A Real-time System of Lane Detection and Tracking Based on Optimized RANSAC B-Spline Fitting

Jiayong Deng
Vision System Laboratory, Department of Electronic Engineering, Soongsil University
511 Sangdo-Dong, Dongjak-Gu, Seoul, Korea
+82-02-821-2050
djiayong5@gmail.com

Youngjoon Han
Vision System Laboratory, Department of Electronic Engineering, Soongsil University
511 Sangdo-Dong, Dongjak-Gu, Seoul, Korea
+82-02-820-0699
young@ssu.ac.kr

ABSTRACT
In driving assistance systems, lane detection can provide significant information for driving safety. In this paper, we proposed a novel real-time lane detection method to extract the location of lane marking lines based on inverse perspective mapping transform (top view) for the region of interest (ROI) of a video frame. The data were then filtered by a selective oriented Gaussian high pass filter, Hough transformation, and Kalman filter to give the initial regions to our optimized RANSAC (Random Sample Consensus) Bezier splines fitting algorithm, which is the main innovation in this paper. Our experimental results and accuracy evaluation indicated that the proposed lane detection algorithm could run robustly in real time, and could achieve an average speed of 32.32 ms per frame for a 320 $\times$ 240 pixel image and 41.64 ms for a 640 $\times$ 480 pixel image, with a correct detection rate of over 92.85%. Moreover, our method was found to be suitable for various conditions.

Categories and Subject Descriptors
I.4 **[Image Processing and Computer Vision]**: Applications

General Terms
Algorithms, Performance

Keywords
Bezier spline, Gaussian filters, Hough transform, Kalman, Lane detection, RANSAC, Top view.

1. INTRODUCTION
A rapidly increasing number of traffic accidents are occurring due to the increase in car ownership on a yearly basis all over the world. Moreover, many accidents involving roadway departure crashes are caused by driver inattention.

An estimated 1.2 million people are killed in road crashes every year around the world, and as many as 50 million are injured. Projections indicate that these figures will increase by about 65% over the next 20 years unless there is a new commitment to prevention [1]. A study conducted in 1985 by K. Rumar, using British and American crash reports as data, found that 57% of roadway and driver factors; 6% to combined vehicle and driverfactors; 3% solely to roadway factors; 3% to combined roadway, driver, and vehicle factors; 2% solely to vehicle factors; and 1% to combined roadway and vehicle factors [2].

Automated driving systems may help to reduce the huge number of human fatalities and injuries resulting from car accidents [3]. The road detection algorithm is one of the key technologies in such a system. Many commercial lane-detection systems are available, and these have shown good performance in many challenging road and illumination conditions. However, although delivering robust results, they give information on lane positions rather than lane curvature [4].

Over the last two decades, a significant amount of research has been carried out in the area of road/lane analysis. This topic can be separated into two essential building blocks: lane detection and tracking [5]. There are several useful technologies employed in lane detection that have achieved good results in terms of application requirements; these include the open uniform B-spline curve model [6], multiple hyperbola road model [7], and a K-means cluster algorithm [8]. Furthermore, essential tracking technologies like the Kalman filter and particle filter are frequently utilized. However, most existing algorithms focus on lane detection where there are few vehicles and obstacles ahead on highways and clearly marked roads; this is an easier task compared to lane detection on general streets.

This paper presents a robust and effective approach to extracting the location of road marking. It is based on taking a top view of the image in a region of interest (ROI), and is termed inverse perspective mapping (IPM). The image is filtered by Gaussian spatial filters in a vertical orientation and then thresholded by keeping only the highest values. In order to eliminate edges around vehicles ahead and all kinds of shadows, an effective template is utilized to obtain a satisfactory result; this is followed by Hough transformation and classification for Hough lines. Our line classifier can separate all of the detected lines into several groups and calculate the central line from every group, providing the initial boundary of lane marking for the following step. Actually, the boundary based on Hough transformation is a shrunken and separated ROI. Subsequently, a novel optimized RANSAC three-degree Bezier spline fitting algorithm is carried out to refine the detected straight lines and correctly detect curved lanes. Finally, a Kalman filter is applied to track the location of lane-marking boundaries. The schematic workflow of the entire detection procedure is shown in Figure 1.

2. APPROACHES
2.1 Inverse Perspective Mapping (IPM)
In the Euclidean space, two kinds of description of traffic scenes are defined, that is, the world coordinate system W and the image coordinate system I. The coordinate transformation relationship from I to W is shown in Figure 2.

Assuming that the coordinates of the mounting position of the camera in the world coordinate system is (d, l, h), the calibrated parameters of the camera are as follows: γ represents the angle between the projection line of the optical axis at the z=0 plane and y-axis; θ represents the deviation angle of the optical axis to the x=0 plane; $2\alpha_u$and $2\alpha_v$ represent the field of view of the camera in the horizontal and vertical directions, respectively.

