International Journal of Automotive Technology, Vol. 7, No. 2, pp. 227−235 (2006)

1229−9138/2006/026−15

STABLE AUTONOMOUS DRIVING METHOD USING MODIFIED OTSU ALGORITHM

D.-E. LEE[1], S.-H. YOO[1] and Y.-B. KIM[2]*

[1]Graduate School of Mechanical Engineering Department, Chonnam National University, Gwangju 500-757, Korea

[2]Department of Mechanical Engineering, Chonnam National University, Gwangju 500-757, Korea

(Received 29 May 2005; Revised 13 October 2005)

ABSTRACT–In this paper a robust image processing method with modified Otsu algorithm to recognize the road lane for a real-time controlled autonomous vehicle is presented. The main objective of a proposed method is to drive an autonomous vehicle safely irrespective of road image qualities. For the steering of real-time controlled autonomous vehicle, a detection area is predefined by lane segment, with previously obtained frame data, and the edges are detected on the basis of a lane width. For stable as well as psudo-robust autonomous driving with "good", "shady" or even "bad" road profiles, the variable threshold with modified Otsu algorithm in the image histogram, is utilized to obtain a binary image from each frame. Also Hough transform is utilized to extract the lane segment. Whether the image is "good", "shady"or "bad", always robust and reliable edges are obtained from the algorithms applied in this paper in a real-time basis. For verifying the adaptability of the proposed algorithm, a miniature vehicle with a camera is constructed and tested with various road conditions. Also, various highway road images are analyzed with proposed algorithm to prove its usefulness.

KEY WORDS : Modified otsu algorithm, Stable autonomous driving, Image processing, RC vehicle, Real-time driving, IDF (Intensity Distribution Function)

1. INTRODUCTION

As one of the advanced traffic control systems, autonomous vehicle, irrespective of road conditions, has been studied within the various countries. For example ITS (Intelligent Transportation System) projects such as AVCS (Advanced Vehicle Control System), PROMETUEUS, ASV (Autonomous Safety Vehicle), and ITSK have been progressed in USA, Europe, Japan, and Korea, respectively (Behringer *at al.*, 1998; Turk *et al.*, 1998; Smith *et al.*, 1994). Intelligent transportation system could be classified as an advanced highway system (AHS), car navigation system (CNS), and an advanced vehicle control system (AVCS). Among them, an advanced vehicle control system (AVCS) renders the vehicle more intelligent that offers a lot of driving information to the driver, therefore, it can be utilized to help a driver in preventing unexpected dangers or accidents. In order to recognize environmental driving conditions for a safe driving, various types of sensors are generally required. Up to date, a vision sensor has been considered as one of the most adaptable sensors, which has the following advantages: (1) extra sensors are not required, i.e. it could be utilized without extra equipment for an autonomous driving, (2) it could offer the driver relatively wide range of front view, and (3) it could give very accurate data in the near-sight view.

In order to drive a vehicle autonomously using the image information obtained by a vision sensor, various methods have been studied and proposed such as fuzzy inferring method for detecting the lane edges (Li *et al.*, 1996). Also Taylor *et al.* (1997) used a stereo vision, Pomeleau *et al.* (1992) used a neural network system named ALVINN (Autonomous Land Vehicle in a Neural Network), Turk *et al.* (1998) used color CCD camera, and Bhringer (1995) used a Kalman filter to track road lanes. These researches are mainly focused to extract vehicle heading direction using various edge operators. Also, most of researches are not concerned about the road noise such as traveling with various lighting and sometimes lane occlusion, which is a usual case in real road conditions. Also their works are primarily concerned with the edge detection algorithms rather than the possibility of real vehicle control. Lee *et al.* (2002) used Leveberg-Marquardt neural network algorithm to extract the lane mark with results of the real-time vehicle control. Although they presented the possibility of the real-time autonomous vehicle with a test vehicle, they did not mention the road qualities, and their methods also

**Corresponding author.* e-mail: ybkim@chonnam.ac.kr

required long simulation time.

Meanwhile, other researchers were studying to pre-decide the road quality with various methodologies. Most of researches used Hough transform or approximated curve fitting method to obtain road lanes from image data and this method is widely used until today. Kreucher *et al.* (1999) detected the characteristics of lane marks in the frequency domain and it offers successful lane mark results regardless of varying light environment or lane occlusion. However, their researches had still problems with respect to noise such as pseudo-lane marks or apparent boundaries, reflection of sunlight and ambiguous boundaries, which made the system performance unfavorable.

Recently, Yi *et al.* (2005) utilized fuzzy neural network-based decision of road image for extracting lane related information in order to pre-determine "good", "shady" and "bad" image. Using fuzzy neural network-based decision, more reliable road images could be determined for further process. However, they did not mention the possibility of the real-time steering control application without experimental results. Also, it seems very complicated in applying to a real-time vehicle control with neuro fuzzy method even though it gives excellent results in deciding the road lane conditions.

