ELSEVIER

Automation in Construction 10 (2000) 1–25

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

Vision-based obstacle detection and avoidance for autonomous land vehicle navigation in outdoor roads

Kuang-Hsiung Chen, Wen-Hsiang Tsai *

Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, 30050 Taiwan

Accepted 23 April 1999

Abstract

An effective approach to obstacle detection and avoidance for autonomous land vehicle (ALV) navigation in outdoor road environments using computer vision and image sequence techniques is proposed. To judge whether an object newly appearing in the image of the current cycle taken by the ALV is an obstacle, the object shape boundary is first extracted from the image. After the translation from the ALV location in the current cycle to that in the next cycle is estimated, the position of the object shape in the image of the next cycle is predicted, using coordinate transformation techniques based on the assumption that the height of the object is zero. The predicted object shape is then matched with the extracted shape of the object in the image of the next cycle to decide whether the object is an obstacle. We use a reasonable distance measure to compute the correlation measure between two shapes for shape matching. Finally, a safe navigation point is determined, and a turn angle is computed to guide the ALV toward the navigation point for obstacle avoidance. Successful navigation tests show that the proposed approach is effective for obstacle detection and avoidance in outdoor road environments.

Keywords: Autonomous land vehicle; Obstacle detection and avoidance; Image sequence; Computer vision; Image processing

1. Introduction

Autonomous land vehicles (ALVs) are useful for many automation applications both in indoor and outdoor environments. Vision-based obstacle detection for ALV navigation in outdoor road environments is a difficult and challenging task because of the great variety of object and road conditions, like irregular and unstable features on objects, moving objects, changes of illumination, and even rain. Successful ALV navigation requires the integration of the techniques of environment sensing and learning, image processing and feature extraction, ALV location, path planning, wheel control, and so on. This study is mainly concerned with obstacle detection and avoidance for ALV navigation in outdoor road environments using computer vision and image sequence techniques.

Many research works have been reported for obstacle detection in outdoor roads [1–33]. Most systems, such as the CMU Navlab [1–8], the vehicle constructed by Martin Marietta Denver Aerospace [9–11], and the navigation system developed at the

* Corresponding author. Tel.: +886-357159090; fax: +886-35721490

0926-5805/00/$ - see front matter
PII: S0926-5805(99)00010-2

University of Maryland [12], use laser range sensors to detect obstacles in outdoor roads or cross-country terrain. FMC [13] 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. In the first class, obstacles are extracted directly from 2D images [14–21]. Only one camera and only the image in the current navigation cycle are used, with certain a priori knowledge and predefined assumptions being considered. In the second class of approaches, motion information obtained from a sequence of images are utilized to detect obstacles [22–26]. The most popular approaches in this class are based on optical flow. In the third class of approaches, obstacles are detected using stereo-vision techniques [27–32]. Besides, Xie et al. [33] used a range finder coupled with a CCD camera to acquire 3D information of obstacles. Although the first class in general takes less computing time and has better detection results than the second and the third classes, in fact, it does not really detect obstacles because obstacles are extracted directly from the 2D image. Shadows on roads may also be regarded as obstacles in this class of approaches. On the contrary, in the second and the third classes, 3D computer vision techniques are used to really judge whether objects on roads are obstacles, although more computing time is required in these two classes than the first class.

In this paper, an effective approach to obstacle detection and avoidance for autonomous land vehicle navigation in outdoor road environments using computer vision and image sequence techniques is proposed. To judge whether one object newly appearing in the image of the current cycle is an obstacle, we first extract the object shape boundary from the image. After the translation from the ALV location in the current cycle to that in the next cycle is estimated, the position of the object shape in the image of the next cycle is predicted using coordinate transformation techniques, based on the assumption that the height of the object is zero. The predicted object shape is then matched with the extracted shape of the object in the image of the next cycle to decide whether the object is an obstacle. We use the *distance-weighted correlation* (DWC) [34] as the similarity measure between the two shapes for shape matching. Then a navigation point is computed, and a turn angle is decided accordingly to guide the ALV to follow the navigation point for obstacle avoidance. Successful navigation tests show that the effectiveness of the proposed approach for obstacle detection and avoidance in outdoor road environments.