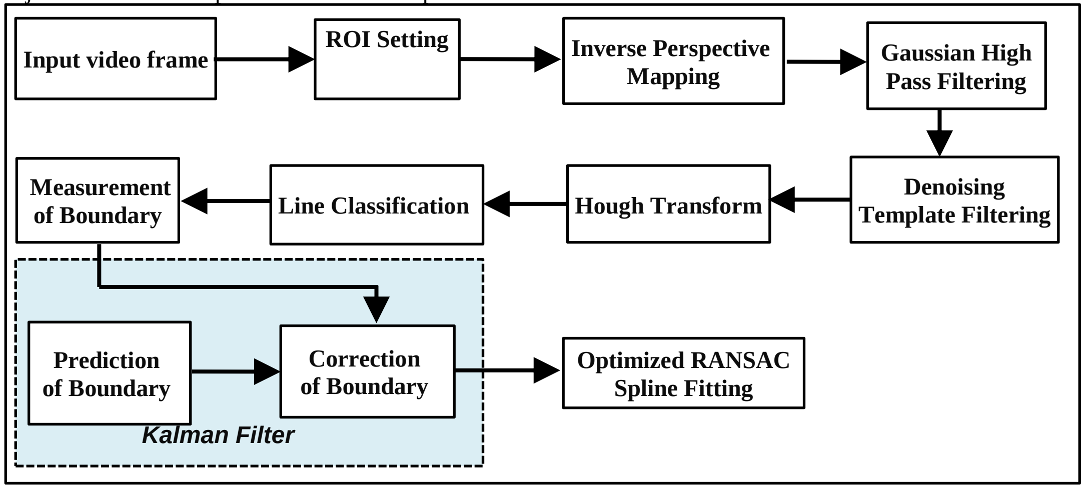


Figure 1. Framework of the lane-detecting procedure.

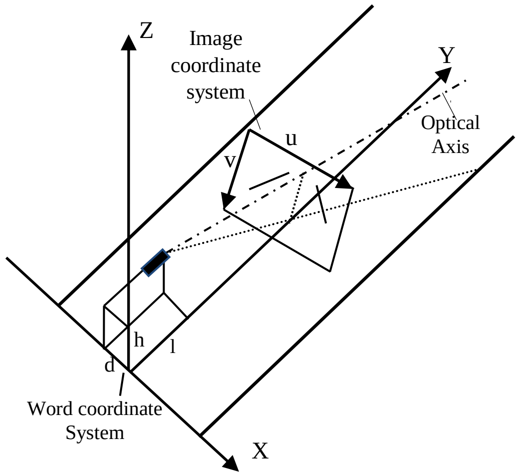


Figure 2. Schematic diagram of IPM.

The inverse perspective transform model from I to W using coordinate transformation, is shown in equation (1) and (2). However, in practical systems, the direct utilization of equation (1) and (2) will bring about blank pixels in the resultant image, because these equations are nonlinear and such a transformation is not bidirectional one-to-one mapping, but rather unidirectional mapping [9]. Therefore, we adopt a reverse IPM transform model as shown in equation (3) and (4), which stuffs the resultant image by calculating its mapping source in the original image.

$$x(u,v) = h \times \cot\left(\frac{2\alpha_v}{R_v - 1} \times v - \alpha_v + \theta\right) \times \sin\left(\frac{2\alpha_u}{R_u - 1} \times u - \alpha_u + \gamma\right) + d. \tag{1}$$

$$y(u,v) = h \times \cot\left(\frac{2\alpha_v}{R_v - 1} \times v - \alpha_v + \theta\right) \times \cos\left(\frac{2\alpha_u}{R_u - 1} \times u - \alpha_u + \gamma\right) + l\,. \tag{2}$$

$$v = \frac{(R_v - 1) \times (arctan\frac{h}{\sqrt{(x-d)^2 + (y-l)^2}} + \alpha_v - \theta)}{2\alpha_v} \tag{3}$$

$$u = \frac{(R_u - 1) \times (arctan\frac{(x-d)}{(y-l)} + \alpha_u - \gamma)}{2\alpha_u} \tag{4}$$

In this way, we can ensure that each pixel in the resultant image will find a correspondence in the original image, and accordingly remove blank points effectively; the IPM image is shown in Figure 3(b) and the red rectangle region in Figure 3(a) is the ROI.

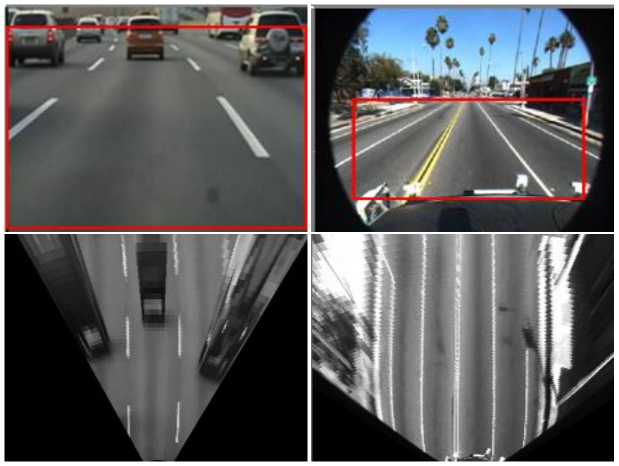


Figure 3. Results of IPM algorithm: (a) Original image (the first two pictures), (b) result imaging with IPM transformation (the last two pictures).