In this paper we propose the modified Otsu algorithm to extract a lane mark regardless of a road image quality, i.e. "good", "shady" or "bad" with a relatively simple idea. Originally, Otsu algorithm utilizes an inflection point with a parabola of IDF (Intensity Distribution Function), a function of edge histogram, to identify a lane mark (Otsu, 1979; Choi, 1997). Otsu algorithm is very well applicable for the road which has high quality image with only having two image boundaries, i.e. two distinct foreground and background images. However, this algorithm could not be used for a real-time controlled autonomous vehicle because it can provide wide view image information, resulting in containing unnecessary image data. Therefore, in order to reduce the calculation time for a real-time vehicle control, shrinking lane mark area from the whole view is necessary. Also, real road always has not prefect conditions, i.e. it has many IDF peaks, which representing true lane mark is buried with background noise images. Therefore, it is necessary to modify this algorithm to extract real lane one from unnecessary images. The proposed algorithm is the same as Otsu algorithm in finding the inflection point, but it can have various threshold IDF value decided *in priori*, to extract true lane mark from "shady" or "bad" road images. Its algorithm is not sophisticated and it could be applied easily for the control of real-time autonomous vehicle.

The proposed algorithm was tested with miniatured car with a vision system, and it was verified that the algorithm works very well in harsh road conditions, i.e. night road condition, concrete road, asphalt road, and the occluded road due to a proceeding vehicle. The proposed algorithm was applied with conventional PC based hardware to prove it could be applied to the real world vehicle with rather simple hardware systems.

2. MODIFIED OTSU ALGORITHM FOR EXTRACTING LANE MARK

The speed of image processing is highly dependent on the hardware, which means the speed of a real-time calculation is decided upon which hardware (CPU board, Grabber board and Camera, etc.) is utilized. However, in this paper, conventional PC based vision system was utilized. The purpose of using conventional hardware system is to verify that the proposed lane mark detection algorithm could be applicable to most of the vision system well, once it could be effective with simple system. The proposed lane mark detection flow chart from the road image to calculate steering angle of the vehicle is represented as shown in Figure 1.

2.1. Decision of Searching Area with Linear Equation Representing Lane Mark

It is generally assumed that a linear equation, which represents the true lane mark, does have a similar pattern and almost same decline angle between previous image

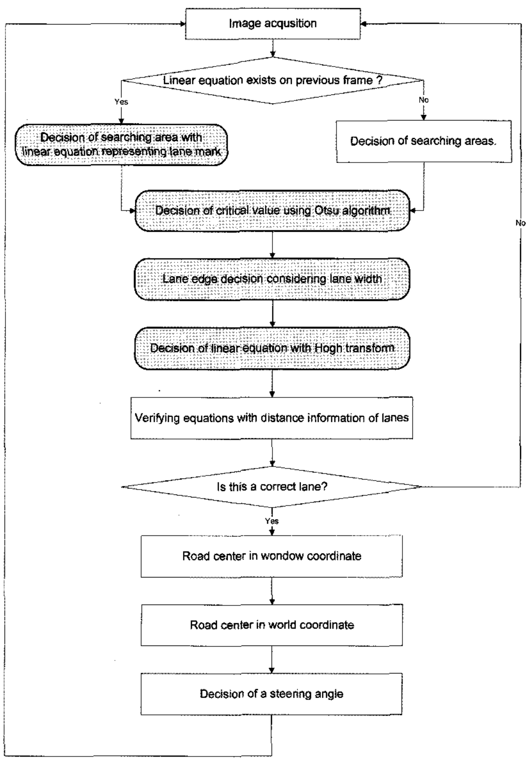


Figure 1. Lane detection algorithm.

and present one. If the vehicle is controlled in real-time, the minimum image processing time should be less than 33 *ms*, which implies, during this short period time, the vehicle does not move a long distance and it does not experience rapid road curvature variation. Therefore, it is predictable to locate the specific area, from present whole image, where the lane mark exists by utilizing a linear equation obtained from the previous image. If the lane mark is assumed to have a linear equation form, the result of this equation using Hough transformation has the following shape

$$x \sin\theta + y \cos\theta = \rho \quad (1)$$

in which, θ represents the orientation of an edge, x, y stands for displacements of x, y coorinate in image plane, and ρ is a constant. In equation (1), if one moves the lane mark to other location with the constant distance, then equation (1) can be represented as in equation (2).

$$x \sin\theta + y \cos\theta = \rho \pm width \quad (2)$$

in which constant distance is expressed as a "*width.*"

Therefore, if there is an assumption that the vehicle does not change its heading direction rapidly, the image area surrounded by above two linear equations could be the location where the lane mark exists. The main objective with a proposed idea, by narrowing the searching area represented with two linear equations, is to reduce lane mark extraction time drastically. There still remains the possibility of regarding the wrong marks as a real one, for example, when the letter on the road or lane change mark exists. In these cases, possible error could be eliminated *in priori* by modifying the searching algorithm wisely to render the proposed method extracting a "true" lane. Therefore, more robust and safe autonomous vehicle driving is possible.