A new prototype ALV with smart, compact, and ridable characteristics, as shown in Fig. 1, is constructed as a testbed for this study, whose dimension is 118.5 cm by 58.5 cm. It has four wheels in which the front two are the turning wheels and the rear two the driving wheels. Above the front wheels is a cross-shaped rack on which some CCD cameras are mounted, and above the rack is a platform on which two monitors, one being the computer monitor and the other the image display, are placed. Above the platform is a vertical bar on which another camera used for obstacle detection and avoidance in this study is mounted. The central processor is an IBM PC/AT compatible personal computer (Pentium-166) with a color image frame grabber which takes 512×486 RGB images, with eight bits of intensity per image pixel.

The ALV is computer-controlled with a modular architecture, as shown in Fig. 2, including four components, namely, a vision system, a central processor Pentium-166, a motor control system, and a DC power system. The vision system consists of a cam-

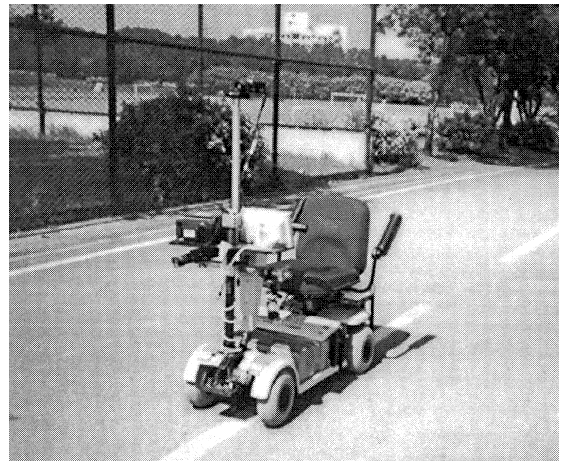

Fig. 1. The prototype ALV used in this study.

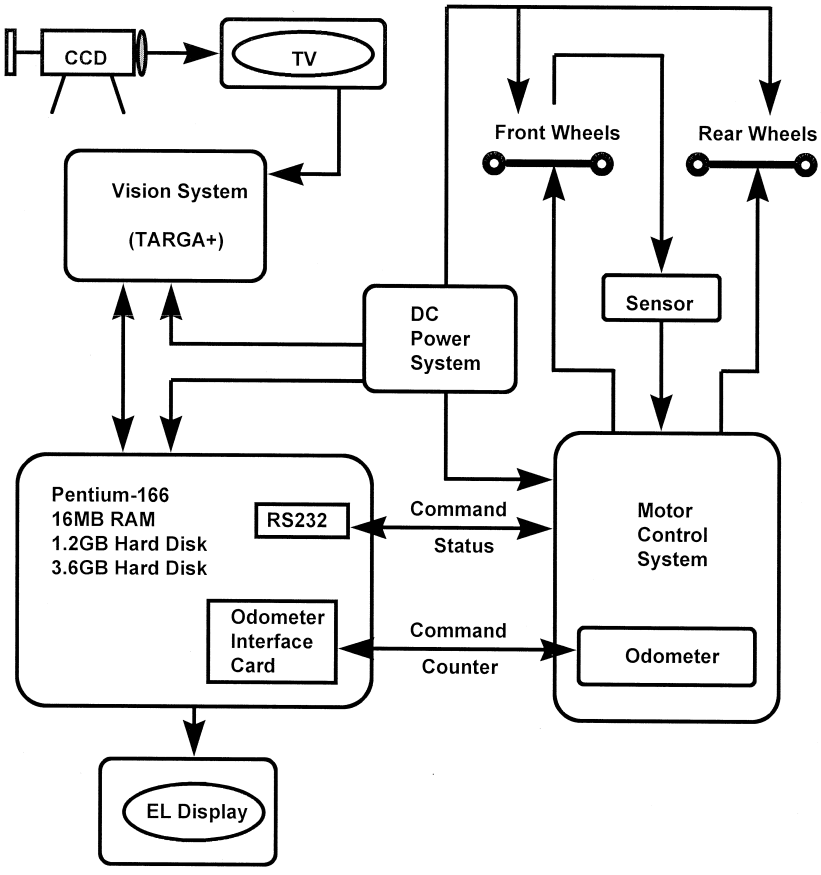

Fig. 2. System structure of prototype ALV.

era, a TV monitor, and a Targa Plus color image frame grabber. The central processor Pentium-166 has an RAM with 16 megabytes, one floppy disk, a 1.2-gigabyte hard disk, a 3.6-gigabyte hard disk, and an EL slim display. The motor control system includes a main control board with an Intel 8085 controller, a motor driver, and two motors. The power of the system is supplied by a battery set including two 12-V power source, each being divided into various voltages using a DC-to-DC converter set to provide power to the ALV components.

