korea
{soonyk, dajunding}@dgist.ac.kr

***Abstract*— Multi-lane recognition system gives a significant assistance for driver's safety. In this paper, we try to use a dual parabolic model for more precious and robust lane description. For gathering an efficient feature, we used a parallel RANSAC algorithm. The line model characterization is combined with adjacent point histogram peak analysis and two-axis parabola. Proposed algorithm can detect lane automatically and adaptively in every frame. The designed method is applied using various video images captured from the commercial black box camera and is verified to be robust.**

Keywords*— *Multi-lane detection; RANSAC; Dual Parabola

I. Introduction

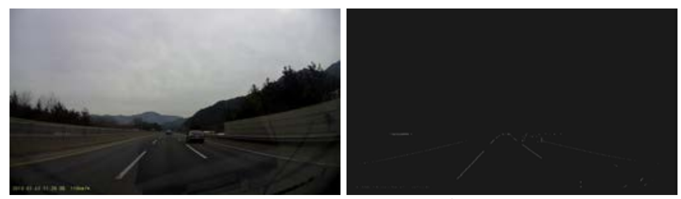
In the ADAS(advanced driver-assistance system), vision based on-road environment sensing issues are being intensively considered over the past few decades and now it moves to the higher functionalities for applying autonomous driving or emergency driving support[1].

As one of higher level sensing functionalities, multi-lane detection can gives further information than conventional ego-lane method. In multi-lane detection, the algorithm detects not only ego-lane where the vehicle is driving, but also the lanes adjacent to the ego-lane. We proposed multi-lane detection algorithm which consists of the following three steps: lane feature extraction, geometrical model estimation and the model parameters' tracking. We first use the road ROI setting[2] for saving processing time. And we use a band pass filter to get the lane features[3][4]. For the geometrical model estimation, we tried a multiple lane description method based on the RANSAC straight lines' histogram[5] and RANSAC parabolic model. The four straight lines will be evaluated by road width as well as vanishing point detection. In recent papers[6-9], other researcher used various curve line model for In an adjacent searching region, a dual coordinate axis parabola approximation based on least square method will be applied. Lanes are displayed as horizontal parallel lines which are adjacent to curves results. We just track the linear straight line instead of whole curve in the last step.

This structure of this paper is described as follows. The dual RANSAC based method for both linear and curve lane feature description is shown in Section II. In this section, we used the line slope angle histogram for multi-lane detection. The lane markings' number in current frame is related with the histogram peaks' number. Finally, section III gives concluding remarks.

II. Lane Module Description Using Ransac

The lane markings have dark-white-dark natural structure. In the feature extraction step, we used a scale-changed one-dimensional kernel for lane detection. The lane extraction result is shown in Fig.1.



a. input image　　b. band pass filter result
Fig.1. Lane marking extraction result

Secondly, we considered geometrical model estimation. In this step, we apply the straight line detection based on a linear line model, dual-axis parabola is followed for further characterization. And a flexible method for lane marking result highlighting is present in the end of this section.

In the previous step, the output image contains thousands of separating points. Most of them are from the correct lanes, and a number of these points are from vehicles or barriers. We use RANSAC for correct points concentrating and redundant points removing. Flowchart for describing the algorithm is shown in Fig.2. Proposed process consists of 4 main parts: A. Using RANSAC algorithm for data collection. B. Applying the angle histogram for linear model parameter analysis. C. Curvature analysis and dual-axis parabola. D. Result tracking.

A. RANSAC data collection

In the getting random points pairs step, we select two points from left keypoints and right keypoints group separately. The total number of point pairs in each side is

$$\mathrm{N}_{\text{total}} = C_N^2 = \frac{N!}{(N-2)!*2} \tag{1}$$

where N is the total keypoints number on left or right side. The bigger RANSAC sample number is, the probability of covering all point pairs will be larger. As the original RANSAC algorithm takes too much time in this iterative step, we use GPGPU for the computing acceleration. In that case, the original iterative RANSAC is turned to a parallel issue.