2.2 Gaussian Filter and Binarization

The transformed IPM image is filtered by a two-dimensional Gaussian kernel. The vertical direction is a Gaussian high pass filter whose ρ_y is adjusted according to the specific requirements of the lane segment (set to the equivalent of 10) to be detected, and the convolution function shown in equation (5). The horizontal direction is a Gaussian low pass filter whose ρ_x is adjusted according to the specific requirements of the lane segment (set to the equivalent of 5) to be detected, and the convolution shown in equation (6).

$$f_v(y) = \mathrm{EXP}\left(-\frac{2 \times \rho_y^2}{y^2}\right) \tag{5}$$

$$f_u(x) = \mathrm{EXP}\left(-\frac{x^2}{2 \times \rho_x^2}\right) \tag{6}$$

The filter is tuned specifically for vertical bright lines on a dark background of a specific width, which is our assumption of lanes in the IPM image; however, it can also handle quasi-vertical lines, which produce considerable output after the thresholding process.

Use of this separable kernel allows for efficient implementation, and is much faster than using a non-separable kernel [3]. The filtered image is shown in Figure 4(a). As can be seen from the filtered image, it exhibits a high response to lane markers, so we only retain the highest values. We keep the actual pixel values of the thresholded image, which represent the input in the following steps. In this step, we use the assumption that the vehicle is parallel/near parallel to the lanes. Figure 4(b) shows the result image after binarization.

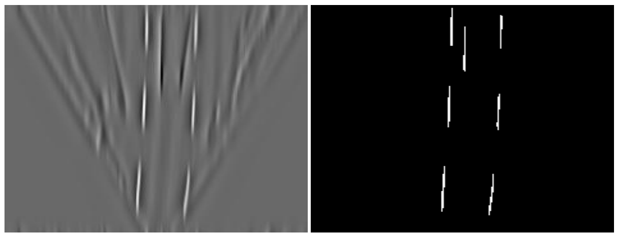
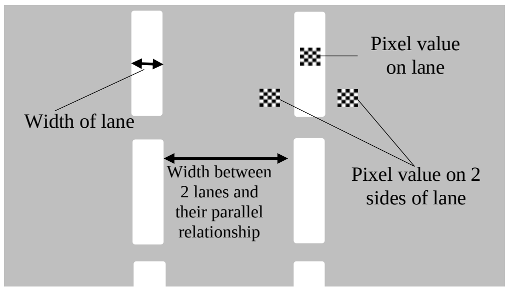


Figure 4. Filtering and thresholding the image: (a) Images after filtering, (b) image after thresholding.

2.3 The Denoising Template

This paper proposes a novel template denoising algorithm. The template was designed according to the following rule set with a specific pixel value, width, edge orientation of lane marking, and gray distribution of two sides of the lane. Figure 5 shows the characteristics of lane-marking distribution used in the denoising template.

Our proposed algorithm is effective, especially when there are many vehicle edges ahead or shadows of objects such as trees, telegraph poles, traffic signs, and so on. It can be shown theoretically and experimentally that the computational cost of our algorithm is much smaller than that of the existing methods. The resulting denoising images are given in Figure 6(b).


Pixel value on lane
Width of lane
Width between 2 lanes and their parallel relationship
Pixel value on 2 sides of lane
Figure 5. Characteristics of lane-marking distribution.

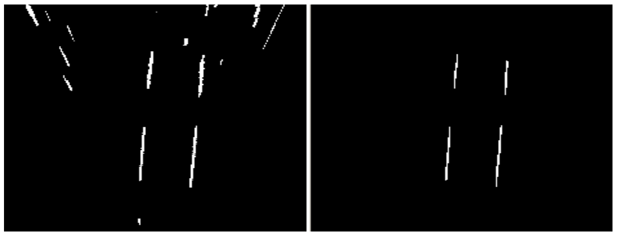


Figure 6. (a) Input binary image, (b) resulting image after filtering by the denoising template.

2.4 Lane Boundary Detection and Tracking

This stage is concerned with detecting lines in the denoising image. We use two techniques: Hough transform to estimate the location of lane marking where Hough lines are distributed together, and a line classifier.

The Hough transform approximately detects straight lines in a vertical orientation. Moreover, the nearby lines are grouped together to eliminate multiple responses to the same line; thus, all of the detected lines are classified into several line groups which form specific lane markings according to the line classifier. Figure 7(b) shows the resulting image after Hough transform. For every line group, the line classifier can also calculate its central line and extract the corresponding boundary, as shown in Figure 7(c) and Figure 7(d).

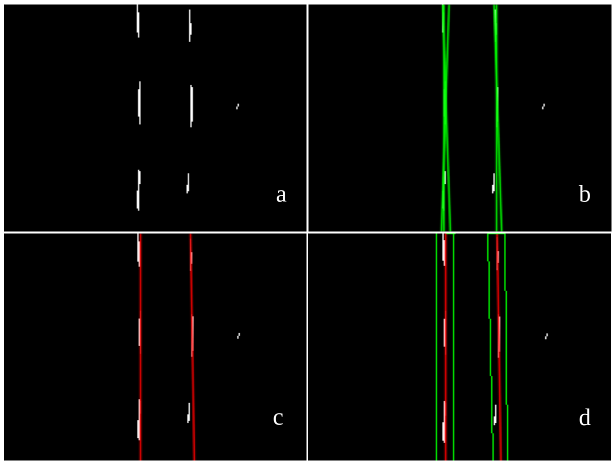


Figure 7. (a) Input image, (b) result image after Hough transform, (c) resulting image with central lines, (d) resulting image with boundaries.

It is very important to track the lane-marking boundary for driver assistance. Kalman filtering allows a linearized version of the system dynamics to be incorporated in order to generate optimal estimates under the assumption of Gaussian noise. Boundary tracking can provide improved results in noisy situations and generate useful metrics for lane detection in the next frame.

