利用電腦視覺原理使自動車能安全行駛於路面有汽車的室外道路上

Vision-based Guidance for Autonomous Land Vehicle Navigation in Outdoor Road Environments with Static and Moving Cars[@]

陳光雄
Chen Kuang-Hsiung
國立交通大學資訊科學系
Department of Computer and Information Science, National Chiao-Tung University
gis81543@cis.nctu.edu.tw

蔡文祥*
Tsai Wen-Hsiang
國立交通大學資訊科學系
Department of Computer and Information Science, National Chiao-Tung University
whtsai@cis.nctu.edu.tw

摘要

本文提出一種利用電腦視覺、模式比對、及彩色分群等技巧的新方法使自動車能安全行駛於路面有汽車的室外道路上。本方法偵測道路上可以安全行駛的路面來達到避碰的效果。我們使用路的邊界來建立模式並用路面的明亮度來當作視覺特徵。我們檢查左右兩條邊界線附近的點來決定路的寬度是否因爲路面上存在著靜止或行駛中的汽車而產生變化。如果左右兩邊車道的寬度均未發生變化，我們執行模式比對來對自動車作定位。如果有任一車道的寬度發生變化，我們則執行相關的程序來求出新路面的寬度。如果新路面的寬度不等於舊路面的寬度，我們便重建模式，然後執行模式比對來找出自動車在路面上的位置。接著我們計算出一個適當的車子轉角使自動車沿著新路面的中心線行駛。我們做了許多成功的航行測試驗證出所提方法的有效性。

Abstract

A new effective approach to vision-based guidance for autonomous land vehicle (ALV) navigation in outdoor road environments with static and moving cars using model matching and color information clustering techniques is proposed. Collision-free road area detection is adopted for collision avoidance. Road boundaries are used to construct the reference model, and road surface intensity is selected as the visual feature. The pixels in a road image near the two lines representing the road boundary shape are checked to judge whether the lane width is changed due to occlusion caused by nearby static or moving cars on the road. If both lane widths are not changed, model matching is performed to find the ALV location. If either or both lane widths are changed, corresponding processes are performed to find the width of the occluded road, and the model is recreated if the new road width is different from the old one in the previous navigation cycle. Model matching is then performed to locate the ALV on the road. A turn angle is then computed to guide the ALV to follow the central path line on the extracted road for safe navigation. Successful navigation tests show that the proposed approach is effective for ALV guidance in common roads with static and moving cars.

[@]This work was supported by National Science Council, Republic of China under Grant NSC86-2213-E009-114.
*To whom all correspondence should be sent.

1. Introduction

Many research works have been reported for obstacle detection in outdoor roads [1-13]. Most systems, such as the CMU Navlab [1-2], the vehicle constructed by Martin Marietta Denver Aerospace [3-4], and the navigation system developed at the university of Maryland [5], use laser range sensors to detect obstacles in outdoor roads or cross-country terrain. The FMC Corporation [6] uses a sonic imaging sensor and an infrared sensor for obstacle avoidance and target detection.

As to vision-based approaches to obstacle detection, they basically can be divided into three classes. The first class extracts obstacles directly from 2-D images [7-10]. The second class of approaches uses motion information obtained from a sequence of images to detect obstacles [11-12]. The most popular approaches in this class are based on optical flow. The third class of approaches detects obstacles using stereo-vision techniques [13].

The major obstacles in outdoor roads are static and moving cars. Several approaches have been proposed to detect cars. Schwarzinger et al. [7], Thomanek, Dickmanns, and Dickmanns [8], and Regensbruger and Graefe [10] used some symmetry features in the rear side of a car to detect and track cars in front. Cappello, Camponi, and Succi [9] proposed a car detection method based on the assumptions that cars are approximately solid blocks with parallel or orthogonal sides and that the sides are parallel or orthogonal to road boundaries. Heisele and Ritter [11], and Smith and Brady [12] detected cars based on optical flow. BART [13] presented a visual control system for vehicle following. It tracks an apparent feature, a single point, on the back of the leading vehicle using stereo-vision techniques.

In this paper, we propose an effective approach to guiding an ALV in general roads with static and moving cars. The approach allows the variations of road widths, which are caused by existence of static cars on the roadside or moving cars on the road lane. New algorithms are proposed for finding the best-fit road region without having to do exhaustive search. We do not detect cars in the image; instead we detect road areas by using the road surface intensity as the visual feature. The road width is not assumed to be constant; instead the road width changes are allowed and computed as cars appear in the image. Besides, we do not plan a complex navigation route but just compute the central line on the extracted road as the route and approach it immediately after the ALV location on the extracted road is found. Neither with additional feature extraction from the car body nor with additional planning for the navigation path, fast navigation can be achieved.