2.2 Digitization using Modified Otsu Algorithm

In extracting the lane mark from the obtained IDF, which has a convex parabola shape as shown in Figure 3, the inflection point is the one which clearly separate two areas.

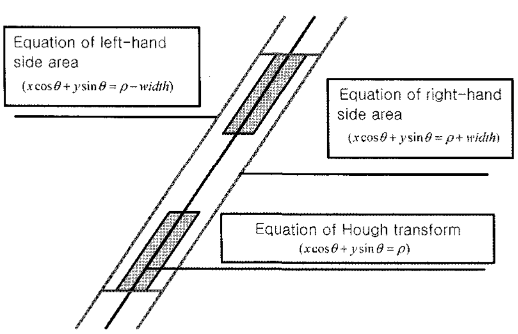


Figure 2. Detection area.

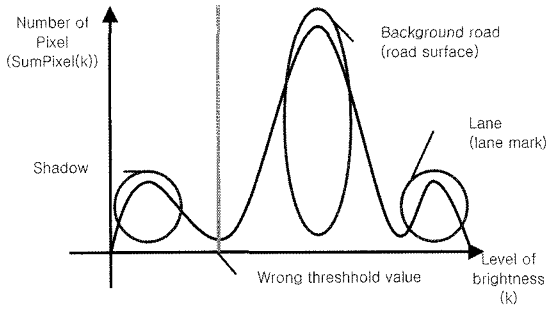


Figure 3. Histogram of general road image.

As shown in this figure, the histogram clearly shows that convex parabola represents the black road with lower intensity value and white or yellow lane mark with higher intensity one. Otsu algorithm is utilized to discern exact inflection point between two areas of each classes having different intensities (Otsu, 1979). As the algorithm always divides all the images with two distinct intensity areas, it gives very fast and accurate lane mark extraction performance in the case where foreground and background intensities are distinctly separable. Although this algorithm offers an easy way to discern a lane mark only with a near-perfect road condition, it is not directly utilized for a real-time controlled autonomous vehicle, because the whole image has tremendous data to be processed. Therefore, an area reduction technique, searching only for a lane mark, should be considered first. Also, Otsu algorithm is not directly applicable, since all the paved road are not built by asphalt only, which means, more sophisticated road conditions should be considered more carefully for extracting the lane mark. As some paved road is composed from cement, the road and lane mark could not be easily distinguishable because background road is grey color, which has high intensity value. Therefore, it is difficult to identify where the lane mark exists because many parabolas with high intensity functions are present. If the cement road has the shade, there might exist a case having three different intensity areas, i.e. one representing the road itself, which has the intensity value of 100, one standing for the shade having the almost zero intensity, and the lane mark having the bright intensity with value of near 255. The typical case as explained is shown in Figure 3, and it shows that there are more than two convex parabolas instead of one. In general, the lane mark, usually has bright color having very high intensity, should be located at the inflection point of a right-hand area in the intensity histogram, but in applying the Otsu algorithm in the cement road case, it happens that the inflection point located at the left convex parabola is decided as a threshold for a lane mark searching area. According to the Otsu algorithm, the left-hand side area should be assumed to be a searching area, in this case all the pixels of roads and lane marks are mutually coexist. By assigning the intensity value of 255 for both road and lane, it is impossible to separate the exact lane mark successfully.

In order to improve this difficulty, the present method is proposed, i.e. firstly, search the point where its intensity has the largest pixel numbers. This point is considered as a threshold in IDF. Secondly, the inflection point is determined at the right-hand area of the threshold. By applying this approach, searching area could be reduced drastically. However, the reduced area still contains a lane mark as well as its background ones in "shady" or "bad" images. Therefore, the intensity, having the largest pixel numbers, could be picked up as a *representative* of a background image. By doing so, it can drastically reduce the searching area in comparison with conventional searching schemes, i.e. standard Otsu algorithm, because it only searches the area located at the right-hand side of a threshold. If the intensity number of the background is larger than the number of a lane mark, the largest intensity number is considered as the *representative* of background intensity. Therefore, instead of applying the Otsu algorithm with the whole area which has the intensity distribution from 0 to 255, the modified algorithm extracts the lane mark by searching the area surrounded from the *representative* intensity to the point where the intensity is 255. The proposed algorithm could be explained with the following equation

$$SumPixel\ (k) = \begin{cases} 0 & ,\ if\ k < k_{SumMax} \\ SumPixel\ (k), & if\ k \geq k_{SumMax} \end{cases} \quad (3)$$

in which, *SumPixel*(*k*) represents the number of pixels with intensity k, and k_{SumMax} is the intensity value having the largest pixel numbers among all *SumPixel*(*k*).