The Kalman filter estimates a process by using a form of feedback control: The filter estimates the process state at a certain time and then obtains feedback in the form of measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback, that is, for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate [10].

The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed, the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems.

In the typical Kalman filter, the system and measurement model and the Kalman update equations from time k-1 to k are defined in simple equations as follows.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_k \tag{7}$$

$$z_k = Hx_k + v_k \tag{8}$$

Here, A is an n × n state transformation matrix, while the n × l matrix B relates the control input u_{k-1} to the state x_k as follows shown in equation (9).

$$x_k = \begin{bmatrix} x_{top-left} \\ y_{top-left} \\ v_{x(top-left)} \\ v_{y(top-left)} \\ x_{bottom-right} \\ y_{bottom-right} \\ v_{x(bottom-right)} \\ v_{y(bottom-right)} \end{bmatrix} \tag{9}$$

Here, x_k is the state model of the lane boundary and $x_{top-left}$, $y_{top-left}$, $x_{butt-right}$, and $y_{butt-right}$ are the x, y coordinates of the top-left and button-right of the rectangle boundary. In the state transformation matrix, $v_{x(top-left)}$, $v_{y(top-left)}$, $v_{x(butt-right)}$, and $v_{y(butt-right)}$ are the corresponding velocities of the x and y direction. The velocity is defined as the difference of x and y coordinates between the current frame and previous frame. The m × n H matrix relates the state to the measurement z_k as shown in equation (10).

$$z_k = \begin{bmatrix} x_{top-left} \\ y_{top-left} \\ x_{bottom-right} \\ y_{bottom-right} \end{bmatrix} \tag{10}$$

The random variables w_k and v_k are the process and measurement noise, respectively. They are assumed to be independent of each other and covariance is Q and R, which has a zero average and white Gaussian probability distribution. This is shown in equation (11) and (12).

$$Q = E[w_k {w_k}^T] \tag{11}$$

$$R = E[v_k {v_k}^T] \tag{12}$$

In our experiment, the Kalman filter was applied to the tracking lane boundaries, whose state variables are updated using the coordinates of the lane-marking boundary. Considering that there are only small changes between two consecutive frames, we can regard the estimated parameters of the boundaries in the previous frame as the initial parameters for the current frame. These measurements are then used to update the discrete-time Kalman filter for the road and vehicle state.

The tracking parameters of the lane-marking boundary consist of the top-left and bottom-right coordinates, which are shown as the red points in Figure 8.

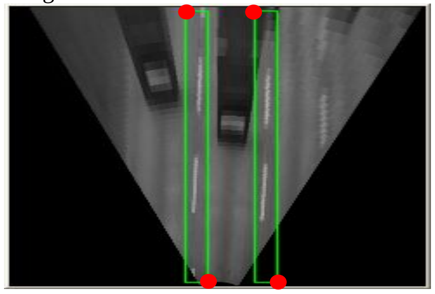


Figure 8. Location of the boundaries is shown with red marks in the IPM image.

There is no control input, and therefore $U_{k-1}=0$. The typical Kalman filter tracks the boundary using the Q (process noise covariance) and R (measurement noise covariance), which are initialized as constant values and do not change until the tracking is over. To track the lane markings, the results of the separation and the dynamic Kalman filter are used. Thus, we first need to generate the model of the state and parameters in the Kalman filter as follows.

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

For solid lane markings, the Hough transform provides a robust means of determining the location and angle of individual lane markings. When a valid lane marking cannot be detected using Hough transform, the estimation by Kalman filter can provide the measurement value instead. Consequently, lane-boundary detection using Hough transform provides a critical initialization of the candidate range for the RANSAC algorithm described in the next section.

2.5 Optimized RANSAC B-Spline Fitting

The previous step gave us candidate ranges for the image; these are used in the present step. For each boundary, we will run the spline fitting algorithm.

The spline used in the present experiment is a third degree Bezier spline, which has the useful property that the control points form a bounding polygon around the spline itself. The equation of the third-degree Bezier spline is defined below.

$$p(t) = \frac{1}{6}[1 \ \ t \ \ t^2 \ \ t^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}, t \in [0,1] \tag{15}$$

Here, $t \in [0, 1]$, $p(0) = P_0$ and $p(1) = P_3$ and the points P_0 and P_3 control the shape of the B-spline (Figure 9).

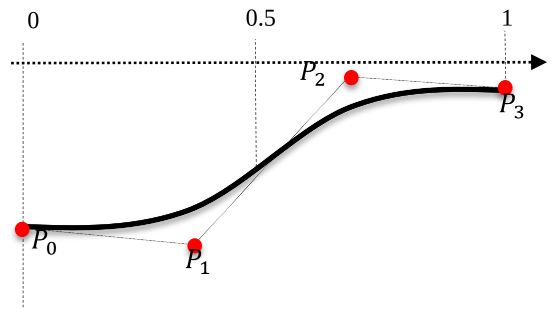


Figure 9. Schematic diagram of a three-degree Bezier spline.