In this study, the shape of a road is represented by its two road boundaries. To find the road boundaries on a road, edge points are extracted directly from the road image first. The edge points are then used to figure out the road boundaries. When complex obstacles appear on the road, the edges of the road become irregular and cause difficulty in road boundary estimation using the edge points. This motivates in this study the design of a model matching method which uses several candidate boundary lines in the model to match the road pixels nearing the estimated road boundaries in the image according to a reasonable similarity measure.

The remainder of this paper is organized as follows. In Section 2, the details of the proposed model-based ALV guidance method is described. The description of the employed image processing and feature extraction techniques is included in Section 3. In Section 4, the results of some experiments are described. Finally, some conclusions are stated in Section 5.

2. Proposed ALV Guidance Method

The proposed guidance scheme is performed in a cycle-by-cycle manner. In each navigation cycle, the system identifies the visual feature of the road surface in the road environment to locate the ALV, and guides the ALV accordingly from the current position to the navigation path which is assumed to be the central line of the unoccluded road portion. After the two lines representing the road boundary shape in the image are estimated at the beginning of the cycle, the system checks the pixels near the two lines to judge whether the left or right lane width is changed in the current cycle. Model matching is then performed to find the ALV location if both lane widths are not changed. If either or both lane widths are changed, appropriate procedures are executed to find the new road width, and the model is recreated if the new road width is different from the old road width computed in the previous cycle. Then model matching is performed to find the ALV location on the changed road.

In the following, we state first the steps of model creation which involves several coordinate systems and transformations, then the processes for the detection of lane width changes as well as the estimation of the new road width, followed by the model matching process.

2.1 Model Creation

Several coordinate systems and coordinate transformations are used in this approach. The image coordinate system (ICS), denoted as u-w, is attached to the image plane of the camera mounted on the ALV. The camera coordinate system (CCS), denoted as u-v-w, is attached to the camera lens center. The vehicle coordinate system (VCS), denoted as x-y-z, is attached to the middle point of the line segment which connects the two contact points of the two front wheels of the ALV with the ground. The x-axis and the y-axis are on the ground and parallel to the short and the long sides of the vehicle body, respectively. The z-axis is vertical to the ground. The transformation between these coordinate systems can be seen in [15,16].

An ALV location can be represented by two parameters (d, θ), where d is the distance of the ALV to the central path line in the road and θ is the pan angle of the ALV relative to the road direction. The equations of the two road boundaries in the VCS are assumed to be known before navigation. We transform the road boundaries into the ICS which are displayed on the TV monitor during each navigation session.

For each ALV location (d_i, θ_j), we create a corresponding template $T_{ij}[a_l, b_l, a_r, b_r]$, where a_l and a_r are the slopes and b_l and b_r are the intercepts, of the equations of the left and the right road boundaries in the ICS, respectively. The transformation is shown in Fig. 1. We sample the road width at 41 positions, i.e., $-20 \le i \le 20$, with the interval of 1/40 of the road width. At each position, we sample the vehicle direction at 31 angles from -15 degrees to +15 degrees, i.e., $-15 \le j \le 15$, with the interval of one degree. Hence the model contains $41 \times 31 = 1271$ templates, and each template represents a specific ALV location. Because the templates are represented in the ICS, the model matching process described later is also operated in the ICS.

2.2 Detection of Lane Width Changes

The ALV keeps moving forward after an image is taken at the beginning of each navigation cycle. After the image is processed and corresponding algorithms are performed, the ALV location at the time instant of image taking can be found. At this moment, however, the ALV has travelled a distance. Hence the ALV never knows its actual current position unless the cycle time is zero. To overcome this difficulty, the system uses the ALV control information to estimate the current ALV location according to a method proposed by Cheng and Tsai [17]. We then define the *reference ALV location* in cycle i to be the estimated ALV location at the beginning time of cycle i.