As shown in Figure 4, the proposed method always choose the far right inflection point as a threshold, therefore, it could render rapid separation between a lane mark and a background road even the road image has lots

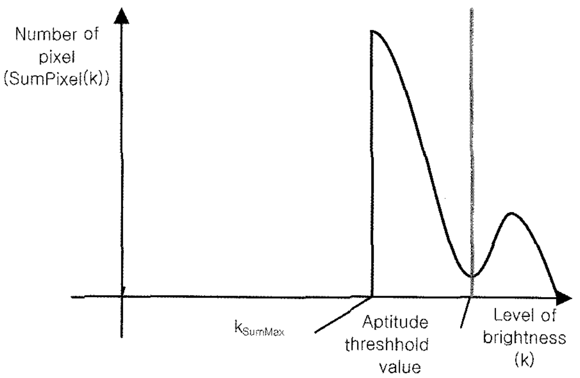


Figure 4. Modified histogram of general road image.

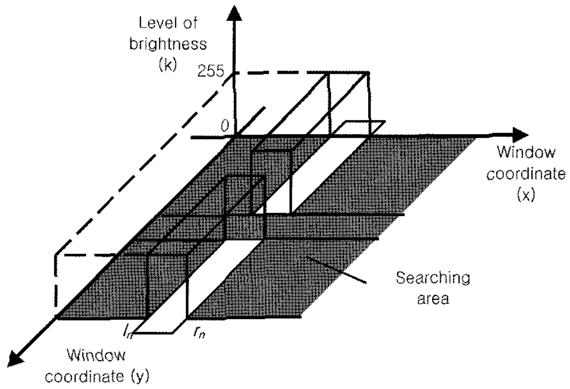


Figure 5. Coordinate system of lane position and brightness.

of noise.

2.3. Lane Center Decision Considering Lane Width

In general, calculation time increases as number of edge increases in doing Hough transformation. Therefore, extracting the edge with the first trial, if it is considered as a "true" lane mark, tremendously reduces transformation time in compared with the case where all the edges are transformed and searched. In this research, in order to reduce computational burden for Hough transformation by eliminating unnecessary edges, the number of Hough transformation is reduced by only performing the transformation where the edges have a lane mark pattern. In this paper the "edge pattern" represents the center line of the real lane mark with a certain width. Figure 5 shows "edge" locations and its intensity functions in x, y plane of the image.

As shown in the figure, a lane mark and a background are clearly separated. With assumption that the lane mark width is constant, the width, obtained from the distance made by two locations where binary value of intensity changes from 0 to 255 and the one where intensity drops from 255 to 0, is compared with real lane mark width. As the image is transformed to have a binary value, the lane mark has binary value of 255 and, the lane *width* has the consecutive intensity value of 255 along the x axis. Therefore, after finding each lines having consecutive 255 binary value in vertical axis, and assuming that the searched line has almost the same width of real lane one, the center of the width is selected as true center lane. If the variation of intensity value is represented by binary number along the x direction, the shape has the form as shown in Figure 6, which contains many step functions.

In the figure, if one define l_n as the point where intensity changes from 0 to 255 in the x axis, and r_n as the point where the intensity changes from 255 to 0, each step function has represented with the following form.

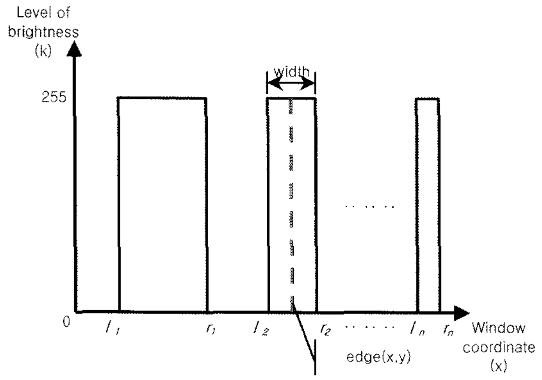


Figure 6. Histogram of binary image.

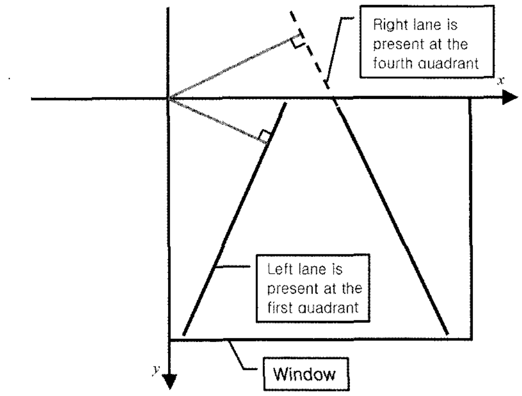


Figure 7. Left and right position in window coordinate.

$$u(x-l_n)\begin{cases} 0\ , if\, x<l_n \\ 255,\ if\, x\ge l_n \end{cases}$$
$$u(x-r_n)\begin{cases} 0\ , if\, x\ge r_n \\ 255,\ if\, x< r_n \end{cases} \tag{4}$$

If one presumes that the lane width is w, the possible lane width error is e, and there exists an edge having l_n and r_n satisfying the equation, the edge could be represented as

$$edge=\left\{(x,y)\middle|\left(\frac{l_n+r_n}{2},y\right)\right\} \tag{5}$$

In equation (5), as only one linear line is decided as a real lane mark, the first determined linear line is considered as a *representative* of a *real* lane mark. As the real lane is now found, the remaining other edges are discarded. Therefore, the proposed idea reduces edge detecting calculation time by eliminating other unnecessary edges.