In the loop procedure of optimized RANSAC B-spline fitting, in a normal RANSAC algorithm, we would be interested in computing the normal distance from every point to the third-degree B-spline; however, this method would require us to solve the problem through time-consuming processing to calculate the recent distance from every candidate point to current spline. Instead, we decided to follow a more efficient approach. Because the distribution of candidate points is on the vertical direction, so we just need to compute the distance in horizontal orientation from every candidate point to current spline, and set the reciprocal of average distance as the spline score. The final step is an iterative procedure. We assume that iterations equals N, and we just want to keep the max score and its corresponding control points as the fitting result from entire loops. The iteration of RANSAC B-Spline fitting is shown as Figure 10.

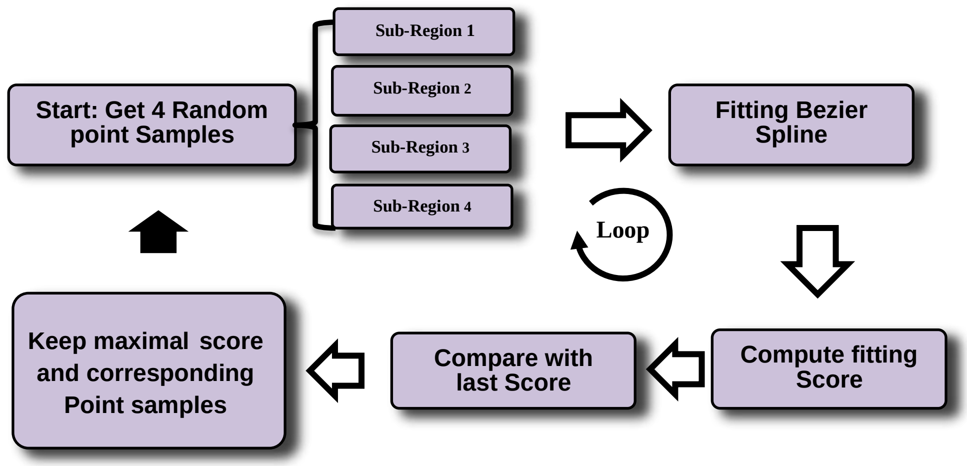


Figure 10. Iteration of RANSAC B-spline fitting.

As we know, obtaining sample points in the general RANSAC algorithm is absolutely random; there is just one candidate region for the selection of all sample points. This is a time-consuming process. However, our optimized RANSAC algorithm can largely improve the efficiency, as our method can save at least 60% of the run time used by the general RANSAC algorithm and achieve a same precision. The relationship between candidate points and sub-regions is shown as Figure 11.

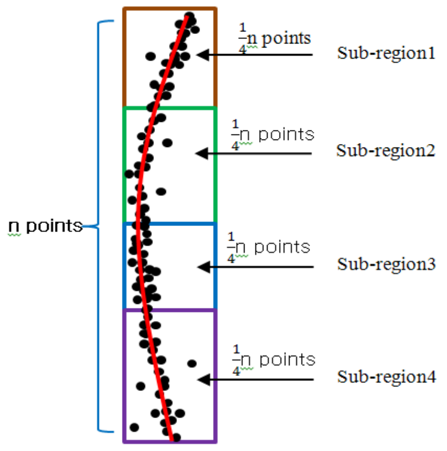


Figure 11. The relationship between candidate points and sub-regions.

The entire procedure of the RANSAC Bezier spline-fitting algorithm is shown in the following representation.

Optimized RANSAC three-degree Bezier spline fitting algorithm:

```
Sub-R1=SplitToSubRegions(AllCandidatePoints, 4, 1)
………
Sub-R4=SplitToSubRegions(AllCandidatePoints, 4, 4)
for i=0 to iterations do
    SamplePoint1=GetRandomSamples(Sub-R1)
    ………
    SamplePoint4=GetRandomSamples(Sub-R4)
    SamplePoints=JoiningPoints(SamplePoint1, SamplePoint2,
    SamplePoint3, SamplePoint4)
    Spline=RansacBezierSplineFitting(SamplePoints)
    Score=ComputerScore(AllCandidatePoints, Spline)
    if Score > BestScore then
        BestScore = Score
        ControlPoints= SamplePoints
    end if
end for
```

We tested the run time between normal RANSAC and our optimized RANSAC for six different-sized figures, and the experimental result is shown as Figure 12; moreover, the calculation of run time (shown as Table 1) indicates that the optimized RANSAC B-spline fitting utilized approximately 30 percent of the run time used by normal RANSAC B-spline fitting. Figure 13 shows the step result of our optimized RANSAC algorithm.

Table 1. Run time test of RANSAC algorithm

Run time (ms)	**Fig. 1**	**Fig. 2**	**Fig. 3**	**Fig. 4**	**Fig. 5**	**Fig. 6**
Normal RANSAC	173	792	1621	1630	1639	1505
Optimized RANSAC	63	265	531	500	531	500