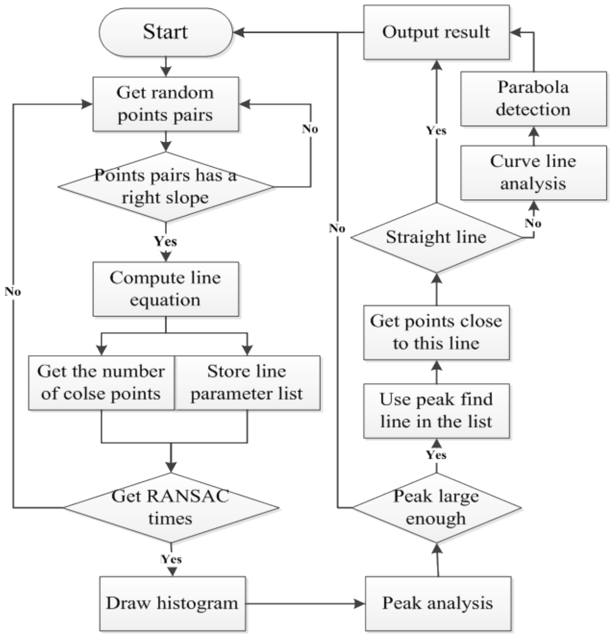


Fig.2. Flow chart

After we select two points P1 and P2, the angle slope filter is used for making sure all these two points can fit the correct slope range. If we use a coordinate system like Fig3.a, all left lines' slope should have a negative value. Any points pair with a wrong slope is ignored, and a new random pair will be selected until the slope was right.

Those temporary testing lines' slope angles are then got by the equation like (2).

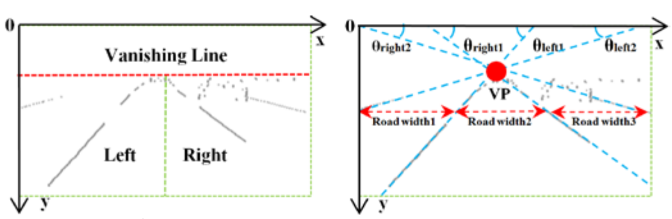
$$\theta = \arctan(\frac{P1.y - P2.y}{P1.x - P2.x}) \quad (2)$$

The line's slope k and intercept d are stored in the line parameter list for further inquiring.

City block distance of the current temporary line and all these keypoints are calculated for adjacent clustering in (3).

$$D = \min(abs(y - k * x - d), abs(x - y / k + d / k)) \quad (3)$$

Any point's distance that is smaller than 2 will be gather into this line's group.

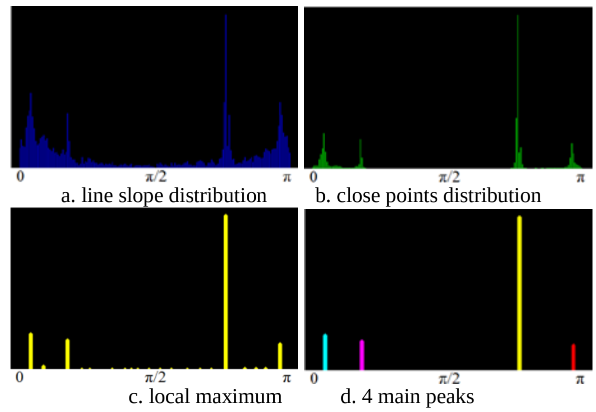


a. left and right image b. 4 lane markings' slope angle

Fig.3. Coordinate System

B. *Linear line model parameter analysis*

We draw the histogram of the adjacent points groups. Fig.4.a shows the distribution of the line slope, and Fig.4.b shows the close points' allocation histogram. The line slope histogram and the close point histogram have a similar apportion. However, the points' histogram is more obvious for crest detection with a simple thinking. That is because adjacent point histogram has a double strengthening to describe straight lines. In that case, we assigned the adjacent point histogram for further computing.



a. line slope distribution b. close points distribution

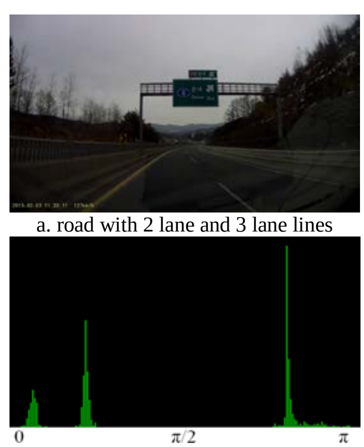
c. local maximum d. 4 main peaks

Fig.4. Angle Histogram

Four main peaks are detected in the bar chart. Every local maximum bar $\theta_{LocalMax}$ in histogram is selected if this bar is the largest one among a range as $[\theta_{LocalMax} - \pi/36, \theta_{LocalMax} + \pi/36]$. Local crest result is shown in Fig.4.c. Then two maximum local peaks from left or right part are detected easily based on the previous result. Four main peaks result is shown in Fig.4.d.

Correct straight lines should fit the following requirements in order to removing false results. The road width in each line should be in a same proper range. All vanishing points which are composed of the intersection points of line and horizon line should have a sufficiently small variance amount.

