Three-Feature Based Automatic Lane Detection Algorithm (TFALDA) for Autonomous Driving

Young Uk Yim and Se-Young Oh, *Senior Member, IEEE*

***Abstract*—Three-feature based automatic lane detection algorithm (TFALDA) is a new lane detection algorithm which is simple, robust, and efficient, thus suitable for real-time processing in cluttered road environments without *a priori* knowledge on them. Three features of a lane boundary—starting position, direction (or orientation), and its gray-level intensity features comprising a lane vector are obtained via simple image processing. Out of the many possible lane boundary candidates, the best one is then chosen as the one at a minimum distance from the previous lane vector according to a weighted distance metric in which each feature is assigned a different weight. An evolutionary algorithm then finds the optimal weights for combination of the three features that minimize the rate of detection error. The proposed algorithm was successfully applied to a series of actual road following experiments using the PRV (POSTECH research vehicle) II both on campus roads and nearby highways.**

***Index Terms*—Evolutionary algorithm, lane boundary candidate, lane detection, lane vector, road following.**

I. Introduction

LANE detection is crucial to vision-based lateral control as well lane departure warning for autonomous driving [1]. Since erroneous findings will generate wrong steering commands which may jeopardize vehicle safety, a robust and reliable algorithm is a minimum requirement. However, the great variety of road environments necessitates the use of complex vision algorithms that not only requires expensive hardware to implement but also relies on many adjustable parameters that are typically determined from experience. A good example is the threshold for binarization for which no fixed value exists for a variety of environments.

The AURORA system [1] applied a binormalized adjustable template correlation technique using downward looking cameras. However, it worked well only for slowly-varying roads. The LOIS system [2] used a deformable template approach in order to handle situations where the lane edges in an image have relatively weak local contrast, or where there are strong distracting edges due to shadows, puddles, pavement cracks, etc. The RALPH system [3]–[5], extracted a trapezoidal region of the forward looking road image by considering the vehicle's velocity, current visibility, and perspective effect obtained by straightening transform of the image intensity. This region, after being resampled, is then transformed into a plan-view road (viewed from the sky). The straightening transform is applied to this plan-view road image and the lanes are found from the intensity information of the transformed result.

The MOB-LAB land vehicle [6]–[10], transformed the perspective view image of the forward looking camera into a plan-view image. The lanes are detected from the result of binarization of this plan-view image assuming that the road is flat. The YARF system [11], [12] and the SCARF system [13] utilized color vision (from yellow/white lines) and semantic meaning of single and double lines for lane detection while Waxman [14] and S.G. Jeong [15] used a two-stage image processing procedure to enhance detection speed. The first stage spends some time to initially detect lane boundaries through fine search and the second stage will utilize the previous lane information to narrow down the search regions for increased speed and performance. Kluge [16], [17], and Nashman [18] took advantage of the histogram-based threshold selection, least median of squares estimation to determine the left and right edge slopes, and the shared vanishing point constraint while Redmill [19] used a matched filter and a Kalman filter for lane detection. The Kalman filter is applied to preserve smoothness and to predict lane parameters for the next image frame.

Three-feature based automatic lane detection algorithm (TFALDA) developed in this paper, is primarily intended for: 1) automatic extraction of the lane boundaries without manual initialization or *a priori* information under different road environments and 2) real-time processing. It is based upon similarity match in a three dimensional (3-D) space spanned by the three features of a lane boundary—starting position, direction (or orientation), and its gray-level intensity features-comprising a lane vector are obtained via simple image processing. The best lane candidate is chosen as the one at a minimum distance from the previous lane vector according to a weighted distance metric in which each feature is assigned a different weight for equalization. An evolutionary algorithm then finds the optimal weights for combination of the three features that minimize the rate of detection error. Finally, the lane boundaries thus found are inverse perspective transformed [20] to their corresponding plan-view image for the derivation of the steering angles to be used for subsequent road following. This TFALDA algorithm has been used as part of a complete system to drive an autonomous vehicle, PRV II (Fig. 1). The system has been used to successfully drive the vehicle on campus roads as well as nearby highways.

Manuscript received August 1, 2000; revised May 9, 2002. This work was supported by the Ministry of Education of Korea toward the ECE Division at POSTECH through its BK21 program and by the Ministry of Science and Technology for its Brain Science and Engineering Research Program. The Associate Editor for this paper was H. Takahashi.