Let T_{r_s} denote the reference ALV location (d_r, θ_s) in the current cycle, and we call T_{r_s} the *reference template* in the current cycle. To judge whether the left or right lane width is changed, we first check those templates, called *candidate templates*, which are "close" to the reference template in the current cycle. A *candidate template set* W is shown in Fig. 2, where the left line of each template in W lies in the area bounded by the left line of the leftmost candidate template (LCT) and the left line of the rightmost candidate template (RCT), and the right line of each template in W lies in the area bounded the right line of the LCT and the right line of the RCT. We also call T_{r_s} the *center template* in W.

Basically, a road in an image can be divided into three clusters according to their intensity values: (1) cluster-0: dark area, like shadows and trees; (2) cluster-1: gray area, coming from the main body of the road; and (3) cluster-2: bright area, like the sky and the white path lines on the road. Then, the ratio of the number of the cluster-1 pixels to the number of all pixels in the area bounded by the left line of the LCT and the left line of the RCT is checked to judge whether the left lane width is changed. If the ratio is smaller than some threshold value, say TH-1, the left lane is decided to be narrowed. If it is larger than some threshold value, say TH-2, the left lane is decided to be widened. If it is between TH-1 and TH-2, the left lane width is decided to be unchanged. Similarly, We check the ratio of the number of the cluster-1 pixels to the number of all pixels in the area bounded by the right line of the LCT and the right line of the RCT to judge whether the right lane width is changed. If both lane widths are not changed, model matching (MM) is then performed to find the ALV location, which will be described later. If either or both lane widths are changed, corresponding processes are executed to calculate the new road width, as stated in the following.

2.3 Estimation of New Road Width

If the left lane width is checked to be changed, then the changed amount W_l of the left lane width is estimated as illustrated in Fig. 3. Fig. 3(a) illustrates the estimation process for the case that the left lane is narrowed, where $T_{r_0,s} = (d_{r_0}, \theta_s)$ is the reference template, L_i is the left line of template $T_{r_0-i,s}$, and the dark area bounded by B_l and B_r is composed of the cluster-1

pixels (extracted road pixels). To calculate W_ℓ, we first check the cluster-1 pixels in the area bounded by L_0 and L_1. If they are sufficient in number, then L_0 is decided to be close enough to B_ℓ and is selected as the "approximate left road boundary", and W_ℓ is set to 0. Otherwise, the cluster-1 pixels in the area bounded by L_1 and L_2 are checked. If they are sufficient in number, then L_1 is decided to be close enough to B_ℓ and is selected as the approximate left road boundary, and W_ℓ is set to $d_{r_0-1} - d_{r_0}$. Otherwise, the cluster-1 pixels in the area bounded by L_2 and L_3 are checked in a similar way.

This process is repeated until a certain L_{n+1} is chosen such that the cluster-1 pixels in the area bounded by L_n and L_{n-1} are sufficient in number. Then L_n is selected as the approximate left road boundary, and W_ℓ is set to $d_{r_0-n} - d_{r_0}$. Note that this way of detecting the approximate road boundary", as proposed above in this study, facilitates the avoidance of direct estimation of road boundary lines from the edge points of broken- or irregular-shaped road boundaries resulting from existing roadside cars or obstacles. On the other hand, if the left lane is widened, a similar criterion is used to find W_ℓ, which is illustrated in Fig. 3(b), where L_n is the approximate left road boundary.

Similarly, if the right lane width is changed, the changed amount W_γ of the right lane width and the "approximate right road boundary" R_n, which is close enough to the real right road boundary B_γ in the image, are estimated in a similar way. After the changed amounts W_ℓ and W_γ of both lane widths are found, the new road width is calculated by

$$\text{New road width } W_{new} = \text{old road width } W_{old} + W_\ell + W_\gamma, \quad (1)$$

and the *approximate template*, which is defined to be composed of the approximate left and right road boundaries L_n and R_n, is estimated to be

$$T_{ms} = (d_m, \theta_s) = (d_{r_0} + W_\ell/2 - W_\gamma/2, \theta_s), \quad (2)$$

where $T_{r_0,s} = (d_{r_0}, \theta_s)$ is the reference template. If $W_{new} \neq W_{old}$, then we recreate the model in the way as described in Section 2.1 according to the new road width W_{new}. Finally, model matching is performed to find the ALV location, which is described in the following.