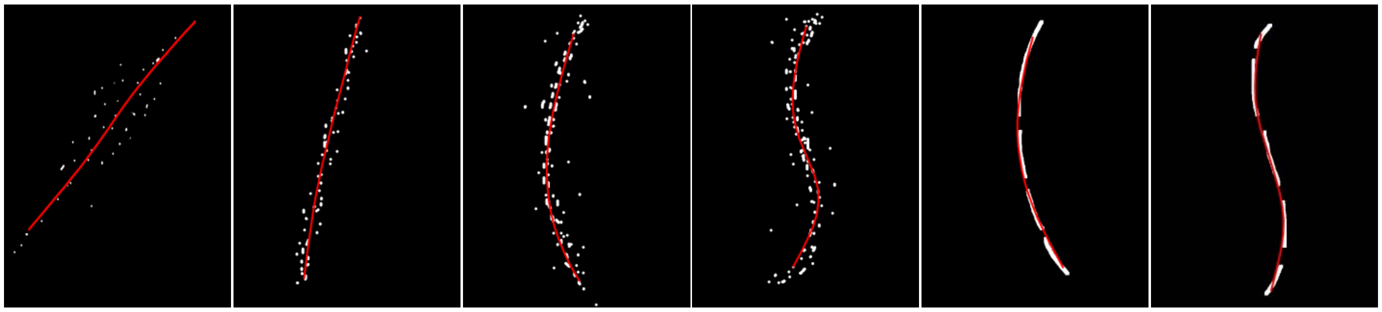


Figure 12. Test for RANSAC B-spline fitting using scatter diagrams (write pixels are candidate points and the red curves are fitting results).

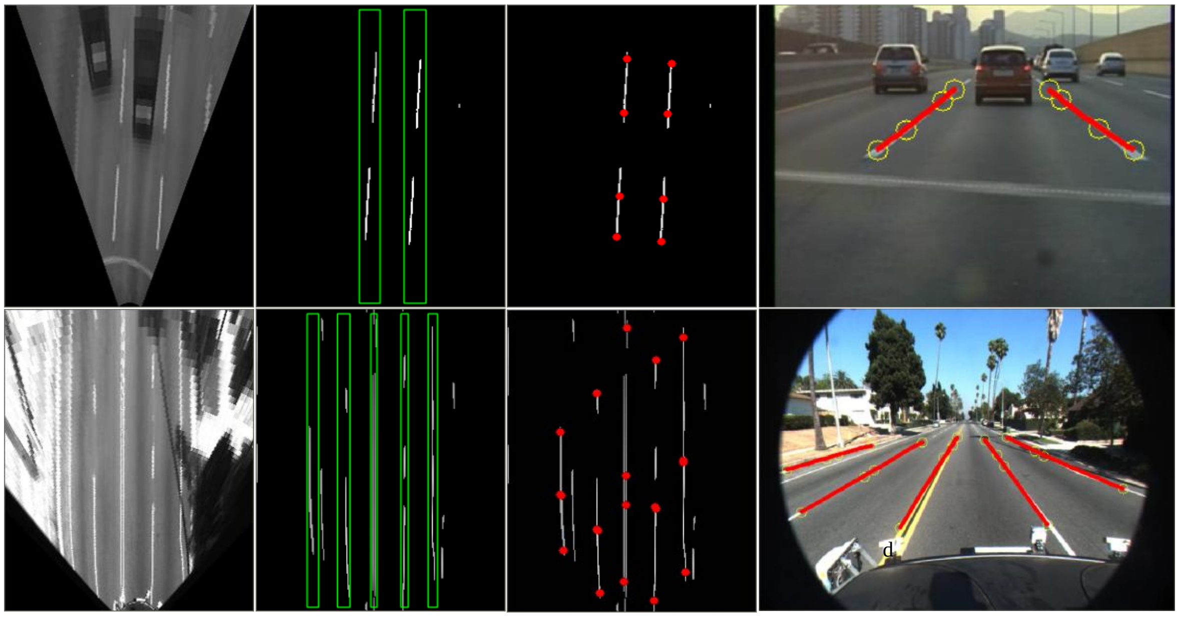


Figure 13. Test for RANSAC B-spline fitting using real lane-marking frames: (a) IMP image, (b) lane marking boundary image, (c) feature point using the optimized RANSAC algorithm, (d) resulting image of the inverse transformation of IPM.

3. EXPERIMENT AND ACCURACY EVALUATION

We ran the algorithm to detect only two lanes and all lanes of the current frame, respectively, and the detected results of four test video clips are shown in Figure 14 and Figure 15. We conducted the experiment using a typical desktop PC with Intel Core (TM) 2 2.0 GHz, RAM 3.25G and the pixel size of two original test videos are 320 by 240 and 640 by 480 respectively. As illustrated in Table 2, the processing time for the video frames was in accordance with the requirements of real-time systems.

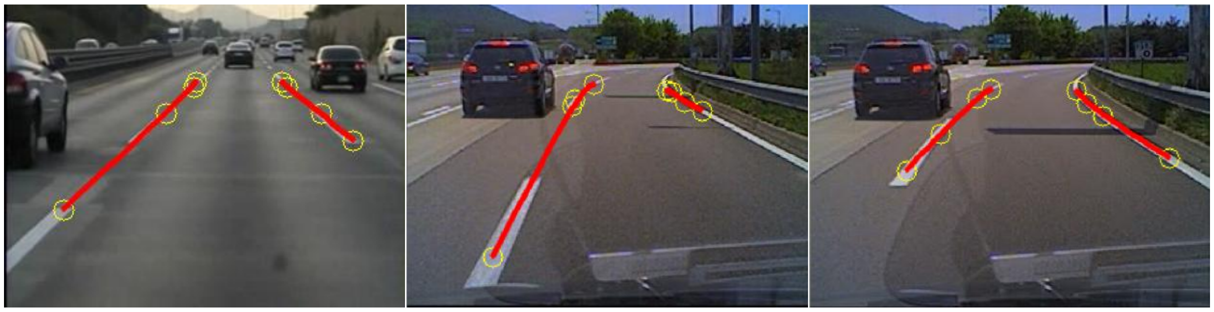


Figure 14. Resulting images for two-lane detection.