In applying Hough transformation to extract a lane mark, first, size of the accumulator should be determined, and the size of the accumulator decides the calculation time. The factors to decide the size of accumulator are ρ and θ. ρ is obtained from all the edges located inside of θ, and the line is detected by accumulating ρ and θ. If every line detection time is assumed to be constant, only the number of lines and range of θ are the factors which determine the calculation time. Therefore, if the range of θ is reduced, the edge detection time is reduced accordingly.

The shape of two lane marks has the form of trapezoidal one if it is viewed from the center of a camera. Therefore, the θ of a left lane is located at the first quadrant, and the θ of a right lane is present at the fourth quadrant.

As shown in Figure 7, in general, the range of θ has the value between -90^0 and $+90^0$. In a real paved road, as the curvature of the road is not changed rapidly, the θ range could be reduced smaller, therefore, it is assumed, in this paper, that the range of θ is set to from -60^0 to $+60^0$. At the first process, the right and left searching areas are predetermined, and θ value of the accumulator is set to 60^0.

As ρ represents the distance between two lines, it could have the minimum value of 0, and the maximum one of the diagonal distance of the whole image screen. Therefore, as the range of ρ has the range of 0 $\sim\sqrt{x_{\max}^2+y_{\max}^2}$, and as the image size of 640×480 is utilized in the paper, the range of ρ in this case is between 0 and 800. In conclusion, the accumulator has two-dimensional array, i.e. the dimension of [60][800]. If the range of the lane mark is larger than maximum θ, it is excluded during Hough transformation process. Therefore, the proposed algorithm has the advantage to eliminate possible errors. It is manifest that the real lane mark has the largest number of edges. Therefore, during the imaging process, the candidate of the possible lane mark is selected, and among these candidates, the largest edge is chosen as a true lane mark.

2.4. Decision of a Steering Angle

In this research, in order to decide the vehicle steering angle for a lateral motion, the center point of each lane located on the image screen is transformed into the real world coordinate where the vehicle locates. During the transformation, three coordinate systems are introduced, i.e. (x_s, y_s) as a window coordinate, (x_e, y_e, z_e) as a camera coordinate, and (x_w, y_w, z_w) as a world coordinate. The screen coordinate is constructed from the road image obtained by a CCD camera. The origin of the screen coordinate is located at the center of the image, and the x_s coordinate is an axis parallel to the horizontal screen, and y_s is an axis vertical to it. The z_e axis of the camera coordinate is aligned with the optical axis of the lens, and x_e, y_e axes are defined mutually vertical to z_e axis. The world coordinate is determined according to the location

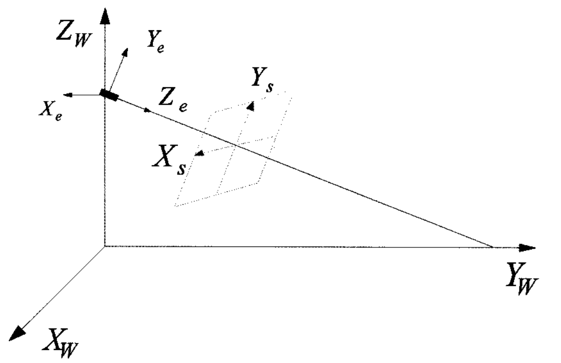


Figure 8. Coordinate system.

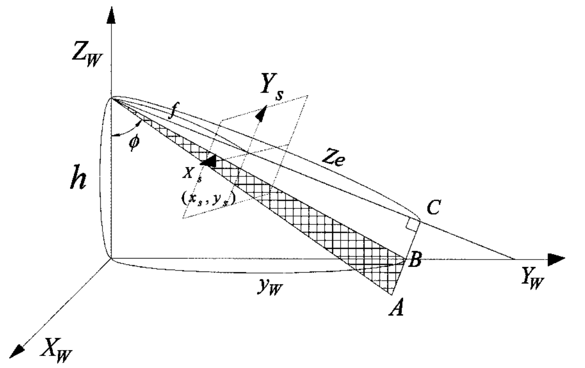


Figure 9. Coordinate system between a camera and a point in real world.

of the vehicle. The y_w is the axis parallel to the vehicle's forwarding direction, z_w is defined parallel to the height of the vehicle, and x_w is the axis parallel to the right-hand side of it. Figure 8 represents each coordinate relationship.

To determine the vehicle's steering angle, first, the point represented by the screen coordinate should be transformed into the world coordinate.

As shown in Figure 9, if the position of the vehicle in the screen coordinate is assumed to be (x_s, y_s), and one also assumes that the height of a camera is h and an incident angle Φ, with assumption that these are known values, then the next moving position of the vehicle at the world coordinate could be easily obtained. The next moving position is located at the floor, where the coordinate of z_w is zero. From equation (6), coordinate of y_w is easily obtained from the y_s coordinate as

$$\alpha = \tan^{-1}\frac{y_s}{f}$$
$$y_w = h\tan\{\phi - \alpha\} \tag{6}$$

Next, in order to obtain x_w in the world coordinate, the length of line segment AB as shown in Figure 9 should be calculated. It can be derived as shown in equation (7).

