Vehicle Classification by Road Lane Detection and Model Fitting Using a Surveillance Camera

Wook-Sun Shin*, Doo-Heon Song**, and Chang-Hun Lee*

Abstract: One of the important functions of an Intelligent Transportation System (ITS) is to classify vehicle types using a vision system. We propose a method using machine-learning algorithms for this classification problem with 3-D object model fitting. It is also necessary to detect road lanes from a fixed traffic surveillance camera in preparation for model fitting. We apply a background mask and line analysis algorithm based on statistical measures to Hough Transform (HT) in order to remove noise and false positive road lanes. The results show that this method is quite efficient in terms of quality.

Keywords: Vehicle Type classification, Road Lane Detection, Model fitting, Vanishing Point, Machine Learning

1. Introduction

An Intelligent Transportation System (ITS) obtains information from special purpose sensors installed inside the road and/or CC-TVs and analyzes it in order to make decisions to avoid abnormal traffic situations. It is easier to obtain information from special purpose sensors than CC-TVs, but it is more expensive and has less general applications since the information obtained from those sensors is limited to planned parameters such as the number of vehicles crossed, vehicle weights, and the velocity, etc., while information from CC-TVs can be used in more general situations if efficiently analyzed [1]. Thus, it is necessary to use video/image-processing techniques to obtain valuable information efficiently from fixed traffic surveillance cameras on the road [2]. However, they are very sensitive to natural phenomena such as wind, shadow, dirt, and fog, all of which create noise information. The sources of noise also include pedestrians, trees, and signal boards along the road. Thus, it is essential to develop noise-tolerant image processing algorithms when we use CC-TVs.

One of the basic but important algorithms in CC-TV image analysis is lane detection. It is a basic building block of many traffic analysis applications such as lane change detection and collision detection. There has been a great deal of research into lane detection for operating intelligent vehicles [3]–[4]. However, lane detection by traffic surveillance cameras contains more noises and smaller, less clear road lanes than those by intelligent vehicles, so the characteristics of algorithms are different from those of intelligent vehicles [5]-[6]. Various algorithms finding background models are available [7]-[9], and these are used in detecting lanes and/or in tracking objects in various ways.

Another interesting and important application building block is vehicle type classification [10]-[11] by surveillance cameras. In a traffic monitoring system, if a vision-based vehicle type classification is sufficiently accurate, it can be used in the analysis of traffic volumes and auto-toll systems, etc. Vision-based monitoring systems have a clear advantage over magnetic loop detectors in that they are less disruptive in terms of installation and have many other traffic parameters for further analysis. However, vision-based vehicle type classification is sensitive as regards how to model the vehicle object and how to resolve the shadow/occlusion effect [11].

In this paper, our main goal is to classify vehicles into four vehicle types from one fixed surveillance camera on a multiple lane road. Our vehicle classes are: cars, SUVs, trucks, and buses. In [10], the classes concern only cars and non-cars although, recently, and especially in Korea, some models of SUV are as small as ordinary cars. In [11], they tried to model 3-D coordinates from 2-D images and found that their estimates of the coordinates for buses, taxis, and SUVs were sufficiently correct. However, the real problem lies not in estimating the real coordinates but in finding the distinguishable attribute(s) over vehicle classes

Thus, we are following the same line of past research [10]-[11] in that we use 3-D model fitting from the vanishing point. In order to find a sufficiently correct vanishing point, we first detect the road lanes effectively [12].

For lane detection, we use the background model and Hough Transform (HT). However, HT in its basic form contains too many pixels so that it creates excessive lines with noise, and, consequently, the HT algorithm becomes slow. Thus, we try to decrease the number of pixels as the input of HT by applying sobel edge detection and thinning

Manuscript received January 27, 2006; accepted March 3, 2006.
This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).
Corresponding Author: Wook-Sun Shin
* Dept. of Computer Science, Konkuk University, Seoul, Korea ({wsshin,chlee}@konkuk.ac.kr)
** Dept. of Computer Games & Information, Yong-in Songdam College, Young-in, Korea (dsong@ysc.ac.kr)

first. Then, we apply a background mask to remove the pixels located in the non-road area. We also apply a line analysis algorithm after HT. Our line analysis algorithm examines the density of pixels and the distance between pixels from each candidate line produced by HT and rejects that candidate if it does not meet our statistical measure. This procedure successfully removes false positive lines from the HT output.