The remainder of this paper is organized as follows. In Section 2, the details of the proposed vision-based obstacle detection method is described. In Section 3, the proposed obstacle avoidance method is introduced in detail. The descriptions of the employed image processing techniques and experimental results are included in Section 4. Finally, some conclusions are stated in Section 5.

2. Proposed vision-based obstacle detection method

Basically, three types of objects may be extracted from the road image in this approach, which are: (1) *type-1* objects: the objects that newly appear in the road image of the current cycle, which will be judged to be obstacles or not in the next cycle; (2) *type-2* objects: the objects that appear in the road image of the previous cycle, which are judged to be obstacles or not in the current cycle; and (3) *type-3* objects: the objects that have been decided to be obstacles or non-obstacles in the current or subsequent cycles.

Initially, no objects appear in the road image, and we employ the approach proposed in Chen and Tsai [35], which utilizes color information clustering and combined line and road following techniques, to guide the ALV to follow the road. When new objects appear in the road image, all of them are regarded as type-1 objects and whether they are obstacles will be judged in the next cycle. In the next cycle, these type-1 objects become type-2 objects and whether they are obstacles are judged in this cycle. After the judgment, the type-2 objects become type-3 objects. Type-3 objects may still appear in the images of several subsequent cycles. We then compute a navigation point and drive the ALV toward the point such that the ALV can avoid collision with the type-3 objects that have been decided to be obstacles. The entire process is repeated one cycle after another. Note that type-1, type-2, and type-3 objects may appear in the image simultaneously, and some type-3 objects may disappear from the road image in subsequent cycles.

To judge whether one object is an obstacle or not in the next cycle, we first extract the shape boundary of the object from the road image of the current cycle. After the translation from the ALV location in the current cycle to that in the next cycle is esti-

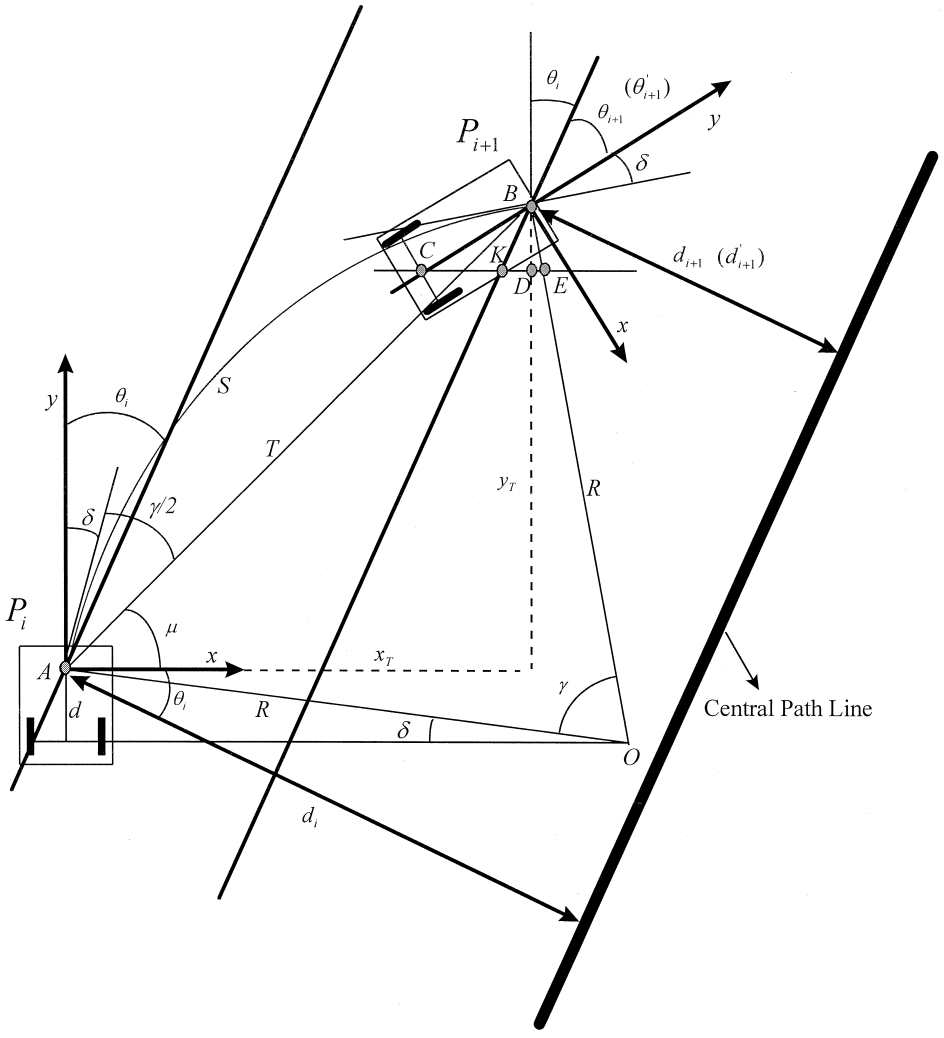

Fig. 3. Illustration of how we estimate the translation vector between two continuous ALV locations.