Y. U. Yim is with the Department of Electrical Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: yimy@rpi.edu).

S.-Y. Oh is with the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, South Korea 790-784 (e-mail: syoh@postech.ac.kr).

Digital Object Identifier 10.1109/TITS.2003.821339

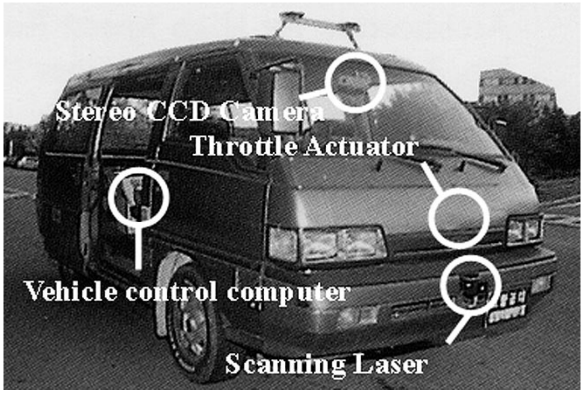

Fig. 1. Experimental testbed (PRV II).

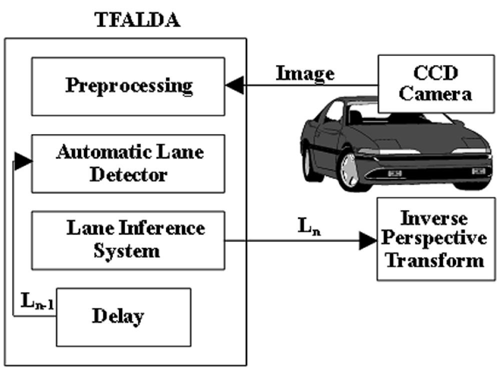

Fig. 2. Block diagram of TFALDA.

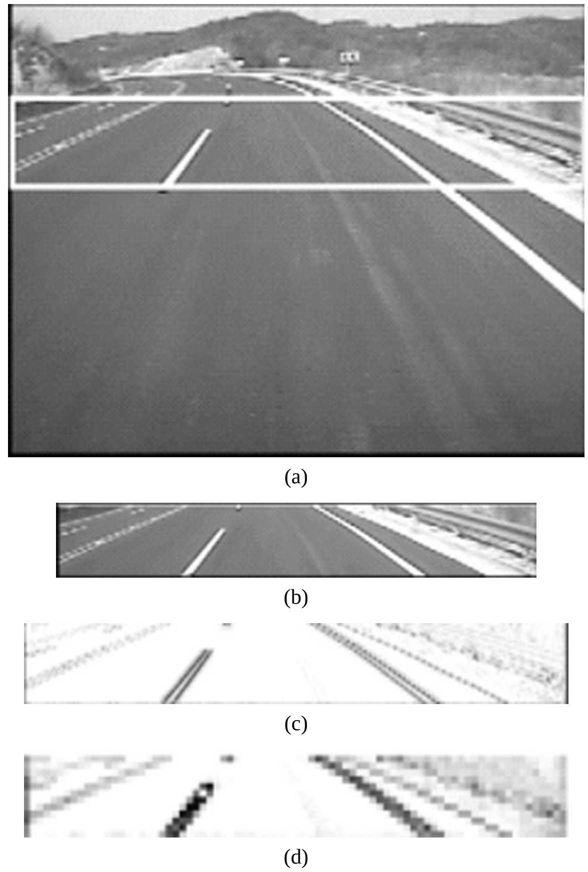

Fig. 3. Image preprocessing. (a) Full image, (b) partial image, (c) edge enhancement, and (d) resampling.