We cared about these four peaks, nevertheless, not all roads have the same situation. In our system, if the second maximum peak is 15 times lesser than the maximum one, we will ignore it immediately. For example in Fig.5.a, it has two lanes and three road markings lines. Fig.5.b shows that the point histogram peak can illuminate the verdict clearly.



a. road with 2 lane and 3 lane lines

b. close points histogram

Fig.5. Non-four lane lines situation

After the crest analysis we only have the slope data of these appropriate lines. Full line equation should be checked in the line parameter list which we've stored already. It is also a parallel problem in this step. Inside of every GPGPU item, we compare the difference between the slope angle we've detected in the maximum crest and the testing one in the parameter list. If the difference is smaller than 5%, we will output the m_{th} peak's line formula result using a weighting method like (4-6):

$$y = k_n x + d_n \tag{4}$$

$$k_m = \sum_{n=1}^{total\_num} \left(k_n * \frac{score_n}{score_{total}}\right) \tag{5}$$

$$d_m = \sum_{n=1}^{total\_num} \left(d_n * \frac{score_n}{score_{total}}\right) \tag{6}$$

Which k_n and d_n are the current slopes and intercepts which we've selected in parameter list. And km and d_m is the output result of the m_{th} peak's line. If the current testing line in parameter list has a bigger score, it shall have more influence with the peak line output result.

C. *Parabolic line model parameter analysis*

The straight line formula which we've detected in last step gives us the approximate line of the road markings. We use a searching region for the curvature analysis. Every keypoint (x0,y0) will be collected into a curve-points group if it fits (7).

$$abs(k_m * x_0 + d_m - y_0) < Width_{group} \tag{7}$$

The $Width_{group}$ is the current group searching width, it is determined by empirical method. In our system, it is set as $image_{width}/50$ for scale invariance.

According to the structure of the curvature, we use two kinds of parabola that centered in x and y coordinate axis (1-2). Predication of the current group's line curvature is judged by (8), the C_m is the curvature analysis result in the line m's group.

$$C_m = \sum_{n=1}^{points\_num} k_m * x_n + d_m - y_n \tag{8}$$

One curvature analysis example is shown in Fig.6. As positioning in left lane, the current group's straight line slope k_m has a negative value. T_c is the curvature analysis threshold. If the road marking is adjacent to the approximate straight line, the curvature response C_m will fit the judge of $-T_c<C_m<T_c$. If the point distribution fit $C_m>=T_c$, it means more points are located on the right part of the center line such as the blue curve in Fig.6. In that case, approximation for parabola which is centered in x axis (1) will be applied to describe this curve. On the other hand, any curve lines that fits $C_m<=-T_c$, will be estimated by (2) such as the red curve in Fig.6.

After the parabola model analysis, Least Square combined with RANSAC is used for parabola's formula calculation. The function solving problem will be done if we can find a best fit of the data to a general quadratic equation.

$$d = c_3 t^2 + c_2 t + c_1 \tag{9}$$

The functional form for error is a simple generalization of the linear error function (10).

$$Error = \in (c_1, c_2, c_3) = \sum_{i=1}^{n} 2 * (c_3 t_i^2 + c_2 t_i + c_i - d_i) \tag{10}$$

The minimum error is at the point where the partial derivatives of the error function with respect to the coefficients are all zero. The equation results from evaluating the partial derivative. Using standard notation for linear algebra, the relationship of (10) can be solved by (11):

$$\begin{bmatrix} n & \sum_{i=1}^{n} t_i & \sum_{i=1}^{n} t_i^2 \\ \sum_{i=1}^{n} t_i & \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i^3 \\ \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i^3 & \sum_{i=1}^{n} t_i^4 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{bmatrix} \sum_{i=1}^{n} d_i \\ \sum_{i=1}^{n} d_i t_i \\ \sum_{i=1}^{n} d_i t_i^2 \end{bmatrix} \tag{11}$$

RANSAC algorithm is applied again for parabola formula calculation. In our method, we select 5 controlling points for every parabola's approximation. Two of them are these points with the maximum or the minimum y coordinate, which means they are endpoints of the current curve. We need only three random points in the middle of the curve. As the points number amount is smaller and the computation complexity is larger than the straight line detection, the iterations of RANSAC in this step is reduced to 1/10 of the straight line step. Fig.5.a's keypoints happened to fit three different line models, x-axis parabola shown in red, y-axis parabola in blue and straight lines in green. The parabola detection result is shown in Fig.6.