mated, the position of the object shape in the image of the next cycle is predicted using coordinate transformation techniques, based on the assumption that the height of the object is zero. The predicted object shape is then matched with the extracted shape boundary of the object in the image of the next cycle to decide whether the object is a non-obstacle, a static obstacle, or a moving obstacle. We use the DWC [34] as the similarity measure between the two shapes for shape matching.

In the following, we state first the estimation of the translation between two continuous ALV locations, then the prediction process of the object shape in the image of the next cycle, followed by the shape matching process.

2.1. Estimation of translation between two continuous ALV locations

When the ALV keeps driving on a road, we use the approach proposed in Ref. [35] to guide the ALV to follow the road. In this approach, the ALV location on the road in cycle i is represented by two parameters $(d_i,\ \theta_i)$, where d_i is the distance of the ALV to the central path line on the road and θ_i is the pan angle of the ALV relative to the road direction (positive to the left). Let P_i denote the obtained ALV location $(d_i,\ \theta_i)$ in cycle i and P_{i+1} denote the obtained ALV location $(d_{i+1},\ \theta_{i+1})$ in cycle $i+1$. What we desire to know is the translation vector from P_i to P_{i+1}, denoted by T, which can be derived in terms of d_i, θ_i, d_{i+1}, and θ_{i+1}. Without loss of generality, we first assume that the ALV turns to the right from P_i to P_{i+1}, i.e., $\theta_{i+1} < \theta_i$. Then as shown in Fig. 3, where $\theta_i > 0$, $\theta_{i+1} < 0$, and $d_i < d_{i+1} < 0$, the angle $\angle$CBD can be expressed as

$$\begin{aligned}\angle\mathrm{CBD} &= \frac{\pi}{2} - \delta - \angle\mathrm{DBE} \\ &= \frac{\pi}{2} - \delta - \left(\frac{\pi}{2} - \angle\mathrm{DEB}\right) \\ &= -\delta + \delta + \gamma = \gamma, \end{aligned} \tag{1}$$

where δ is the turn angle of the front wheels. Alternatively, $\angle$CBD can be expressed as

$$\angle\mathrm{CBD} = \angle\mathrm{CBK} + \angle\mathrm{KBD} = -\theta_{i+1} + \theta_i. \tag{2}$$

So, angle γ can be determined by

$$\gamma = \theta_i - \theta_{i+1}. \tag{3}$$

The length of vector $\boldsymbol{T}$ can be solved to be

$$l_{\boldsymbol{T}} = \frac{d_{i+1} - d_i}{\cos(\mu + \theta_i)}. \tag{4}$$

By the basic kinematics of the ALV, the rotation radius R can be found to be

$$R = d/\sin\delta = d/\sin\left(\frac{\pi}{2} - \mu - \frac{\gamma}{2}\right), \tag{5}$$

where d is the distance between the front wheels and the rear wheels, and $l_{\boldsymbol{T}}$ can be expressed as

$$l_{\boldsymbol{T}} = R\sqrt{2(1-\cos\gamma)} \tag{6}$$

according to Ref. [36]. From Eqs. (4) and (6), we get

$$\frac{d_{i+1} - d_i}{\cos(\mu + \theta_i)} = \frac{d\sqrt{2(1-\cos\gamma)}}{\sin\left(\frac{\pi}{2} - \mu - \frac{\gamma}{2}\right)}. \tag{7}$$

Replacing γ by $\theta_i - \theta_{i+1}$, the direction of $\boldsymbol{T}$ is determined by the angle

$$\mu = \tan^{-1}\left(\frac{A\cos\theta_i - B\cos\left(\frac{\theta_i - \theta_{i+1}}{2}\right)}{A\sin\theta_i - B\sin\left(\frac{\theta_i - \theta_{i+1}}{2}\right)}\right), \tag{8}$$

where

$$\begin{aligned} A &= \sqrt{2d^2\left(1 - \cos(\theta_i - \theta_{i+1})\right)}, \\ B &= d_{i+1} - d_i. \end{aligned} \tag{9}$$