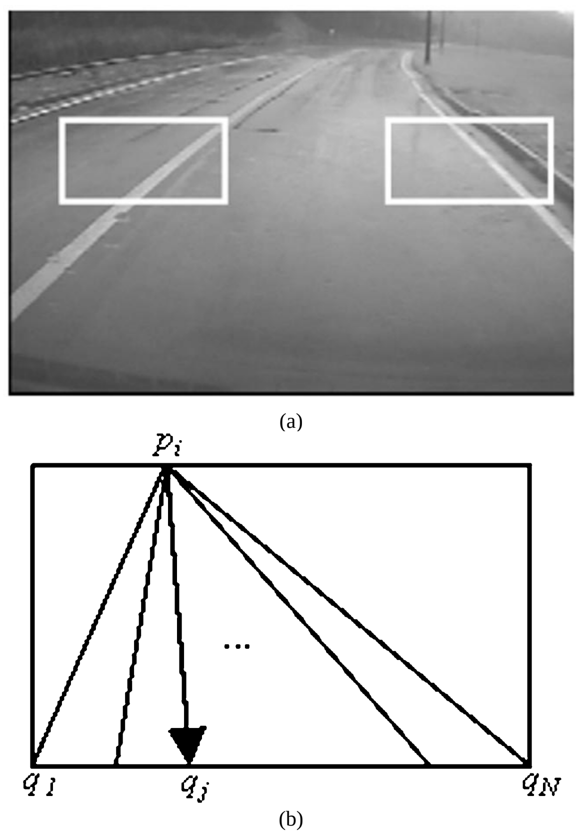

Fig. 4. Finding lane candidate vectors. (a) Road image with two windows from p_i. (b) Notation for determination of the lane candidate starting.

II. TFALDA Algorithm

The three processing components of TFALDA are shown in Fig. 2. Their details follow.

A. Preprocessing

This step consists of Sobel edge enhancement after data reduction through windowing and resampling as shown in Fig. 3. Edge enhancement is basically a differentiation step which provides robustness independent of the brightness variation due to the shadows and the moving sun.

B. Automatic Lane Detector

TFALDA represents each lane boundary as a 3-D lane vector in its feature space. It locates a new lane vector from the current road-view image and the previously found lane vector for fast and accurate detection.

The two rectangular windows of Fig. 4(a) indicate the regions of interest in search of the left and right boundaries. Assuming N pixels per horizontal line, there are N possible candidate lane vectors. The ith such candidate, C_i is then defined as

$$C_i = [P(C_i), I(C_i), D(C_i)] \tag{1}$$

where

$P(C_i)= i$;

$I(C_i) = \mathrm{Max}_j$ (Σ pixel intensities along $\overline{p_i q_j}$ as defined in Fig. 4(b);

$D(C_i)=$ Angle of $\overline{p_i q_k}$ where $\overline{p_i q_k} = \mathrm{Max}_j\, \overline{p_i q_j}$.

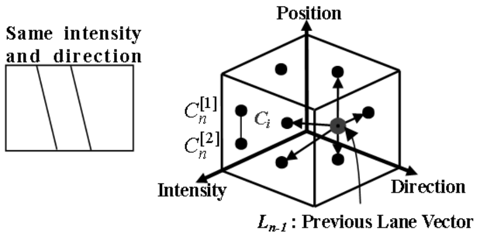

Fig. 5. Lane candidate vectors.

Fig. 5 shows N of these candidate vectors mapped into the 3-D feature space. The information on the previous lane vector is very important for fast and stable identification of the current lane since lane detection is a continuous tracking process. However, at the very beginning, there is no previous lane information and thus it has to be found on its own. This presents no problem since typically autonomous driving starts from rest and there is enough time to find the initial lane vector through the Hough transform.

Based upon the fact that lane markings cannot exhibit abrupt discontinuity, the particular candidate lane vector which is nearest to the previous one L_{n-1}, becomes the current lane L_n where n denotes the image sequence. To this end, the following distance metric is defined:

$$\lambda_i = K_P|P(C_i) - P(L_{n-1})| \\ + K_D|D(C_i) - D(L_{n-1})| + K_I(I(L_{n-1}) - I(C_i)). \quad (2)$$

The last term may look strange at first since it does not contain the absolute sign. The edge intensity could vary widely with the shadows and the direction of lighting. The stronger the edge intensity, the greater chance of its being the lane marking and this is why there is no absolute sign for the intensity term in (2). The three gains K_P, K_D, and K_I represent the scaling factors or weights applied to each feature direction. These weights have been optimized over different environments as will be explained in Section II-C. Since it uses three features together, TFALDA automatically can find the correct lane boundary by searching for the brightest and longest edge even, for example, when the wrong position and/or the direction of the previous lane boundary has been identified.

C. Lane Inference System

This is the last safeguard feature for the detection algorithm. The Lane Inference System is a kind of temporal predictor. The sequence of lane boundary information obtained by the Automatic Lane Detector is stored in the Lane Inference System which acts as a memory. When a new road image is incident, the Lane Inference System checks the temporal change of the computed lane width as well as of the direction of the lane boundaries in order to validate the newly found lane boundary. In other words, a low pass filtering of the recent lane vectors is performed to verify the plausibility of the newly found one. For example, if the road width goes through an unrealistic big jump, the current lane found by the TFALDA may be discarded and the previous lane may be reused as the current one.