2.4 Model Matching for ALV Location

If both lane widths are detected to be unchanged in the current cycle, we use the candidate template set W described in Section 2.2 to perform model matching because the templates in W are close to the reference template which is the estimated road boundary shape of the unchanged road in the current cycle. If either or both lane widths are detected to be changed by nearby cars or other obstacles on the road ahead in the current cycle, the reference template could be far away from the real road boundary shape in the image, and it is unsuitable to use the set W described above to perform the matching. As described in the previous section, after the changed amounts W_ℓ and/or W_γ are found, we obtain the approximate template T_{ms} which is close to the real road boundary shape in the image. We thus choose T_{ms} as the center template to reconstruct W such that the templates in the reconstructed set W can be made closer to the real road boundary shape.

After W is chosen or reconstructed, we use it to perform model matching to find the ALV location. Without loss of generality, we first state how we match the templates in W with the set LCP of the left-check-pixels, which are defined as the cluster-1 pixels in the area bounded by the left line of the LCT and the left line of the RCT, to find the template T_ℓ whose left line is most likely to be the left road boundary. The matching process is based on a criterion which is described as follows. We define

Left-bounded-ratio LBR_{ij} = (the number of the cluster-1 pixels in the area bounded by the left line of T_{ij} and the left line of the RCT) / (the number of all pixels in the area bounded by the left line of T_{ij} and the left line of the RCT), and

Left-road-ratio LRR_{ij} = (the number of the cluster-1 pixels in the area bounded by the left line of T_{ij} and the left line of the RCT) / (the number of the LCP). (3)

Fig. 4 illustrates how we check the LBR and LRR values to decide which line is closer to the real left road boundary B_ℓ in different cases, where L_i is the left line of template T_i. In Fig. 4(a), L_0 and L_1 are on the left side of B_ℓ, and L_1 is closer to B_ℓ. It can be seen from the figure that T_0 and T_1 have the same LRR value (=100%). But the LBR value of T_1 is greater than that of T_0 because L_1 is closer to B_ℓ. This means that all of the templates in W, whose left lines are on the left side of B_ℓ, have the same LRR value, and the template whose left line is closer to B_ℓ has a larger LBR value.

In Fig. 4(b), L_0 and L_1 are on the right side of B_ℓ, and L_0 is closer to B_ℓ. It can be seen from the figure that T_0 and T_1 have the same LBR value (=100%). But the LRR value of T_0 is greater than that of T_1 because L_0 is closer to B_ℓ. This means that all of the templates in W, whose left lines are on the right side of B_ℓ, have the same LBR value, and the template whose left line is closer to B_ℓ has a larger LRR value.

In Fig. 4(c), L_0 and L_1 are on the left and right sides of B_ℓ, respectively. It can seen from the figure that the LRR value of T_0 is equal to the LBR value of T_1 (=100%). To decide which of L_0 and L_1 is closer to B_ℓ, we can compare the LBR value of T_0 and the LRR value of T_1. If the LBR value of T_0 is larger than the LRR value of T_1, then it is decided that L_0 is closer to B_ℓ. Otherwise, it is decided that L_1 is closer to B_ℓ.

From the above observation and analysis, it is reasonable to define the following similarity measure based on the two ratio values, LBR_{ij} and LRR_{ij}, for model matching:

$$LS_{ij} = LBR_{ij} + LRR_{ij}. \quad (4)$$

Then the best-matched template $T_\ell = (d_\ell, \theta_\ell)$, whose left line is the closest to B_ℓ, has the maximum LS value. We call this criterion of choosing T_ℓ the maximum-sum-of-double-ratios (MSODR) criterion.

Similarly, we can match the templates in W with the set RCP of the right-check-pixels based on the MSODR criterion to obtain the best-matched template T_γ whose right line is most likely to be the real right road boundary.

After $T_l = (d_l, \theta_l)$ and $T_r = (d_r, \theta_r)$ are found, the ALV location $T = (d, \theta)$ is estimated more accurately by averaging T_l and T_r, i.e.,

$$T = (d, \theta) = \frac{T_l + T_r}{2} = \left(\frac{d_l + d_r}{2}, \frac{\theta_l + \theta_r}{2}\right). \quad (5)$$

The proposed model matching (MM) algorithm based on the MSODR criterion is stated below.

Algorithm MM (Model Matching).

Input. (a) A set W of candidate templates.
(b) The set LCP of the left-check-pixels and the set RCP of the right-check-pixels.

Output. The ALV location T.

Step 1. For each candidate template T_{ij} in W, compute the left and right similarities LS_{ij} and RS_{ij}.

Step 2. From all of the computed LS_{ij} values, find the best-matched template T_l with the largest LS value.

Step 3. From all of the computed RS_{ij} values, find the best-matched template T_r with the largest RS value.