The detected lanes form a vanishing point which is used to model a vehicle in 3-D that is also useful in resolving vehicle occlusion and shadow elimination [13]-[14]. Then, we use machine-learning methods in vehicle type classification. Two well-known algorithms are used in our experiment. C4.5-rules [15] are quick and provide comprehensible if-then rules after learning. IBL [16], which is a variant of the nearest neighbor algorithm, can also be used in this problem.

The organization of the paper is as follows. In section 2, we explain the basic idea of HT and the other known algorithms used in our system as well as the algorithms developed by ourselves, such as the background mask and line analysis. Then, the overall structure of our system and model fitting methodology are explained in section 3, followed by our experiment results in section 4, ending with a discussion of the limitations and future steps of this research as a conclusion.

2. Related Algorithms

2.1 Background Estimation

The basic method of finding the background from an image requires the use of the accumulated data of each pixel from the first frame to the current frame. We take the most frequently appearing brightness as the background pixels from the accumulated data [6]. When (x,y) represents the coordinates of a pixel, the background pixels are obtained by the formula (1) below:

$$B_{N-1}(x,y) = Select\{I_k(x,y) \mid k = 0,\ldots,N-1\} \qquad (1)$$

where $B_{N-1}(x, y)$ is the estimated background pixel and $I_k(x, y)$ represents the k^{th} frame of N continuous images, and the function select means:

$$select\{I_k(x,y)\} = \max\{\rho_{x,y}(r) \mid 0 \le r \le 255\}$$

where $\rho_{x,y}(r)$ is the number of possible pixels in (x, y) and r denotes the brightness.

(a) (b) (c)

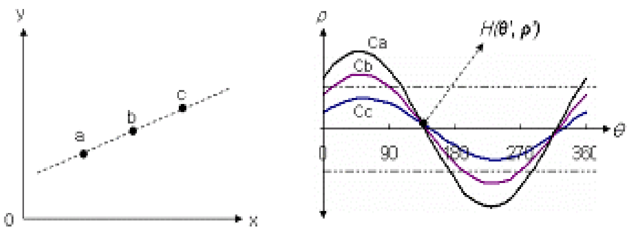
Fig. 1. (a) Background Image; (b) Edging of (a); (c) Thinning of (b)

2.2 Lane Detection by Hough Transform

We use sobel edge detection as a pre-process to extract lane information from the binarized background image. However, the result usually contains too many pixels. Thus, we apply a thinning algorithm that only finds the skeleton of the line. Thus, we can decrease the number of pixels without losing lane information [9].

2.2.1 Hough Transform (HT)

HT is frequently used to detect a line from an image. A line is mapped as an intersection point in the Hough area if it is translated to a coordinate system that has the relative rotation angle and distance from origin, i.e. the polar coordinate. Using this, HT transforms the Cartesian coordinates (x, y) into the polar (ρ, θ), and moves information from there and the retransform to the Cartesian coordinates system. If we take a straight line in the equation $y=ax+b$, we transform it to the polar coordinates using $\rho = x\cos\theta + y\sin\theta$.



(a) (b)

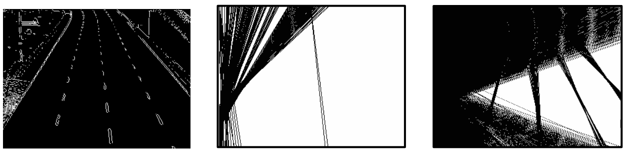
Fig. 2. (a) Cartesian coordinates; (b) Corresponding Polar coordinates

Then, as the above figure shows, every point on the same line (point a, b, c here) in the x-y Cartesian coordinate system is represented as a curve in the polar coordinate system and the intersection point of these curves denotes the line of the Cartesian coordinates. Thus the equation to find a line is expressed as follows:

$$Cross\,Lines\{H_{x,y}(\theta,\rho) \mid \rho = x\cos\theta + y\sin\theta, 0 \le \theta \le \pi\} > t$$

where t is the threshold of the number of curves crossed on that point.