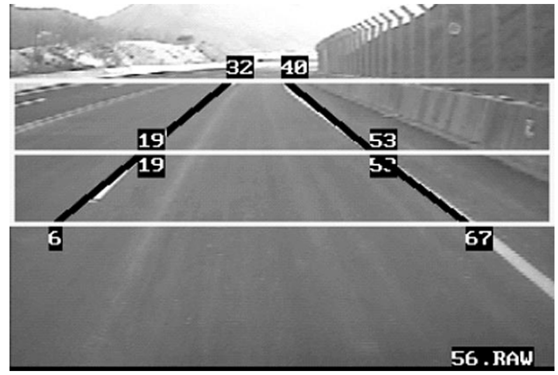

Fig. 6. Lane detection results.

III. Estimation of Optimal Scaling Factors Using an Evolutionary Algorithm

A. Evolutionary Programming

Evolution is in essence a two-step process of random variation and selection [21]–[23]. Evolutionary programming (EP) is one paradigm for simulating evolution which utilizes the concepts of Darwinian evolution to iteratively generate increasingly appropriate solutions under a static or a dynamically changing environment. Simulated evolution may be considered as an optimization technique wherein the algorithm iteratively optimizes behaviors, parameters, or other constructs. It basically utilizes the four main components: initialization, variation, evaluation, and selection. The basic EP paradigm may be described as follows:

$$\begin{aligned}
&t = 0\\
&initialize P(t)\\
&while\ not\ terminate\ do\\
&P'(t) = variation P(t)\\
&Evaluate P'(t)\\
&P(t+1) = select[P'(t) \cup P(t)]\\
&t = t+1\\
&end
\end{aligned}$$

where
$P(t)$ is a population of μ individuals at generation t;
$P'(t)$ is a population of λ individuals generated by variation operators at generation t.

This paper uses the basic EP algorithm with the following mutation rule:

$$\begin{aligned}
x'_i &= x_i + N(0, \sigma_i)\\
\sigma'_i &= \sigma_i \cdot \exp(\tau \cdot N(0,1) + \tau' \cdot N_i(0,1))
\end{aligned} \quad (3)$$

where x_i is the ith component of the real-valued vector representation $\mathbf{x}$, σ_i the standard deviation of the noise for the ith component, τ and τ' are operator-set parameters, and $N(0, \sigma)$ is a Gaussian distribution with zero mean and σ variance.

B. Training Data Generation

Since TFALDA's performance is largely a function of the scaling factors used in the distance metric, an optimal set of

Fig. 7. Wide range of road environments under test. (a) Center lines and dotted, (b) dotted and solid, (c) dotted and dotted, (d) inclined road, (e) shady road, and (f) wet road.

these parameters were found using an evolutionary programming technique. The optimal weights minimizing the misclassification rate are obtained by running an evolutionary algorithm off-line. It attempts to fit a model for how to combine the three distinct features (namely, position, intensity, and orientation) to correctly identify the lane boundaries, using a training set of 200 road image samples taken from a great variety of likely driving environments. In actual driving experiments, the optimal coefficients that have been determined off-line will be used on-line to detect lane boundaries from images. The training images were compiled from various road environments ranging from clear to rainy days, from clear to noisy lane markings, from dashed to solid lines, and from the shadows of the trees to those of the overhead bridges. The correct lane boundary for each image was visually identified to serve as the error reference.

C. Cost Function

Definition of the cost function is very important because it affects the lane detection performance. Since only the relative cost is important, K_I has been set to one. Thus, the input variables for the evolutionary algorithm is $\mathbf{K} = [K_P, K_D]$ with the following cost function:

$$f(\mathbf{K}) = Q \cdot \sum_{n=1}^{M} \left(\left| P\left(C_n^{[2]}\right) - P(L_n) \right| + \left| D\left(C_n^{[2]}\right) - D(L_n) \right| \right) + \left(\frac{\lambda_n^{[1]}}{\lambda_n^{[2]}} \right) \quad (4)$$

where the second index for λ has been added to indicate the image sequence. Here, M is 200, the number of total images and