Step 4. Set $T = (T_l + T_r)/2$ and take T as the desired output.

Finally, a complete ALV location algorithm is described below.

Algorithm CALVL (Complete ALV Location).

Input. (a) A set V of cluster-1 pixels.
(b) A model M of templates created in the previous cycle.
(c) The reference template $T_{r_{ref},s} = (d_{r_{ref}}, \theta_s)$ in the current cycle.
(d) A set W of candidate templates in M whose center template is $T_{r_{ref},s}$.

Output. The ALV location T.

Step 1. Check of lane widths:
(a) Use V and W to check whether the left lane width is changed. If it is changed, compute the changed amount W_l of the left lane width.
(b) Use V and W to check whether the right lane width is changed. If it is changed, compute the changed amount W_r of the right lane width.

Step 2. Model recreation and candidate template set reconstruction: If either or both lane widths are changed, then perform:
(a) Calculate the new road width W_{new} = the old road width $W_{old} + W_l + W_r$.
(b) If $W_{new} \neq W_{old}$, then recreate the model M according to W_{new}.
(c) Estimate the approximate template $T_{ms} = (d_m, \theta_s) = (d_{r_0} + W_l/2 - W_r/2, \theta_s)$.
(d) Use T_{ms} as the center template to reconstruct the candidate template set W.

Step 3. Model matching:
Use V and W to find the LCP and the RCP, and with W, LCP, and RCP as inputs perform the MM algorithm to find the desired ALV location T as output.

After the desired ALV location is found, the navigation path on the extracted new road can be generated immediately. A turn angle for guiding the ALV from the current ALV location to the immediately generated navigation path is then computed. The navigation path estimation and the turn angle computation are described in the following.

3. Image Processing and Feature Extraction Techniques

To reduce the image size, the upper portion in the road image is discarded because it does not contain any road area. Next, pixels are sampled from the remaining image portion with the interval of six pixels in both the horizontal and vertical directions. We then use an ISODATA algorithm [14], which includes an initial-center-choosing (ICC) technique to solve the problem caused by great changes of intensity in navigations, to divide the road image into three clusters.

Intuitively, we can select the resulting centers in the previous navigation cycle as the initial centers in the current cycle to run the clustering algorithm. However, the selection may be unsuitable, producing unacceptable clusters, because some difference may exist between two consecutive images. One kind of the difference comes from the change of intensity. If the change of intensity between two consecutive images is great, the candidate initial centers chosen from the resulting centers in the previous cycle may be far away from their real centers in the current cycle. In this situation, many iterations are needed for the ISODATA algorithm to move the candidate initial centers close to their real centers, which take too much computing time. Fig. 5 shows this situation, where the R-component r_k of the resulting center of cluster k in the previous cycle is far away from the R-component R_k of the real center of cluster k in the current cycle because of the great change of intensity between the two consecutive images.

To choose proper initial centers closer to their real centers in the current cycle, we propose an ICC technique based on the assumption that the changes of the bright, the gray, and the dark areas between two consecutive input images are small. As shown in Fig. 5, the R-component r_k of the candidate initial center of cluster k resulting from the ICC technique is very close to the R-component R_k of the real center of cluster k in the current cycle. The ICC technique is described as follows. Let P_0, P_1, and P_2 be the numbers of pixels belonging to cluster-0, cluster-1, and cluster-2 in the previous cycle, respectively. What we want to compute is the initial centers of the three clusters to run the clustering algorithm in the current cycle based on the values of P_0, P_1, and P_2. We first observe the histogram of the R-plane shown in Fig. 5 and let r_0, r_1, and r_2 be such that the following equalities are satisfied:

$$\sum_{s=0}^{r_0}\left[\text{pixel no. of g.l.(s)}\right] = P_0/2,$$

$$\sum_{t=0}^{r_1}\left[\text{pixel no. of g.l.(t)}\right] = P_0 + P_1/2, \text{ and}$$

$$\sum_{u=0}^{r_2}\left[\text{pixel no. of g.l.(u)}\right] = P_0 + P_1 + P_2/2, \quad (6)$$

where g. l. means gray level. Then r_k, k=0, 1, 2, is taken to be the R-component of the candidate center of cluster k in the current cycle, which is very close to the R-component R_k of the real center of cluster k. Using the same criterion on the G-plane and B-plane, we can find the G-component g_k and B-component b_k. We then use $[r_k, g_k, b_k]$ as the initial center of cluster k to run the clustering algorithm, where k=0, 1, 2. In this way, it is found that three iterations are enough for the ISODATA

algorithm to move the candidate initial centers close to their real centers and much computing time is saved.