Similarly, if the ALV turns to the left from P_i to P_{i+1}, i.e., if $\theta_{i+1} > \theta_i$, the direction of $\boldsymbol{T}$ can be determined by the angle

$$\mu = \tan^{-1}\left((-1)\frac{A\cos\theta_i + B\cos\left(\frac{\theta_i - \theta_{i+1}}{2}\right)}{A\sin\theta_i + B\sin\left(\frac{\theta_i - \theta_{i+1}}{2}\right)}\right). \tag{10}$$

Then the components of the translation vector $\boldsymbol{T} = (x_{\boldsymbol{T}}, y_{\boldsymbol{T}})$ are solved to be

$$\begin{aligned} x_{\boldsymbol{T}} &= l_{\boldsymbol{T}}\cos\mu, \\ y_{\boldsymbol{T}} &= l_{\boldsymbol{T}}\sin\mu. \end{aligned} \tag{11}$$

The above vision-based kinematic model is used only when no location error exists during ALV navigation. If location error exists during ALV navigation, the vision-based kinematic model may be unsuitable for use and an additional control-based kinematic model is used to compensate for the deficiency of the original vision-based model. The location error mainly results from the wheel slippage, the unflatness of the road surface, and/or the coordinate transformations, etc. Fig. 4 illustrates one case of the location error, where the ALV location in cycle i and $i+1$ are $(0,0)$ and $(d,0)$, respectively, for $d \neq 0$. Since the ALV trajectory between two continuous navigation cycles is assumed to be a circle, the ALV can never reach the location $(d,0)$ in cycle $i+1$ and location error will exist in cycle $i+1$. If the two continuous ALV locations $(0,0)$ and $(d,0)$ are used to find their translation based on the vision-based kinematic model described above, an unreasonable solution will be produced. For this, the following navigation checking rule is used to decide whether the vision-based kinematic model is applicable.

Navigation checking rule:

$$
\begin{aligned}
&\text{if } \theta_i < 0, \text{ then if } \delta < 0, \text{ then } \theta_{i+1} < \theta_i \text{ and } d_{i+1} > d_i \\
&\quad \text{else if } \delta > 0, \text{ then } \theta_{i+1} > \theta_i \\
&\quad \text{else } (\delta = 0)\, \theta_{i+1} = \theta_i \text{ and } d_{i+1} > d_i \\
&\text{else if } \theta_i > 0, \text{ then if } \delta > 0, \text{ then } \theta_{i+1} > \theta_i \text{ and } d_{i+1} < d_i \\
&\quad \text{else if } \delta < 0, \text{ then } \theta_{i+1} < \theta_i \\
&\quad \text{else } (\delta = 0)\, \theta_{i+1} = \theta_i \text{ and } d_{i+1} < d_i \\
&\text{else } (\theta_i = 0) \text{ if } \delta < 0, \text{ then } \theta_{i+1} < \theta_i \text{ and } d_{i+1} > d_i \\
&\quad \text{else if } \delta > 0, \text{ then } \theta_{i+1} > \theta_i \text{ and } d_{i+1} < d_i \\
&\quad \text{else } (\delta = 0)\, \theta_{i+1} = 0 \text{ and } d_{i+1} = d_i,
\end{aligned}
\tag{12}
$$

where the distances d_i and d_{i+1} are positive to the right relative to the central path line, the pan and turn angles θ_i, θ_{i+1}, and δ are positive to the left relative to the road direction, and the value of δ can be obtained by checking the ALV control system.