TABLE I
TFALDA PERFORMANCE BEFORE AND AFTER SCALING OPTIMIZATION

Road Type		No. Images	Failed	Success Rate Before Optimization	Success Rate After Optimization
Highway	Dotted-Solid Lane #1	970	16	98.4 %	100 %
	Dotted-Solid Lane #2	303	3	99 %	100 %
	Solid-Dotted Lane	1251	45	96.4 %	100 %
	Dotted-Dotted Lane	964	64	93.4 %	100 %
Campus Road	Left Curved Lane	139	0	100 %	100 %
	Left Curved Lane (Erased)	210	15	92.9 %	100 %
	Left Curved Lane (Rainy)	374	2	99.5 %	100 %
	Right Curved Lane (Shady)	305	3	99 %	100 %
Total		4516	148	96.7 %	100 %

n represents the image index. L_n and L_{n-1} denote the correct current and the previous lane vectors both of which are manually found. $C_n^{[1]}$ is the closest candidate vector to L_{n-1} while $C_n^{[2]}$ is the second closest and $\lambda_n^{[1]}$ and $\lambda_n^{[2]}$ are their distances from L_{n-1}. The relative weight Q was empirically chosen as 10 so that each factor has comparable importance. The cost function becomes smaller as $C_n^{[1]}$ approaches L_n and also as $C_n^{[1]}$ is more likely to be the correct one than $C_n^{[2]}$. This way, the evolutionary algorithm aims for both the absolute and relative performance improvements.

D. Results of the Evolution Experiments

The scaling factors used in defining the cost function are optimal with respect to the number of lane detection errors after applying EP to the training road images. For training set, correct lanes are the ones found by a human. At the evaluation stage, the cost was defined as the difference between the TFALDA-found lane boundaries utilizing the current values of $[K_P, K_D]$ and the human-found boundaries. Finally, probabilistic selection by ranking the noise-added cost was used for EP. The population consists of 40 individuals and the minimum cost found after evolution was 24.97. This signifies that fewer than three misclassifications were reported out of 400 left or right lane markings in the 200 images. The optimal values of the scaling factors at this point were: $K_P = 7.33$, $K_I = 1.0$, and $K_D = 10.67$.

IV. INVERSE PERSPECTIVE TRANSFORM

This transform translates the camera image into a plan-view image. Assuming the camera is fixed to the vehicle and also the flat earth, it becomes the following linear transform with constant coefficients:

$$\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & l \end{bmatrix} \begin{bmatrix} x \\ y \\ l \end{bmatrix} \tag{5}$$

where (x, y) are the image coordinates and the (x', y') are the actual world coordinates of the points on the markings, and s is the scale factor of the homogeneous transform.

In conventional lane detection, several constraints are applied to the plan-view image found by the Inverse Perspective Transform such that the left and right lane markings are parallel. On the other hand, in TFALDA, only several feature points of the perspective image are inverse perspective transformed to determine the position and orientation of the vehicle relative to the lane. While the conventional method takes time since the entire image points must be transformed, TFALDA processes only a few feature points essential for lane departure warning which enables 30 Hz of image processing rate.

V. LANE DETECTION RESULTS

Fig. 6 shows the two regions of interest (one looking close and the other looking farther) in which to detect lane markings. This is done so that the upper and lower line segments can also provide the curvature information. The indices of the six points in Fig. 6 represent the lateral pixel positions representing the left and right markings found by TFALDA.

However, since the roads are in general curved, the three points chosen on a lane boundary have been inverse perspective transformed and then curve fitting of a circular arc passing through these three feature points was performed in a plan view image of the lane. Fig. 7 shows the varying environments used in the experiment.

Table I summarizes the statistics of the TFALDA performance. A detection failure was defined to occur when a manually found lane boundary deviates by more than a lane marking width from the TFALDA result. Notice that no failure occurred when optimal tuning of the TFALDA parameters were effected. The actual detection results on a shady road (noisy lane features) and a rainy road (unclear lane features) are shown in Fig. 8(a) and (b). It demonstrates the power of TFALDA despite bad qualities of the road image. For the other images