HT is neat to find the lines from the image, but the algorithm is slow since it finds far more lines than necessary. In our experiment, that problem arises when we apply HT to our thinned images since the thinned images include much more unnecessary information than just the lanes. Furthermore, there is no clear way to discriminate lines consisting of road lanes from lines consisting of other objects in the image. As we can see in Fig. 3 below, the pure HT may not get enough "real" road lanes (Fig. 3 (b)) or finds too many false positives (Fig. 3 (c)). Thus, the application of HT in its original form is not adequate for the images from the traffic surveillance camera [4]. Thus, we develop a couple of methods to reduce the noise in order to compensate this deficit of the HT.

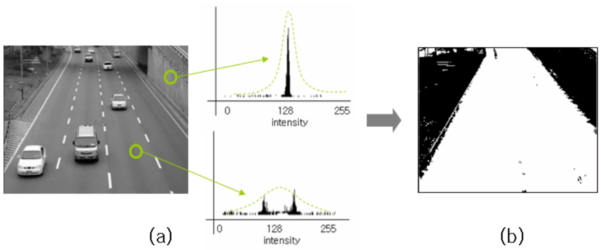


(a) (b) (c)

Fig. 3. (a) Input Image; (b) Output Lines at t=35; (c) Output Lines at t=17

2.2.2 Background Mask

For the process of background estimation, we use the accumulated data of each pixel from the first frame to the current frame. However, as shown in Fig. 4 (a), if the distribution of the intensity between pixels on the inside of the road and on the outside of the road is compared, there exists a clear distinction in the standard deviation. This is because there are more moving objects on the inside of the road than on the outside as time goes on. Thus, after taking the "AND" operation, Fig. 4 (b) shows that most non-ROI is removed from the HT input.



(a) (b)

Fig. 4. (a) Intensity Distribution in Road and Non-road; (b) Background Mask

Fig. 5 (b) shows the input of the HT after background masking and Fig. 5 (c) shows the resultant lines from the HT, which shows much clearer lane detection than without it (Fig. 3(c)).

(a) (b) (c)

Fig. 5. (a) Edged & Thinned Image; (b)After Background masking; (c) Lines after HT with Background

2.2.3 Line Analysis

Although much improved, Fig. 5(c) shows that background masking is far from satisfactory in removing false positives. The source of this error is largely due to the surviving points from the non-road region. After the HT process, we have enough information about the points that consist of a line in the HT, and there is another statistical difference between the false positives and the true positives.

In Fig. 6, lines A and B are the true positives and lines C and D are examples of false positives. Since the camera is fixed, the length of true positives are usually longer than that of false positives if we take the starting and ending points from the output line of the HT (compare l_1 and l_2). Thus, we remove lane candidates like l_2 by the formula below.

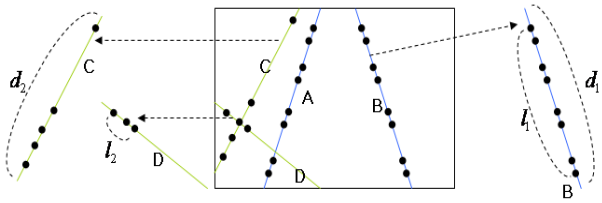


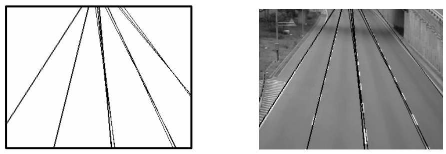
Fig. 6. Line Analysis

$$\{Max\,Length(SP, EP)\,|\,SP, EP \in \text{points}\,in\,Line\} > Lt$$

If the candidate satisfies the first criteria, as seen in line C of Fig. 6, the false positives line usually has greater standard deviation of the distances between points than that of the true positives. Thus, the second criterion for removing the false positive lane is as follows:

$$\frac{\sum_{k=0}^{n}\left(\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} - averageLength\right)^2}{n} < Dt$$