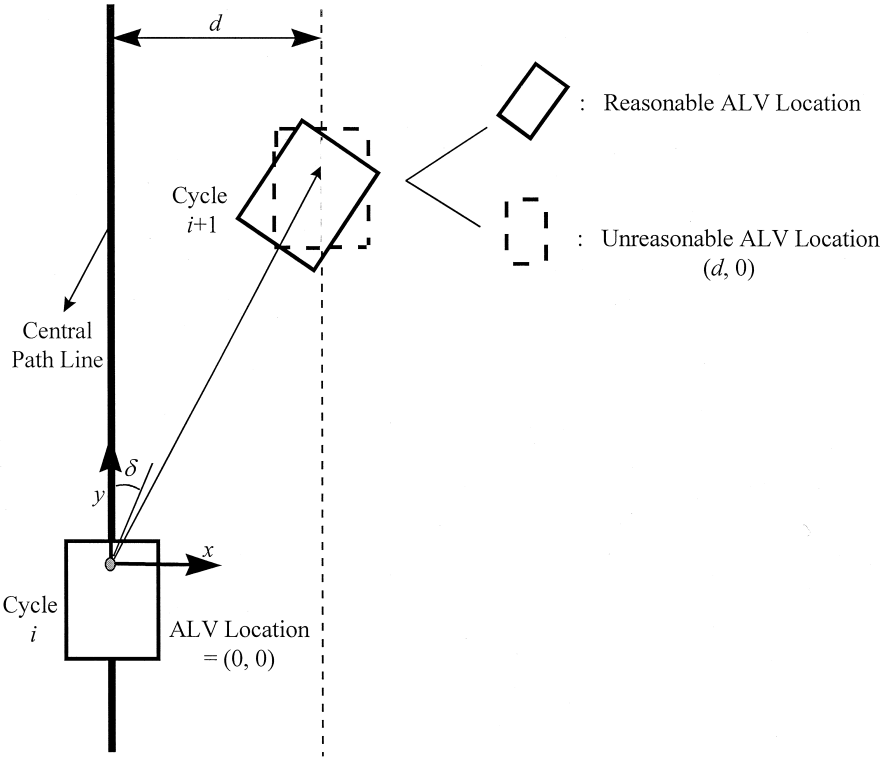

Fig. 4. Illustration of one case of location error.

If the values of d_i, θ_i, d_{i+1}, θ_{i+1}, and δ satisfy the above navigation checking rule, they are used to compute the translation vector from P_i to P_{i+1} based on the vision-based kinematic model. Otherwise, the ALV location (d_{i+1}, θ_{i+1}) in P_{i+1} is unreasonable and a control-based kinematic model is proposed, which uses the ALV control information to compute the translation vector. The computation process based on the control-based kinematic model is described as follows. As shown in Fig. 3, let S denote the travelled distance from P_i to P_{i+1}, which can be obtained from the counter of the odometer in the ALV control system. Then, angle γ can be expressed by

$$\gamma = S/R. \tag{13}$$

Since R can be determined by Eq. (5), where δ can be obtained by checking the ALV control system, angle γ can be determined accordingly, and the length $l_{\boldsymbol{T}}$ of vector $\boldsymbol{T}$ expressed in Eq. (6) can thus be solved. Since the direction of $\boldsymbol{T}$ is determined by

$$\mu = \frac{\pi}{2} - \delta - \frac{\gamma}{2}, \tag{14}$$

the components of the translation vector expressed in Eq. (11) can be solved.

The computed translation vector is then used together with the ALV location (d_i, θ_i) in P_i to estimate a reasonable ALV location in P_{i+1} when the control-based kinematic model is used. Let $(d'_{i+1}, \theta'_{i+1})$ denote the estimated ALV location in P_{i+1}. Then, if $\delta < 0$, as illustrated in Fig. 3, d'_{i+1} and θ'_{i+1} are solved to be

$$\begin{aligned} d'_{i+1} &= d_i + l_{\boldsymbol{T}} \cos(\mu + \theta_i) \\ \theta'_{i+1} &= \theta_i - \gamma. \end{aligned} \tag{15}$$

Otherwise $(\delta \geq 0)$, d'_{i+1} is solved by

$$d'_{i+1} = d_i - l_{\boldsymbol{T}} \cos(\mu - \theta_i), \tag{16}$$

and θ'_{i+1} is identical to that expressed in Eq. (15).

Note that if $d_{i+1} = d_i$, $\theta_{i+1} = \theta_i = 0$, and $\delta = 0$, we cannot derive the translation vector using the vision-based kinematic model even when the navigation from P_i to P_{i+1} is judged to be reasonable by checking the values of d_i, θ_i, d_{i+1}, θ_{i+1}, and δ in the navigation checking rule. In this case, the control-based kinematic model is used, and the components of the translation vector $\boldsymbol{T}$ from P_i to P_{i+1} are just

$$\begin{aligned} x_{\boldsymbol{T}} &= 0, \\ y_{\boldsymbol{T}} &= S. \end{aligned} \tag{17}$$

The vision-based kinematic model combined with the control-based kinematic model and the navigation checking rule enables the ALV to achieve reliable and fault-tolerant navigation.