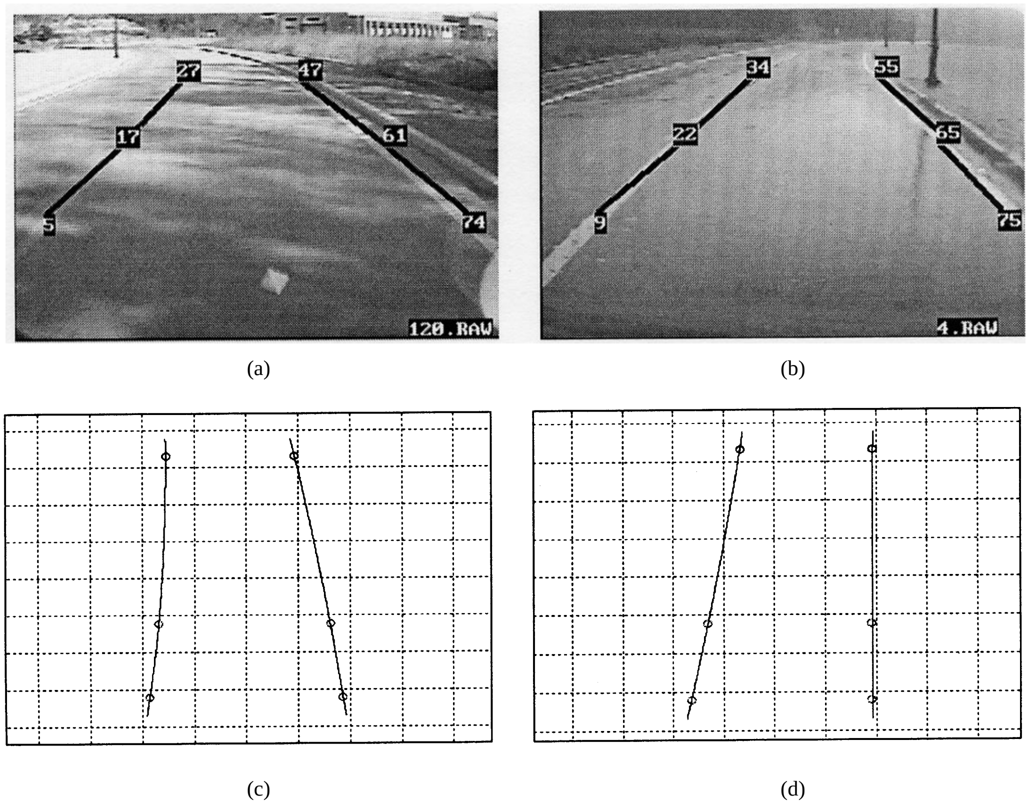

Fig. 8. Lane detection in bad conditions of the road. (a) Shady road image, (b) wet road image, (c) inverse perspective transform of (a), and (d) inverse perspective transform of (b).

which were not shown due to space limitation, it was also able to find the correct lane markings for all the images tested.

Fig. 8(c) and (d) shows the result of curve fitting of the three feature points in the plan view image. The reason why the plan view lane markings do not appear parallel is because the flat earth assumption was violated for the roads under test. TFALDA could process 30 images per second using a Pentium II @ 266 MHz.

VI. Conclusion

TFALDA has been presented that lends itself to rapid automatic detection of the lane boundaries under various environments without tedious manual initialization process or *a priori* information on the road. The strength of the algorithm comes from fusing together all three of the most important features of a lane boundary as well as checking their temporal variation. Furthermore, TFALDA was able to perform robustly under a wide variety of road conditions by optimizing its parameters through an evolutionary algorithm, instead of manually testing and improving them. The lanes thus detected can be inverse perspective transformed into a plan-view image from which the steering angle for road following and/or the necessary information for lane departure check may be directly obtained.

References

[1] M. Chen, T. Jochem, and D. Pomerleau, "AURORA: A vision-based roadway departure warning system," in *Proc. IEEE Intelligent Robots Systems*, 1995, pp. 243–248.

[2] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," in *Proc. IEEE Intelligent Vehicle Symp.*, 1995, pp. 54–59.

[3] D. Pomerleau, "RALPH: Rapidly adapting lateral position handling," in *Proc. IEEE Intelligent Vehicle Symp.*, 1995, pp. 506–511.

[4] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert*, pp. 19–27, Apr. 1996.

[5] D. Pomerleau, "Visibility estimation from a moving vehicle using the RALPH vision system," in *Proc. IEEE Intelligent Transportation Systems*, 1997, pp. 906–911.

[6] A. Broggi, "A massively parallel approach to real-time vision-based road marking detection," in *Proc. IEEE Intelligent Vehicles Symp.*, 1995, pp. 84–89.