This shows that after 700 frames (after 50 seconds), the algorithm almost completes the background mask and there is little change after that. Both processes (Background Mask and Line Analysis) reduce the false positives significantly, but the best result comes from combining both processes, which cuts the number of candidates from about 5000 to 18 without missing the real lanes. Fig. 7 (a) shows the final result and Fig. 7 (b) shows it with the background.



(a) (b)

Fig. 7. (a) Result Lines; (b) Result Line Overlapped with Backgrounds

3. Suggested Method

3.1 Overview of the system

As discussed in section 2, the image after the edge detection and thinning process still includes many unnecessary pixels that are the main source of the false positive lines in HT algorithm. A large part of those noise pixels are from the outside of the target road of the image.

Thus, we create a background mask using the statistical measure explained in section 2.2.2, and that is then used to remove the outside region of the road from the image by the "AND" operation with a thinned image. Then, HT takes this ROI (Region of Interest) image as an input.

However, it still contains many false positive lines. Thus, we take the HT output as "candidate" lanes and apply another statistical measure explained in section 2.2.3 to remove the false positives (Line analysis process). After obtaining the road lanes, we find the vanishing point by intersecting the lanes. This vanishing point is used to model the moving object with the resultant image of the background subtraction techniques.

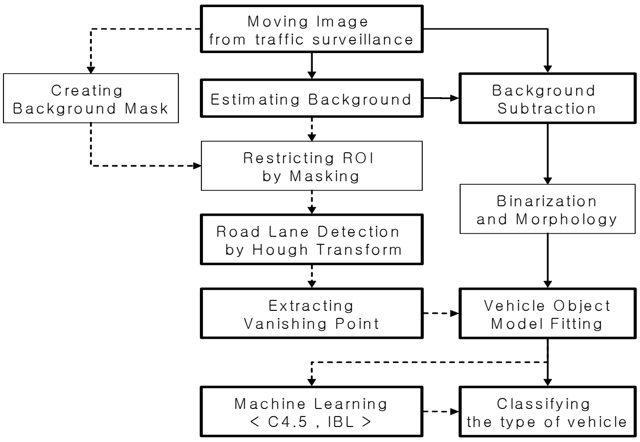
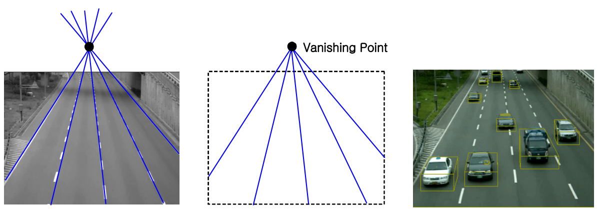


Fig. 8. Block Diagram of the System

This 3-D model fitting procedure will be explained in detail in section 3.3. Since our goal is to classify the vehicle types, the three attributes of the 3-D modeled objects–width, depth, height–are used in supervised learning. Then the result rules (C4.5) or the memorized instances (IBL) after training are used as the actual classification. This learning procedure and the experiment will be explained in section 4.

3.2 Vanishing Point (VP)

The VP is obtained by intersecting the road lanes. Although the VP can be obtained by other methods, we use intersecting road lanes since this method does not need any information about traffic volume or other vehicle information; and the road lanes should be found anyway in any ITS for different applications. However, the real lanes are not always straight lines, and sometimes the intersecting point is not determined as one point due to errors in finding the road lanes. Thus, we take the average of the intersecting points as a VP. Fig. 9 shows the procedure for finding the vanishing point.