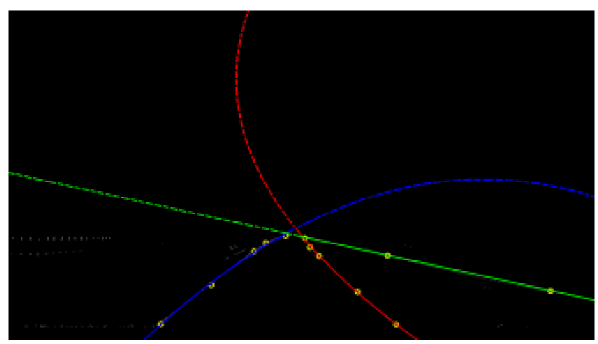
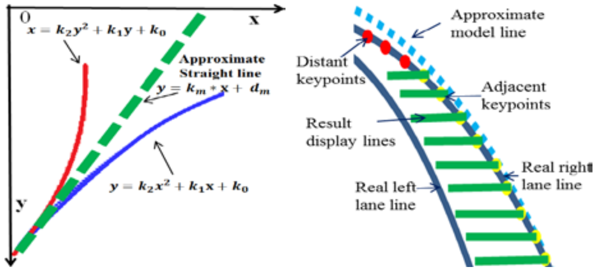


Fig.6 Parabola approximation result

D. *Result tracking*

Even though any line has two parameters, as the lane could only be changed continuously. Just the line slope angle tracking is more important for wrong results prevention. If the angle between two frames is changed over $\pi/18$, we will treat this value as noise and use previous result as the input data instead. We use a simple Kalman filter for this tracking part.

After getting the line functions of the road markings, result displaying is followed here. The displaying method is shown in Fig.7.b. We compute the city block distance between every keypoint and the approximate model line(straight or curve lines). If distance is less than the kernel width, we will draw a 2-step-width segment. Orientation is within the same direction of the left or right part image. If the line model locates on the left part image, the line will be draw to left for covering the road paintings.



a. two parabolic line models b. result display method
Fig.7. Parabola result description and display

Experiment results with regarding to different situations are illuminated in Fig.8.

III. Conclusion

In this paper, we proposed a multiple lane detection method using RANSAC. Features are collected by a simple band-pass filter. Parallel RANSAC algorithm is then applied for feature assembling. A line model description method which combined histogram peak analysis and two-axis parabola is used for line model characterization. As shown in our experiment results, the road markings can be descripted correctly and efficiently.

Acknowledgment

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT and Future Planning (14-IT-02, 14-HRLC-01).

References

[1] McCall J C, et al. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation", Intelligent Transportation Systems, IEEE Transactions, pp.20-37, 2006

[2] Dajun Ding, Chanho Lee, et al, "An Adaptive Road ROI Determination Algorithm for Lane Detection", TENCON 2013, IEEE, pp.1-4, 2013

[3] Veit, Thomas, et al. "Evaluation of road marking feature extraction." Intelligent Transportation Systems, ITSC, IEEE, pp.174-181, 2008

[4] Huang A S, Moore D, Antone M, et al. "Finding multiple lanes in urban road networks with vision and lidar," Autonomous Robots, pp. 103-122, 2009

[5] Aly M. "Real time detection of lane markers in urban streets." Intelligent Vehicles Symposium(IV), IEEE, pp.7-12, 2008

[6] C. Lipski, B. Scholz, K. Berger, et al, "A fast and robust approach to lane marking detection and lane tracking," IEEE, Southwest Symposium on Image Analysis and Interpretation, pp.57-60, 2008

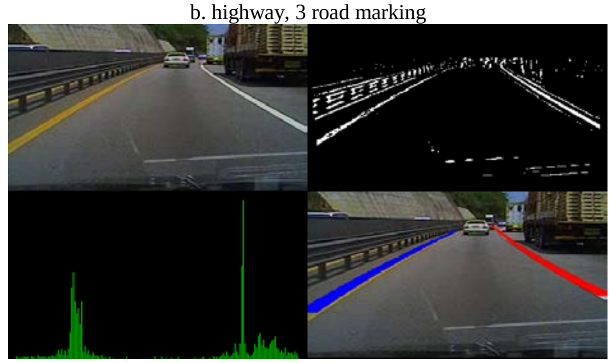
[7] Y. Wang, E. K. Teoh, et al, "Lane detection and tracking using b-snake," Image and Vision Computing, vol. 22, pp. 269–280, 2004

[8] Zhao Kun, et al. "A novel multi-lane detection and tracking system" Intelligent Vehicles Symposium (IV), IEEE, pp.1084-1089, 2012

[9] Li Qing, et al, "Springrobot: A prototype autonomous vehicle and its algorithms for lane detection." Intelligent Transportation Systems, IEEE Transactions on, 5(4): pp. 300-308, 2004

a. city road, 4 road marking

b. highway, 3 road marking



c. highway, 2 road marking
Fig.8. Input images, Filter result, Histogram and Result displaying