[7] ——, "Robust real-time lane and road detection in critical shadow conditions," in *Proc. IEEE Int. Symp. Computer Vision*, 1995, pp. 353–358.

[8] M. Bertozzi and A. Broggi, "Real-time lane and obstacle detection on the GOLD system," in *Proc. IEEE Intelligent Vehicle Symp.*, 1996, pp. 213–218.

[9] ——, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Processing*, vol. 7, pp. 62–81, Jan. 1998.

[10] Broggi, M. Bertozzi, A. Fascioli, C. Guarino, and A. Piazzi, "Visual perception of obstacle and vehicles for platooning," *IEEE Trans. Intelligent Transport. Syst.*, vol. 1, Sept. 2000.

[11] K. Kluge and C. Thorpe, "The YARF system for vision-based road following," *Math. Computer Modeling*, vol. 22, no. 4–7, pp. 213–233, 1995.

[12] S. Lakshmanan and K. Kluge, "Lane detection for autonomous sensors," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1995, pp. 2955–2958.

[13] J. Crisman and C. E. Thrope, "SCARF: A color vision system that tracks roads and intersections," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 49–58, Feb. 1993.

[14] A. M. Waxman, J. J. Lemoigne, L. S. Davis, B. Srinivasan, T. R. Kushner, E. Liang, and T. Siddalingaiah, "A visual navigation system for autonomous land vehicles," *IEEE J. Robot. Automat.*, vol. RA-3, Apr. 1987.

[15] S. G. Jeong, C. S. Kim, K. S. Yoon, J. N. Lee, and J. I. Bae, "Real-time lane detection for autonomous vehicle," in *Proc. IEEE Intelligent Transportation Systems*, 2001, pp. 508–513.

[16] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *Proc. IEEE Intelligent Vehicle Symp.*, 1994, pp. 109–114.

[17] K. Kluge and G. Johnson, "Statistical characterization of the visual characteristics of painted lane marking," in *Proc. IEEE Intelligent Vehicle Symp.*, 1995, pp. 488–493.

[18] M. Nashman and H. Schneiderman, "Real-time visual processing for autonomous driving," in *Proc. IEEE Intelligent Vehicle Symp.*, 1993, pp. 373–378.

[19] K. A. Redmill, S. Upadhya, A. Krishnamurthy, and Ü Özgüner, "A lane tracking system for intelligent vehicle application," in *Proc. IEEE Intelligent Transportation Systems*, 2001, pp. 273–279.

[20] X. Chen, E. L. Dagless, S.-J. Zhang, and B. T. Tomas, "A real-time plan-view method for following bending roads," in *Proc. IEEE Intelligent Vehicles Symp.*, 1993, pp. 219–224.

[21] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 3–17, Apr. 1997.

[22] T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Oxford, U.K.: Oxford Univ. Press, 1997.

[23] D. B. Fogel, Ed., *Evolutionary Computation: The Fossil Record*. Piscataway, NJ: IEEE Press, 1998.

Young Uk Yim received the B.S. degree in electrical engineering from Korea University, Seoul, Korea, in 1996 and the M.S. degree in electrical engineering, Pohang University of Science and Technology, Pohang, Korea, in 1999. Currently, he is working toward the Ph.D. degree at Rensselaer Polytechnic Institute (RPI), Troy, NY.

From 1999 to 2000, he was with Samsung Advanced Institute of Technology, Korea, where he was involved in research on obstacle detection at night driving, and the intelligent robotic system for minimally invasive surgery. In 2000, he enrolled in the Department of Electrical Engineering, RPI, and where he has participated in the design of analog integrated circuits at the Center for Integrated Electronics. His present research focus is on RF/analog integrated circuit design with SiGe HBT technology.

Se-Young Oh (SM'76) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1974 and the M.S. and Ph.D. degrees in electrical engineering from Case Western Reserve University, Cleveland, OH, in 1978 and 1981, respectively.

From 1981 to 1984, he was an Assistant Professor in the Department of Electrical Engineering and Computer Science, University of Illinois at Chicago. From 1984 to 1988, he was an Assistant Professor in the Department of Electrical Engineering at the University of Florida, Gainesville. In 1988, he joined the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, South Korea, where he is currently a Professor. His research interests include soft computing technology, neural networks, fuzzy logic, and evolutionary computation and its application to robotics and intelligent vehicles, sensor-based control, and face recognition.