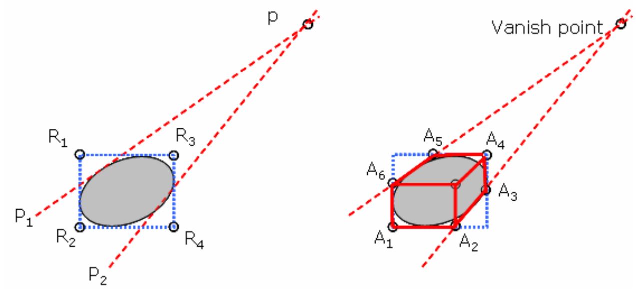


(a) (b) (c)

Fig. 9. Finding Vanishing Point (a) Intersecting road lanes; (b) vanishing point; (c) 3-D fitting by vanishing point

3.3 Model Fitting

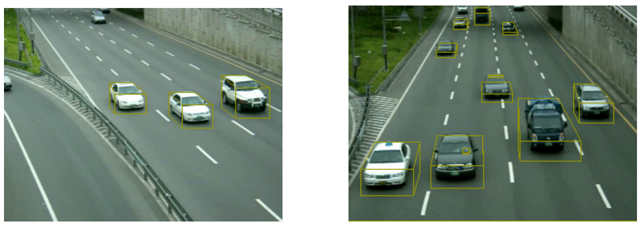
In Fig. 10 (a), we know where the VP is; the object found is represented in the gray circle. First, we fit the object using the rectangle R1~R4 and obtain the two tangent lines P1 and P2 from the VP to the object. If A1~A6 are from the VP to the object (P1 and P2) the goal vertices forming the 3-D model of the object, then from Fig. 11 (a) we can easily find that A1, A4 are the same points as R2, R3 respectively.



(a) (b)

Fig. 10. 3-D Model Fitting from a Vanishing Point

Moreover, A2, A3, A5, A6 are the intersecting points of: $\overline{R_2R_4}$ and $\overline{PP_2}$; $\overline{R_3R_4}$ and $\overline{PP_2}$; $\overline{R_1R_3}$ and $\overline{PP_1}$; $\overline{R_1R_2}$ and $\overline{PP_1}$ respectively. The equations are a little different according to the direction of the VP from the object (center, right, or left), but the way to obtain the six vertices is almost the same. Fig. 11 shows the results of model fitting in various ways.



(a) (b)

Fig. 11. Model Fitting Results; (a) Fitting from side view; (b) Fitting from the front view

4. Experimental Results

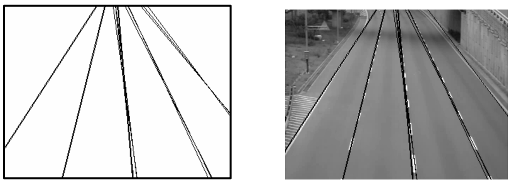
4.1 Road Lane Detection

Our experiment is designed for real color video daytime shooting (320x240, 15frame/s) of moderate traffic and has more than 5000 frames. There are only five real lanes in the video. Thus, we evaluate how many result lines are found and how many "real" lanes are found according to the algorithms when: pure HT is applied (HT); HT with a background mask (&Mask); HT with line analysis (&LA); and all of the processes combined. We also break down the results as the frame continues in order to determine when the algorithm plays stably. Table 1 shows the results.

Table 1. Lane Detection Experiment Results

Frame Number	HT	&Mask		&LA(Line Analysis)		&Mask &LA		Find Line /Totla Line	
200	5417	133	2.5%	1765	33%	6	0.1%	3/5	60%
300	4918	153	3.1%	1538	28%	13	0.2%	3/5	60%
400	4817	168	3.5%	1521	28%	10	0.2%	4/5	80%
500	4675	172	3.7%	1500	28%	15	0.3%	4/5	80%
600	4800	174	3.6%	1457	27%	12	0.2%	4/5	80%
700	5134	182	3.5%	1616	30%	19	0.4%	5/5	100%
800	5272	190	3.6%	1599	30%	18	0.3%	5/5	100%

This shows that after 700 frames (after 50 seconds) the algorithm almost completes the background mask and there is little change after that. Both processes (&Mask and &LA) reduce the false positives significantly, but the best result comes from combining both processes, cutting 5000 candidates to 18 without missing the real lanes.