An example of obvious improvement obtained from using the ICC technique in a real road scene is shown in Fig. 6. Fig. 6(a) shows the input image with high intensity in the previous cycle and Fig. 6(b) shows the input image with low intensity in the current cycle. Fig. 6(c) shows the poor clustering result when the resulting centers in the previous cycle are used as the initial centers to run the clustering algorithm for three iterations in the current cycle. A better clustering result yielded by the ICC technique is shown in Fig. 6(d), in which cluster-1 is taken to be the road area.

4. Experimental Results

Based on the proposed approach and algorithms, a prototype ALV constructed for this study could navigate safely and smoothly along part of the campus road in National Chiao Tung University. A lot of successful navigation tests confirm the feasibility of the approach. The navigation path is not sensitive to sudden changes of intensity because of the effective ICC technique used in the clustering algorithm. The average cycle time is about 1.0 sec, and the average speed is 170 cm/sec or 6.2 km/hr.

Fig. 7 shows a real road image, its clustering result, the reference template represented by the dotted template, and the matched template represented by the black template. In the figure, one static car is on the right lane, another car is driven toward the ALV along the left lane, and the ALV is moving forward successfully along the central line on the extracted road.

5. Conclusions

A model-based approach to ALV guidance in outdoor road environments with static and moving cars by computer vision has been proposed. Several techniques have been integrated in this study to provide a reliable navigation scheme. Contributions made by the proposed approach are summarized in the following. (1) An ISODATA clustering algorithm based on the ICC technique has been proposed to solve the problem caused by great changes of intensity in navigations. (2) With no additional process for feature extraction on the car body and no additional planning for the navigation path, fast navigation can be achieved. (3) A model matching process based on the MSODR criterion has been proposed to identify a road easily without detecting complex obstacles appearing on the road, and to locate the ALV immediately on the identified road without complicated computation. (4) A scheme for ALV speed adjustment on varying road situations have been proposed to achieve safe, steady, and flexible navigation. (5) The proposed complete algorithm has been implemented to extract roads accurately in real time such that the ALV can avoid collision with nearby cars on the road ahead. Successful navigation tests in general roads confirm the effectiveness of the proposed approach. Future research directions may focus on recognition of special road conditions, detection and avoidance of more types of obstacles, and environment sensing and learning, etc.

References

[1] S. Singh and P. Keller, "Obstacle Detection for High Speed Autonomous Navigation," *Proc. IEEE International Conference on Robotics and Automation*, Sacramento, CA, U.S.A., April 1991, pp. 2798-2805.

[2] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Toward Autonomous Driving: The CMU Navlab, Part I -- Perception," *IEEE Expert*, Vol. 6, Iss. 3, pp. 31-42, Aug. 1991.

[3] M. A. Turk, D. G. Morgenthaler, K. D. Germban, and M. Marra, "VITS --- a vision system for autonomous land vehicle navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, pp. 342-361, May 1988.

[4] K. E. Olin and D. Y. Tseng, "Autonomous Cross-Country Navigation," *IEEE Expert*, Vol. 6, Iss. 3, pp. 16-30, Aug. 1991.

[5] L. S. Davis, "Visual navigation at the University of Maryland," *Robotics and Autonomous Systems*, Vol. 7, pp. 99-111, 1991.

[6] D. Kuan, G. Phipps, and A. Hsueh, "Autonomous Robotic Vehicle Road Following," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 648-658, 1988.

[7] M. Schwarzinger, T. Zielke, D. Noll, M. Brauchmann, and W. V. Seelen, "Vision-Based Car-Following: Detection, Tracking, and Identification," *Proc. of the Intelligent Vehicles '92 Symposium*, Detroit, U.S.A., Jun. 1992, pp. 24-29.

[8] F. Thomanek, E. D. Dickmanns and D. Dickmanns, "Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance," *Proc. of the Intelligent Vehicles '94 Symposium*, Paris, France, Oct. 1994, pp. 231-236.

[9] M. Cappello, M. Campani, and A. Succi, "Detection of Lane Boundaries, Intersections and Obstacles," *Proc. of the Intelligent Vehicles '94 Symposium*, Paris, France, Oct. 1994, pp. 284-289.