2.2. Object shape prediction in next cycle

Several coordinate systems and coordinate transformations are used in the prediction process. 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 the CCS and the VCS can be written in terms of homogeneous coordinates [37,38] as

$$(uvw1) = (xyz1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_d & -y_d & -z_d & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{18}$$

where

$$\begin{aligned} r_{11} &= \cos\theta\cos\varphi + \sin\theta\sin\phi\sin\varphi, \\ r_{12} &= -\sin\theta\cos\phi, \\ r_{13} &= \sin\theta\sin\phi\cos\varphi - \cos\theta\sin\varphi, \\ r_{21} &= \sin\theta\cos\varphi - \cos\theta\sin\phi\sin\varphi, \\ r_{22} &= \cos\theta\cos\phi, \\ r_{23} &= -\cos\theta\sin\phi\cos\varphi - \sin\theta\sin\varphi, \\ r_{31} &= \cos\phi\sin\varphi, \\ r_{32} &= \sin\phi, \\ r_{33} &= \cos\phi\cos\varphi, \end{aligned} \tag{19}$$

and θ is the pan angle, ϕ the tilt angle, and φ the swing angle, of the camera with respect to the VCS; and (x_d, y_d, z_d) is the translation vector from the origin of the VCS to the origin of the CCS.

To predict the shape boundary of an object in the ICS in the next cycle, we first find the ICS coordinates of all the boundary points representing the shape of the object in the image of the current cycle. We then *backproject* each boundary point of the object in the ICS into the VCS, based on the assumption that the height of the boundary point in the VCS is zero, i.e., its z coordinate is zero, to obtain the 2D VCS coordinates (x, y) of the boundary point in the current cycle. By using the translation vector derived previously, the 2D VCS coordinates (x', y') of the boundary point in the next cycle can be found, as illustrated in Fig. 5, where $\theta_i > 0$, $\theta_{i+1} < 0$, and the components of the translation vector are (x_T, y_T). In the figure, the VCS coordinates (x', y') of point P in cycle $i+1$ is solved to be

$$(x'y'1) = (xy1)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_T & -y_T & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_{i+1}-\theta_i) & -\sin(\theta_{i+1}-\theta_i) & 0 \\ \sin(\theta_{i+1}-\theta_i) & \cos(\theta_{i+1}-\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

or

$$\begin{aligned} x' &= (x - x_T)\cos(\theta_{i+1}-\theta_i) \\ &\quad + (y - y_T)\sin(\theta_{i+1}-\theta_i) \\ y' &= (x_T - x)\sin(\theta_{i+1}-\theta_i) \\ &\quad + (y - y_T)\cos(\theta_{i+1}-\theta_i). \end{aligned} \tag{21}$$

After the backprojection and translation processes, we *project* the backprojected and translated boundary point in the VCS into the ICS to predict the ICS coordinates (u,w) of the boundary point in the next cycle and the prediction process is finished. The backprojection and projection principles are described as follows.

(1) Backprojection principle: As shown in Fig. 6, assume that point P in the image has the CCS coordinates $(u_P, -f, w_P)$ where (u_P, w_P) indicate the position in the image, i.e., the ICS coordinates, and f is the focus length. After backprojecting the point P in the image into the VCS, we can get a line L that passes P and the lens center O_c. Let P' denote the intersection point of this line L and the horizontal plane $z = h$. Using Eq. (18), we get the VCS coordinates (x_P, y_P, z_P) of point P in the image as

$$\begin{aligned} x_P &= u_P(\cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi) \\ &\quad + f(\sin\theta\cos\phi) + w_P(\sin\theta\sin\phi\cos\psi \\ &\quad + \cos\theta\cos\psi) + x_d, \\ y_P &= u_P(\sin\theta\cos\psi + \cos\theta\sin\phi\sin\psi) \\ &\quad - f(\cos\theta\cos\phi) - w_P(\cos\theta\sin\phi\cos\psi \\ &\quad + \sin\theta\sin\psi) + y_d, \\ z_P &= u_P(\cos\theta\sin\psi) \\ &\quad - f\sin\phi + w_P(\cos\theta\cos\psi) + z_d. \end{aligned} \tag{22}$$