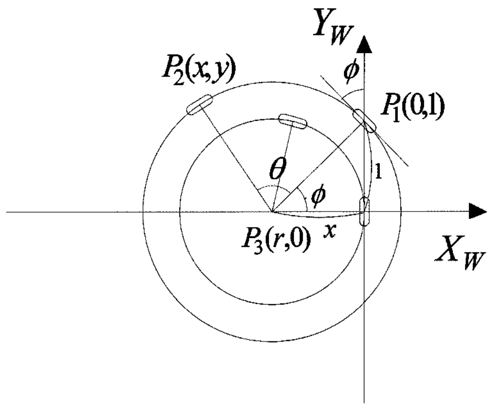


Figure 10. Vehicle trajectories.

$$x_w = \frac{\sqrt{h^2 + y_s^2} \times x_s}{\sqrt{f^2 + y_s^2}} \tag{7}$$

After deciding the target position of $P_2(x, y, 0)$, one should control the steering angle to arrive at the target point. In general, the vehicle steering angle is a function of multi-linked suspension geometry including non-linear tyre and bushing as well as link kinematics. However, in this paper, simple vehicle dynamic model is considered as shown in Figure 10.

The vehicle model has front and rear tires and only yawing motion is possible by excluding rolling and pitching motions. If the distance between front and rear wheel is assumed to be l, then the rear wheel is located at the inner circle as shown in Figure 10. If the front wheel is at P_1, then the vehicle body trajectory is assumed to make double circles with their centers P_3. As shown in Figure 10, the front wheel keeps the outer radius trajectory while rear one follows the inner radius one. If front tire is located at P_1 and its target is to P_3, then the steering angle Φ could be obtained using equation (8).

$$\sqrt{r^2 + l^2} = \sqrt{y^2 + (x - r)^2}$$
$$r = \frac{x^2 + y^2 - l^2}{2x} \tag{8}$$
$$\phi = \tan^{-1}\frac{l}{r}$$

3. EXPERIMENTAL RESULTS

3.1. Image Processing Results

First, various road images, taken from the highway with speed of 100 km/h, were tested for the adaptability and possibility for the real-time controlled autonomous vehicle using the modified Otsu algorithm. The tested road conditions include "good", "shady", and "bad" images.

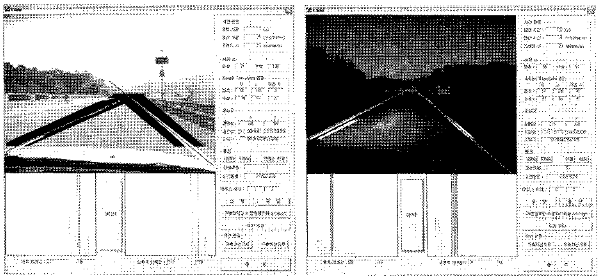
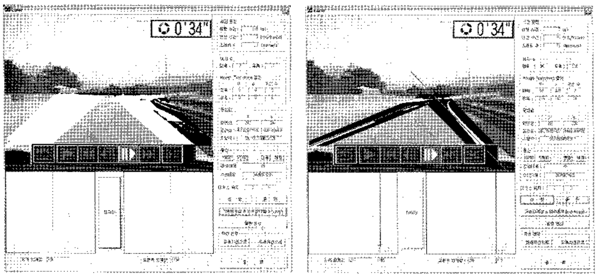


Figure 11. Image of daylight and nignt.



(a) Otsu algorithm **(b)Modified Otsu algorithm**

Figure 12. Comparison between original and modified Otsu algorithm.

For example, in order to apply with proposed algorithm, the images taken at daylight and night were tested and compared.

As shown in Figure 11, every frame has its variable threshold and they show a good stable lane mark tracking tendency regardless of the daylight or night condition. Specially, modified Otsu algorithm was applied and compared with one with regular Otsu algorithm in the daylight image. The results are represented as shown in Figure 12.

As the figures show, when the weather is bright, i.e. with very high intensity level in the image or when the concrete paved road exists, the regular Otsu algorithm did not discern the lane marks. However, by applying the proposed modified Otsu algorithm, the stable and accurate lane marks were clearly obtained. As shown in Figure 12(a), the histogram shows that the area which has the largest number of pixels is about the point where intensity is near 180, and it also shows almost equal distribution of the intensity at both sides of the peak point. In this case, the inflection point should be located at the right-hand side of a parabola. In reality, by applying normal Otsu algorithm, the inflection point was located at the left-hand side of a parabola, which renders most of the searching area having the intensity value of 255. However, by applying the modified Otsu algorithm, as shown in Figure 12(b), the most of the area, where it is located beneath of the largest number of intensity point,

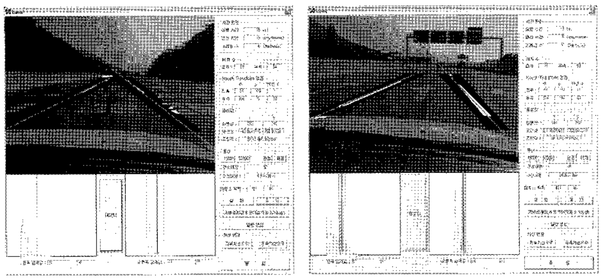


Figure 13. Image of road with mark.