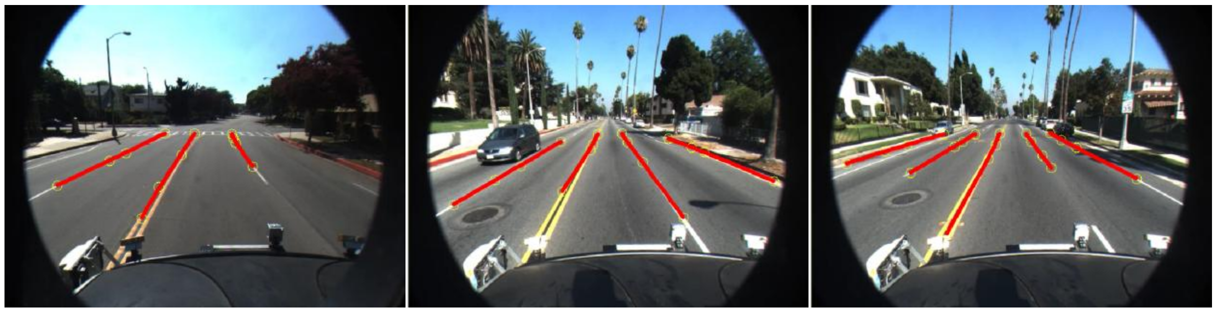


Figure 15. Resulting images for the detection of all lanes.

Table 2. Processing time for one frame

Test video	Pixel size of frame (pixel)	Frames	Minimum of processing time (ms)	Maximum of processing time (ms)	Average of processing time (ms)
Seoul_01.avi	320×240	571	31	59	32.32
Washington_2.avi	640×480	231	33	72	41.64

We collected a number of clips on different types of urban streets, with shadows and vehicles ahead, and using straight and curved streets. Unlike previous papers that just mentioned the rough percentages of detection rates, and in order to obtain an accurate quantitative assessment of the algorithm, we hand marked all visible lanes in two different test videos, totaling 1,746 sample points, as shown in Table 3.

Table 3. Sample collection

Test video	Frames	Lanes	Sample points
Seoul_01.avi	159	285	854
Seoul_02.avi	174	296	892

Our evaluation system was written using C++/MFC based on the OpenCV open source library. Figure 16 shows the interface of our simple point collection system.

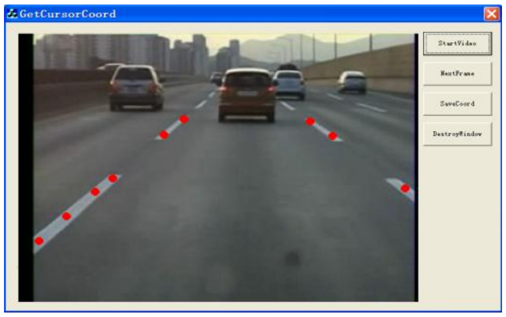


Figure 16. Interface of the sample point collection system.

For our sample point collection system, it is necessary to click on lane marking on the frame plane, and the pixel coordinates of clicked location will be saved automatically to a customized .txt or Excel file. The attributes of a point include frame number, point number, lane number, and u and v coordinates. The description above represents the first step in accuracy evaluation.

Next, we load the sample point file and the distance from every sample point to the corresponding fitted RANSAC spline; if the distance is less than our threshold, which is defined according to a specific pixel width for lane marking in the test video—for instance, the threshold is 3 pixels if the width of lane marking equals 6 pixels—then the corresponding point on the spline is correct.

Finally, statistical data are recorded for the correct point number and accuracy equaling the number of correct points divided by the total sample point number; the accuracy of our experiment is shown in Table 4.

Table 4. Accuracy evaluation

Test video	Sample points	Correct point	Accuracy rate
Seoul_01.avi	854	793	92.857%
Seoul_02.avi	892	838	93.946%

4. CONCLUSION

In this paper, we proposed an efficient, real-time, robust algorithm to detecting lanes on urban roads. The algorithm was based on taking a top view of the road image, filtering with Gaussian high pass kernels and a robust denoising template, then using Hough transform, a line classifier, a Kalman filter, and an optimized RANSAC three-degree Bezier spline fitting technique to detect lane marking on the road.

We compared the performance characteristics of our optimized algorithm, such as efficiency and precision, to others that usually just can run on highways or with a few vehicles and shadows. In contrast to these previous works, our algorithm can detect lanes in various conditions, achieve better accuracy, and spend less time doing so.

5. ACKNOWLEDGMENTS

This research was also supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2013-H0301-13-2006) supervised by the NIPA(National IT Industry Promotion Agency). And this research was also supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2012M3C4A7032182)

6. REFERENCES

[1] Peden, M., Scurfield, R. Sleet, D., Mohan, D., Hyder, A., Jarawan, E. and Mathers, C. 2012. Decade of Action for Road Safety 2011-2020. World report on road traffic injury prevention. (May. 2013).

[2] Harry, Lum. and Jerry A, Reagan. 1995. Interactive Highway Safety Design Model: Accident Predictive Module. Public Roads Magazine. 59, 2(winter, 1995).