[10] U. Regensburger and V. Graefe, "Visual Recognition of Obstacles on Roads," *Proc. of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Munich, Germany, Sep. 1994, pp. 980-987.

[11] B. Heisele and W. Ritter, "Obstacle Detection Based on Color Blob Flow," *Proc. of the Intelligent Vehicles '95 Symposium*, Detroit, Mich., U.S.A., Sep. 1995, pp. 282-286.

[12] S. M. Smith and J. M. Brady, "A Scene Segmenter; Visual Tracking of Moving Vehicles," *Engineering Applications of Artificial Intelligence*, Vol. 7, No. 2, pp. 191-204, 1994.

[13] N. Kehtarnavaz, N. C. Griswold, and J. S. Lee, "Visual Control of an Autonomous Vehicle (BART) – The Vehicle-Following Problem," *IEEE Trans. on Vehicular Technology*, Vol. 40, No. 3, pp. 654-662, Aug. 1991.

[14] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, Inc., New York, U. S. A., 1973.

[15] L. L. Wang, P. Y. Ku, and W. H. Tsai, "Model-based guidance by the longest common subsequence algorithm for indoor autonomous vehicle navigation using computer vision," *Automation in Construction*, Vol. 2, pp. 123-137, 1993.

[16] Y. M. Su and W. H. Tsai, "Autonomous land vehicle guidance for navigation in buildings by computer vision, radio, and photoelectric sensing techniques," *Journal of the Chinese Institute of Engineers*, Vol. 17, No. 1, pp. 63-73, 1994.

[17] S. D. Cheng and W. H. Tsai, "Model-based guidance of autonomous land vehicle in indoor environments by structured light using vertical line information," *Journal of Electrical Engineering*, Vol.34, No.6, pp. 441-452, Dec. 1991.

Figures

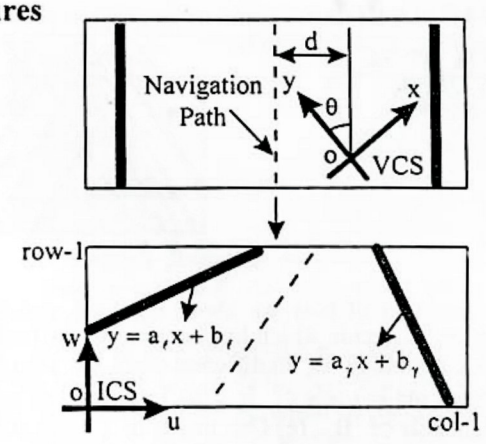

Fig. 1 The road boundary transformation between the VCS and the ICS.

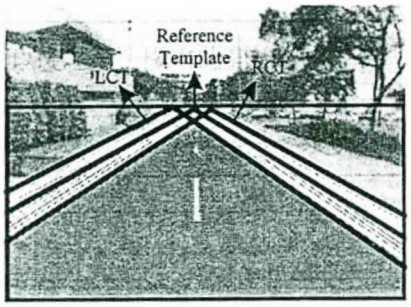

Fig. 2 A candidate template set W, where the LCT denotes the leftmost candidate template and the RCT denotes the rightmost candidate template in W.

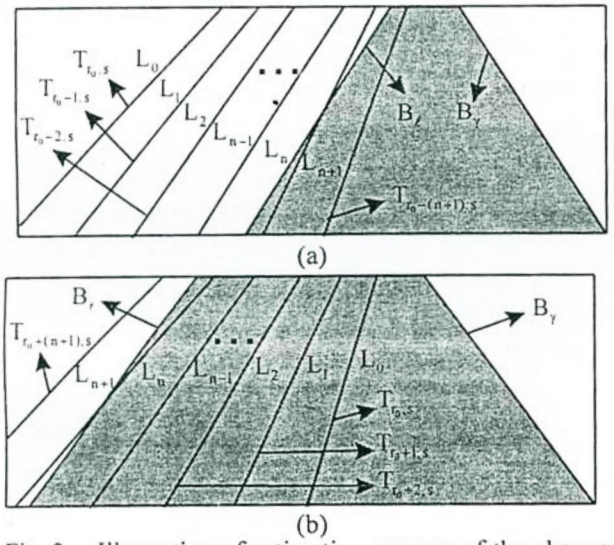

Fig. 3 Illustration of estimation process of the changed left lane width. (a) The left lane is narrowed. (b) The left lane is widened.