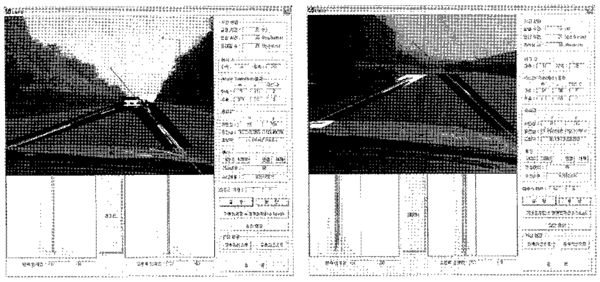


Figure 14. Image of road with another vehicle.

is totally eliminated by making its intensity value zero. Then, real inflection point is obtained resulting in a clear and stable distinction between concrete road and its lane marks. Further "shady" or "bad" images were under test with the proposed algorithm as shown in Figures 13 and 16.

As shown in Figure 13, where the road with some letters or marks are written between two lane marks, the algorithm shows the ability to identify the lane marks clearly.

Also as shown in Figure 14, when the traffic board is present or where the other vehicles are located in front of the vehicle, the proposed algorithm was tested. Both figures show the stable and robust lane marks detection results.

One of the worst case of the road quality is shown in Figure 15 and 16, which shows the road where the lane marks are not clear or most of a lane is nearly missing only with small remaining fractions. In these cases, using the limitation of the θ during Hough transformation and

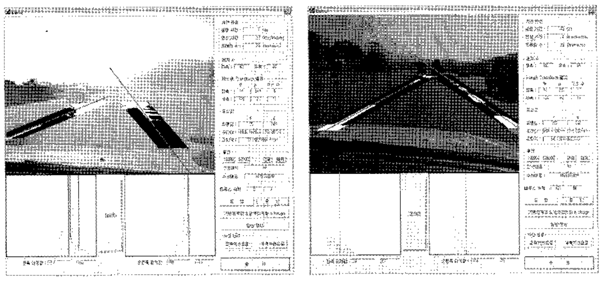


Figure 15. Image of road with unusual lane mark.

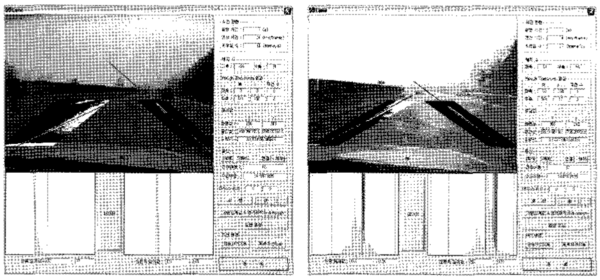


Figure 16. Image of road with unsteady region segmentation.

assuming a lane mark is a linear line having the largest number of edges, a lane mark is possibly detected with the proposed method. However, as shown in Figure 16 where the lane mark are not distinguishable, which are classified as "bad" or "occlusion" images, it was quite difficult to extract the lane mark even if there is no other vehicle ahead. Therefore, it is a topic of a further research for this "bad" or "occlusion" image case to control the vehicle autonomously and stably.

Table 1. Comparison between a real and test vehicle.

	Wheel base	Lane width	Lane distance	Wheel (diameter)	Vehicle speed (km/h)
Real vehicle	1,475	150	3600	590	80.901
RC vehicle	160	15.8	379	64	8.832

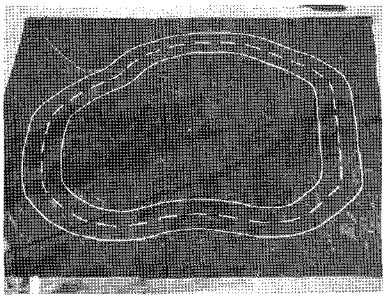


Figure 17. Heavily curved road case.

3.2. Model Vehicle Simulation Results

In the paper, for the sake of safety, a miniature vehicle was constructed for the possible usefulness of the proposed algorithm, and in order to reduce the computational burden, and the miniature vehicle was controlled by remote control using Brutus serial communication. The amount of steering angle for the model vehicle was decided according to the result of chapter 2.4, and the motion of the vehicle was activated by a DC motor with a PID control. In order to discern the lane mark, general desk-top PC with 2 GHz, 768 MByte was utilized with a graphic card of 128 MByte memory. MyVsion board from Microrobot Co. was utilized as an image grabber board. The size of 640×480 and 8 bits grey image was used for a vision image.

First, lane mark detection was verified from the continuous vision image taken from highway. Canon A80 digital camera was used for taking continuous vision image with size of 320×240, and its image has 15 frames at every second. For the worst case, if the vehicle dose not obtain one of the lane marks or totally miss the lane marks, i.e. occlusion case, due to the severe lane curvature or sudden obstacle is present, the previous steering angle obtained from the previous image frame was utilized and tested for a possible autonomous driving.