[3] Aly, Mohamed. 2008. Real time Detection of Lane Markers in Urban Streets. IEEE Intelligent Vehicles Symposium. 10, 1109/IVS.2008.4621152 (2008), 7-12.

[4] ZuWhan, Kim. 2008. Robust Lane Detection and Tracking in Challenging Scenarios. IEEE Transactions on, Intelligent Transportation Systems. 9, 1 (2008), 16 - 26.

[5] Yifei, Wang., Naim, Dahnoun. and Alin, Achim. 2012. A novel system for robust lane detection and tracking. Signal Processing. 92, 2 (February, 2012), 319-334.

[6] Huarong, Xu., Xiaodong, Wang., Hongwu, Huang., Keshou, Wu. and Qiu, Fang. 2009. A fast and stable lane detection method based on B-spline curve. Computer-Aided Industrial Design & Conceptual Design. 10, 1109/CAIDCD.2009.5375392, (2009),1036 - 1040.

[7] Li, Bai. and Yan, Wang. 2011. Road tracking using particle filters with partition sampling and auxiliary variables. Computer Vision and Image Understanding. 115, 10 (October, 2011), 1463-1471.

[8] Xiaodong, Miao., Shunming, Li. and Huan, Shen. 2012. On-Board Lane Detection System for Intelligent Vehicle Based on Monocular Vision. International Journal on Smart Sensing and Intelligent System. 5, 4 (December, 2012).

[9] Chuang, Wang. and Zhong-ke, Shi. 2011. A Novel Traffic Stream Detection Method Based on Inverse Perspective Mapping. Procedia Engineering. 29 (2012)1938-1943.

[10] GREG, WELCH., GARY, BISHOP. 2006. An Introduction to the kalman filter. Department of Computer Science University of North Carolina at Chapel Hill Chapel Hill. NC27599-3175 (2006).

[11] Thomas, Veit., Jean-Philippe, Tarel., Philippe, Nicolle. and Pierre, Charbonnier. 2008. Evaluation of Road Marking Feature Extraction. Intelligent Transportation Systems. 10.1109/ITSC.2008.4732564 (2008), 174-181.

[12] Yue, Wang., Eam Khwang, Teoh. and Dinggang, Shen. 2004. Lane detection and tracking using B-Snake. Image and Vision Computing. 22, 4 (April, 2004), 269-280.

[13] Obradovic, D., Konjovic, Z., Pap, E., Rudas, I. J. 2013. Linear fuzzy space based road lane model and detection. Knowledge-Based Systems. 38 (January, 2013), 37-47.

[14] Yue, Wang., Dinggang, Shen., Eam Khwang, Teoh. 2000. Lane detection using spline model. Pattern Recognition Letters. 21, 8 (July, 2000), 677-689.

[15] Sio-Song, Ieng., Jean Philippe, Tarel., Raphael, Labayrade. 2003. On the design of a single lane-markings detectors regardless the on-board camera's position. IEEE Intelligent Vehicles Symposium. 10.1109/IVS.2003.1212974 (2003), 564 – 569.

[16] Jyun-Guo, Wang., Cheng-Jian, Lin., Shyi-Ming, Chen. 2010. Applying fuzzy method to vision-based lane detection and departure warning system. Expert Systems with Applications. 37, 1 (January, 2010), 113-126.

[17] Li, Bai., Yan, Wang., Michael, Fairhurst. 2007 An extended hyperbola model for road tracking for video-based personal navigation. Knowledge-Based Systems. 21, 3 (April, 2008), 265-272.

[18] Shengyan, Zhou., Yanhua, Jiang., Junqiang, Xi., Jianwei, Gong., Guangming, Xiong. and Huiyan, Chen. 2010. A novel lane detection based on geometrical model and Gabor filter. IEEE Intelligent Vehicles Symposium (IV). 10.1109/IVS.2010.5548087 (2010), 59 - 64.

[19] Nuthong, C., Charoenpong, T. 2010. Lane detection using smoothing spline. Image and Signal Processing (CISP). 2, 10.1109/CISP.2010.5646935 (2010), 989 – 993.

[20] Lopez, A., Canero, C., Serrat, J., Saludes, J., Lumbreras, F., Graf, T. 2005. Detection of lane markings based on ridgeness and RANSAC. Intelligent Transportation Systems. 10.1109/ITSC.2005.1520139 (2005), 254 – 259.

[21] Kunsoo, Huh., Jaehark, Park., Daegun, Hong., Dongil, Dan., Cho, Jahng., Hyon, Park. 2005. Development of a vision-based lane detection system considering configuration aspects. Optics and Lasers in Engineering. 43, 11 (November, 2005), 1193-1213.

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

[23] Massimo, Bertozzi., Constantin, Schimmel., Rudiger, Dillmann. 2007. Road-marking analysis for autonomous vehicle guidance. 19, 21 (September, 2007).

[24] H. G, JUNG., Y. H, LEE., H. J, KANG. 2009. Sensor fusion-based lane detection for LKS + ACC system. International Journal of Automotive Technology, 10, 2 (2009), 219−228.

[25] AIDALA, VINCENT J. Kalman behavior in bearings-only tracking applications. IEEE Trans. On AES. 15, 1, (Jan, 1979), 29-3.

[26] Stefan, Vacek., Alberto, Broggi., Alessandra, Fascioli. 1998. Stereo inverse perspective mapping: theory and applications. Image and Vision Computing. 16 (1998), 585-590.