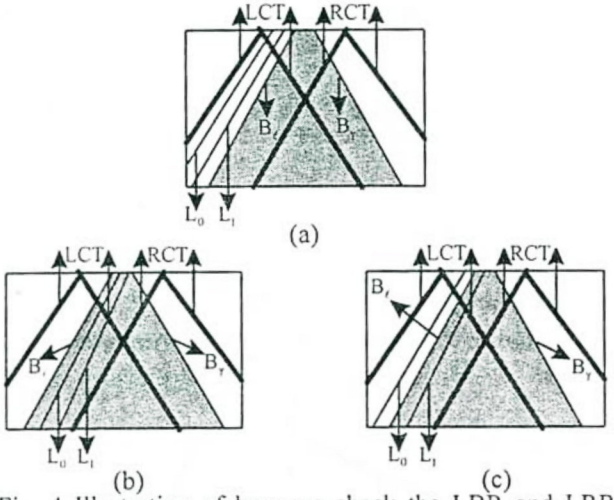

Fig. 4 Illustration of how we check the LBR and LRR values to decide which line is closer to the real left road boundary B_ℓ in different cases. (a) Two lines are on the left side of B_ℓ. (b) Two lines are on the right side of B_ℓ. (c) One line is on the left side of B_ℓ and one line is on the right side of B_ℓ.

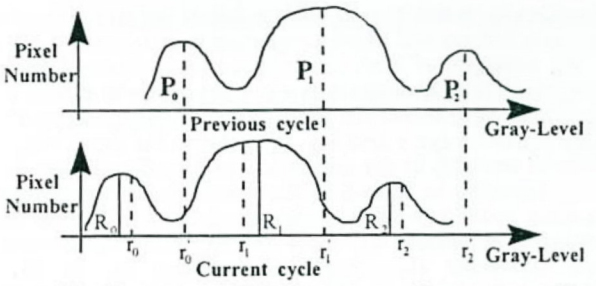

Fig. 5 The histograms of the R-planes of two consecutive images, where r_k' is the R-component of the resulting center of cluster k in the previous cycle, R_k is the R-component of the real center of cluster k in the current cycle, and r_k is the R-component of the candidate center of cluster k resulting from applying the ICC technique in the current cycle.

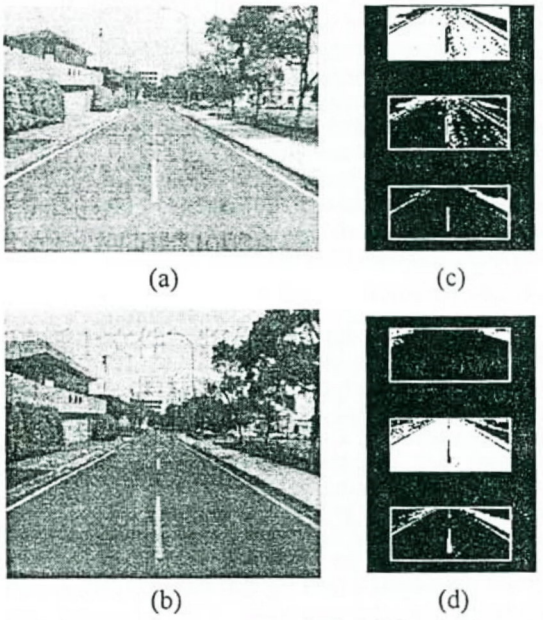

Fig. 6 An obvious improvement obtained from using the ICC technique in a real road scene. (a) The input image with high intensity in the previous cycle. (b) The input image with low intensity in the current cycle. (c) Poor clustering result when the resulting centers in the previous cycle are used as the initial centers to run the clustering algorithm for three iterations in the current cycle. (d) Better clustering result produced by the ICC technique.

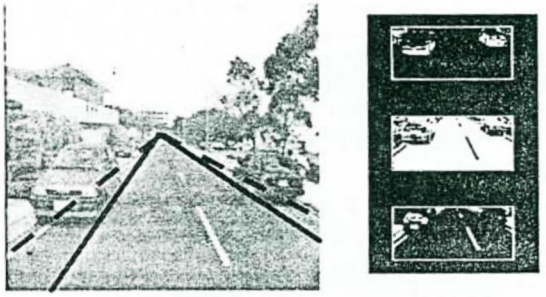

Fig. 7 A real road image, its clustering result, the reference templates represented by the dotted template, and the matched template represented by the black template, where one static car is on the right lane and one car is moving toward the ALV along the left lane.