In order to compare with driving speed for the real autonomous vehicle, the size of the miniature vehicle and the real vehicle is compared as shown in Table 1. Lane mark thickness and its width are calculated and reduced accordingly. The estimated driving speed for a miniature vehicle was estimated with comparison of wheel radius ratio between a real vehicle and model one. As shown from the table, the model vehicle's driving speed was 8.8 km/h, which is approximated to the 80 km/h of a real vehicle. Two tests were performed to verify the possibility of the proposed algorithm and applicability of the real autonomous vehicle.

Firstly, a curved lane, with lane width of Table 1, was made with various brightness as external disturbances, which simulated "good" or "shady" road conditions. We tested different 20 cases with various light conditions from morning to the evening, and the proposed modified Otsu algorithm worked well to track the lane without a loss of the lane where the test length is almost 50 meters.

Secondly, heavily curved road was constructed as shown in Figure 17, which simulates the case where one of the lane marks was missing due to extreme curvatures. The total travel length is 17 meters, and 30 minutes were spent for each test to obtain stable autonomous driving. This simulation case shows that even if some of the lane marks are missed during a real-time processing, and though this is believed to have very rare case for a real road condition, the autonomous vehicle showed very nice lane tracking ability with steering information from modified Otsu algorithm. Even though the floor condition and lane marks are not the same as the real highway, it is safely concluded that the autonomous vehicle, with roughly 30 frames per second image processing speed, could control the vehicle up to 80 km/h speed with "good" or "shady" road conditions. If the test vehicle is equipped with the better devices, the faster autonomous driving could be possible. Even with half occluded road condition, the proposed algorithm worked well in tracking the lane marks.

4. CONCLUSIONS

In this research, more stable lane detecting algorithm for "good" or "shady" road condition with modified Otsu algorithm is suggested. Reduction of a lane searching area and detection of the edges with prior lane thickness information are presented. Also, the proposed modified Otsu algorithm, having each image frame with a varying threshold, renders more stable lane detection even in the case where previously announced algorithm does fail in detecting the lane with the "shady" images. Real road images with "good", "shady", and "bad" are tested with the proposed algorithm and the results show that it could offer a reliable tracking methodology. The test vehicle is constructed for proving the possible autonomous driving with applying the given method.

The conventional method, where the searching algorithm expands half of the left-hand area image, takes 38–42 *ms* for each frame with the hardware utilized in the paper. Its number of image processing is limited to have only 14 frames, which renders it very difficult to apply a real-time autonomous vehicle control. However, with proposed method, even searching all image areas, it only takes 16–26 *ms* per frame, and it gives us minimum 29 frames/sec searching speed. If there is no lane edge in the image, i.e. occlusion case, it takes only 16 *ms*, and if lots of edges are present, it takes 26 *ms* per frame during the Hough transformation.

The simulation results with the proposed method showed very stable and accurate lane detection ability with the various road conditions. The real-time controlled autonomous test vehicle, built for studying the adaptability of the proposed algorithm, showed that the proposed algorithm works well up to 80 km/h speed even the road has very "shady" condition or has very curved road profile.

REFERENCES

Behringer, R. and Muller, N. (1998). Autonomous road vehicle guidance from Autobahn to narrow curves. *IEEE Trans. Robotics and Automation* **14**, **5**, 810–815.

Behringer, R. (1995). Detection of discontinuities of road curvature change by GLR. *Proc. Intelligent Vehicle System,* 78–83.

Choi, K. J. (1997). A *Study of Strengthen the Image Method using Grey-level Approach.* M. S. Thesis, Yonsei University, Korea.

Lee, K. B., Kim, Y. J., Ahn, O. S. and Kim, Y. B. (2002). Lateral control of autonomous vehicle using levenberg-marquard method neural network algorithm. *Int. J. Automotive Technology* **3**, **2**, 71–77.

Li, W., Jiang, X. and Wang, Y. (1996). Rotation of an autonomous vehicle by fuzzy reasoning. *IEEE 5th Int. Conf.,* **1**, 246–250.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernet* **SMC, 9**, 62–66.

Smith, C. E., Brandt, C. A. and Papanikolopoulos, N. P. (1994). Vision sensing for intelligent vehicle and highway system. *IEEE Conf. MFI,* 784–791.

Taylor, C. J., Malik, J. and Weber, J. (1997). A real-time approach to stereopsis and lane-finding. *IFAC Transportation System,* **12**, 1102–1107.

Turk, M. A., Morgenthaler, D. G., Gremban, K. D. and Marra, M. (1988). VITS-A vision system for autonomous land vehicle navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence* **10**, **3**, 342–361.

Yi, U. K., Lee, J. W. and Baek, K. R. (2005). A fuzzy neural network-based decision of road image quality for extraction of lane-related information. *Int. J. Automotive Technology* **6**, **1**, 53–63.