(a) (b)

Fig. 12. (a) Result Lines; (b) Result Line Overlapped with Backgrounds

4.2 Vehicle Type Classification

In our classification experiment, we use two real video shootings of the same background. The first video is used as a training set and the other as a testing set. We have four vehicle classes - cars, SUVs, trucks, and buses. The distributions of the real vehicle types are shown in Table 2.

Table 2. Vehicle Type Distributions

Class	Training Set	Test Set
Type 1 (Cars)	33	37
Type 2 (SUVs)	45	34
Type 3 (Trucks)	3	1
Type 4 (Buses)	3	3
Total	84	75

The modeled vehicles are represented as four attribute-value pairs–width, depth, height, and the vehicle class. We used two learning algorithms. C4.5 [15] makes a decision tree and that tree can be translated into comprehensible if-then rules. However, it is expected that C4.5 will not be as efficient since this problem has only three available attributes without any salient features. Thus, we apply IBL [16], a type of nearest neighbor algorithm. IBL has four different versions, but in this experiment we can only use IB1, which has no noise filtering or selective storing features.

In the training video there were actually 90 moving objects, although six objects have not been recognized. This is mainly due to the occlusion effect and the presence of motorcycles that are too small in our labeling procedure. Table 3 shows the classification results of the two learning algorithms.

Table 3. Vehicle Type Classification Results

Class	C4.5			IBL		
	Correct	Error	Accuracy	Correct	Error	Accuracy
Type 1 (Cars)	33	4	89.2%	31	6	83.8%
Type 2 (SUVs)	23	11	67.6%	27	7	79.4%
Type 3 (Trucks)	1	0	100.0%	1	0	100.0%
Type 4 (Buses)	3	0	100.0%	2	1	66.7%
Total	60	15	80.0%	61	14	81.3%

The results show that the two learning algorithms are equivalent in terms of overall accuracy but have different characteristics in classifying class 1 and 2. If there were more training/testing instances in class 3 and class 4 (so that we can use IB3, which has a noise-filtering engine), the accuracy of the IBL would be better.

5. Discussion

With regard to traffic monitoring, vision-based vehicle type classification is one of the important functions and has many other applications including traffic volume analysis. Since this problem is sensitive as to how the vehicle object is modeled, we try to model moving objects in 3-D using a vanishing point. The vanishing point is obtained from the average of the intersecting points of the road lanes. Our road lane detection scheme is used to improve pure HT with background mask and line analysis algorithms based on statistics, while well-known machine-learning algorithms such as C4.5 and IBL are used for the vehicle type classification.

Although the road lane detection and 3-D model fitting appear to have been performed effectively, the accuracy of the classification is not as great as expected. This is mostly due to the small training samples (90 objects) in the four vehicle classes. Furthermore, we believe that the IBL family could draw a better result than C4.5 if the training set were to be enlarged, since we have only three attributes for use in this problem and the distribution of those three attributes is not clearly contrasted by vehicle type. However, our results are much more realistic than those of previous research in that we have four vehicle types rather than the two (cars/non-cars) reported in [10], and the estimation of the real 3-D coordinates as in [11] is not so important in our environment, which has one fixed surveillance camera for multiple lanes. However, in order to obtain a better result in the classification and 3-D model fitting, we should improve matters by resolving occlusion and diminishing the shadow effect, which were not however the main concern in this paper. Thus, our next effort will be focused in that direction.

References

[1] Hollborn, S., "*Intelligent Transport Systems (ITS) in Japan*", Technische Universitat Darmstadt, 2002.

[2] Palen. J., "*The need for surveillance in Intelligent Transportation Systems*", Intellimotion, vol. 6, no. 1, pp. 1-16, 1997.

[3] Kang; D.J., Choi; J. W., Kweon, I. S., "*Finding and tracking road lanes using line-snakes*", in Proceedings of the IEEE Intelligent Vehicles Symposium, pp.189–194, 1996.

[4] Jochem, T. M., Baluja, S., "*A massively parallel road follower*", in Proceedings of Computer Architectures for Machine Perception, pp.2–12, 1993

[5] Lai, A.H.S., Yung , N. H. C., "*Lane detection by orienttation and length discrimination*", , IEEE Transactions on Systems, Man and Cybernetics, Part B Volume 30, Issue 4, pp.539–548, 2000

[6] Lai, A.H.S.; Yung, N. H. C., "*A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence*" IEEE Circuits and Systems, Volume 4, pp.241–244, 1998

[7] Haritaoglu, I., Harwood, D., Davis, L. S., "A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance" IEEE Pattern Recognition Volume 4, pp.179–183, 2000 .

[8] Farin, D., de With, P.H.N. , Effelsberg, W., "Robust background estimation for complex video sequences", in ICIP Proceedings. Volume 1, pp.14–17, 2003

[9] Gonzalez, R. C., Woods, R. E., "*Digital Image Processing*." Addison-Wesley, 1992

[10] Gupte, S.; Masoud, O. Martin, R.F.K. Papanikolopoulos, N.P. "Detection and classification of vehicles", IEEE Transactions on Intelligent Transportation Systems, Volume 3, Issue 1, pp.37–47, 2002

[11] Lai, A.H.S. Fung, G.S.K. Yung, N.H.C. "Vehicle type classification from visual-based dimension estimation", in IEEE Proceedings of Intelligent Transportation Systems,. pp.201–206, 2001

[12] W. S. Shin, D. H. Song, C. H. Lee, "Road Lane Detection by Background Mask and Hough Transform" in Proceedings of APIS, Hangzhou, China, 2006

[13] Pang, C.C.C. Lam, W.W.L. Yung, N.H.C, "A novel method for resolving vehicle occlusion in a monocular traffic-image sequence", IEEE Transactions on Intelligent Transportation Systems, Volume 5., No. 3, pp.129–141, 2004

[14] Yoneyama, A. Yeh, C.H., Kuo, C.-C.J., "Moving cast shadow elimination for robust vehicle extraction based on 2D joint vehicle/shadow models", in Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance. 2003

[15] Aha, D.W., "A Study of Instance-Base Algorithms for Supervised Learning Task: Mathematical, Empirical, and Psychological Evaluations", Ph.D. Dissertation, University of California, Irvine, 1990

[16] Quinlan, J. R., "*C4.5: Programs for Machine Learning*". Morgan Kaufmann 1993

Wook-Sun Shin
He received a BS degree in Computer Science from Konkuk University, Korea, in 1998. He also received an MS degree in Computer Engineering from Konkuk University, Korea in 2000. He is now undertaking a doctorate course as a member of the Intelligent System Lab at Konkuk University. His current research interests include vision, image processing, and machine learning.

Doo-Heon Song
He received a BS degree in Statistics & Computer Sciencefrom Seoul National University and an MS degree in Computer Science from the Korea Advanced Institute of Science and Technologyin 1983. He received his PhD Certificate in Computer Science from the University of California in 1994. From 1983~1986, he was a researcher at the Korea Institute of Science and Technology. He has been a professor at the Department of Computer Games & Information, Songdam College, Korea, since 1997. His research interests include database, security problems, ITS, and game intelligence.

Chang-Hun Lee
He received a BS degree in Mathematics from Yonsei University, Korea, in 1980, and an MS and a Ph.D. in Computer Science from the Korea Advanced Institute of Science and Technologyin 1987 and 1993 respectively. From 1996~ 2000, he was the head of the Information Technology Center at Konkuk University. He has been a professor at the department of Computer Engineering, Konkuk University, Korea, since 1980. His research interests include artificial intelligence, operating system, embedded systems, and security.