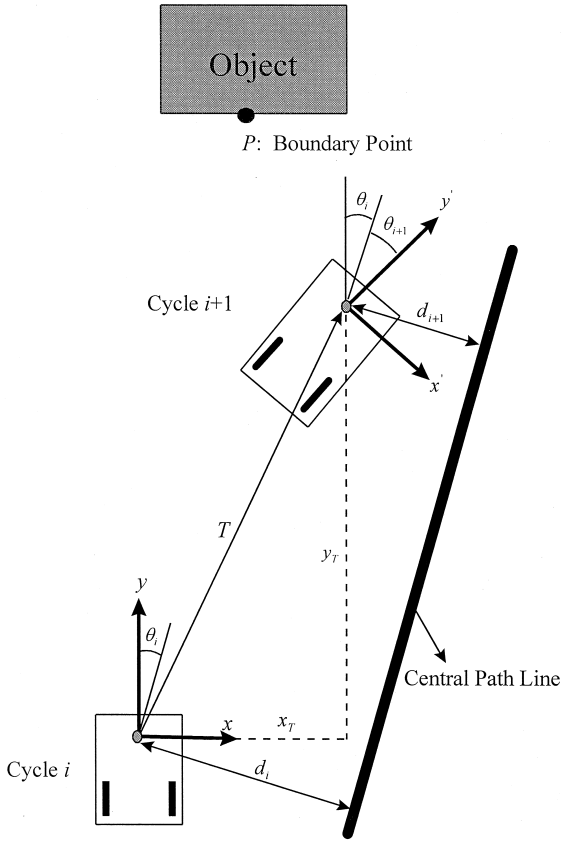

Fig. 5. Illustration of how the VCS coordinates (x, y) of some boundary point P in the next cycle are computed.

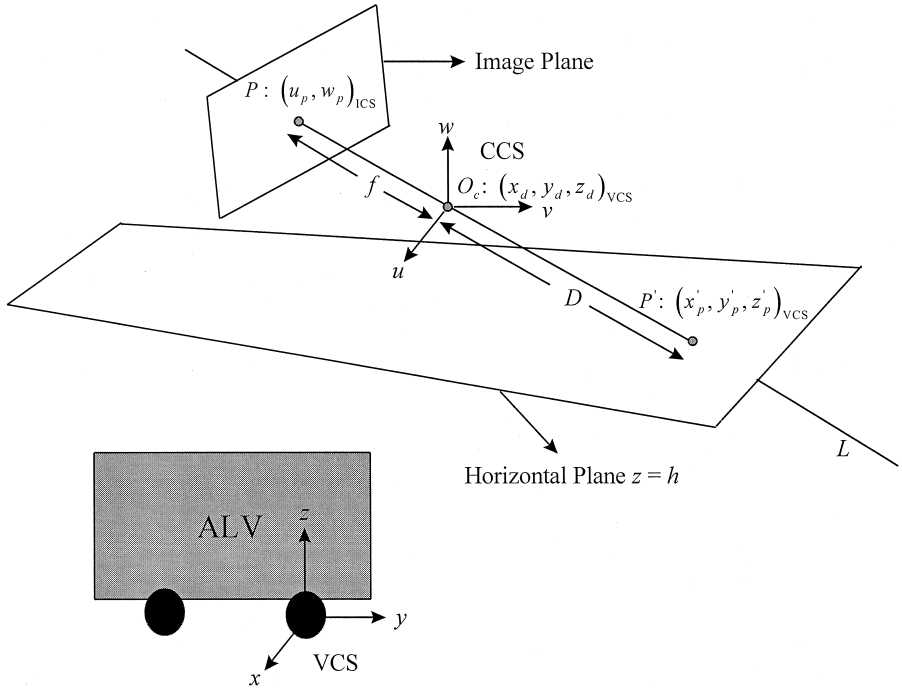

Fig. 6. Illustration of the backprojection and projection processes.

Additionally, the equation of line L can be expressed as

$$\frac{x - x_d}{x_P - x_d} = \frac{y - y_d}{y_P - y_d} = \frac{z - z_d}{z_P - z_d} = k, \tag{23}$$

where k is a constant. By substituting $z = h$ into Eq. (23), the VCS coordinates $(x'_P,\ y'_P,\ z'_P)$ of point P' can be solved to be

$$x'_P = x_d + \frac{h - z_d}{z_P - z_d}(x_P - x_d),$$

$$y'_P = y_d + \frac{h - z_d}{z_P - z_d}(y_P - y_d),$$

$$z'_P = h. \tag{24}$$

Since we backproject each boundary point in the image into the VCS using the assumption that the height of the boundary point in the VCS is zero, we substitute $h = 0$ into Eq. (24) and the desired 2D VCS coordinates $(x'_P,\ y'_P)$ of point P' are solved to be

$$x'_P = x_d - \frac{z_d(x_P - x_d)}{z_P - z_d},$$

$$y'_P = y_d - \frac{z_d(y_P - y_d)}{z_P - z_d}. \tag{25}$$

(2) Projection principle: As shown in Fig. 6, assume that point P' has the VCS coordinates $(x'_P,\ y'_P,\ z'_P)$. After projecting the point P' in the VCS into the ICS, we get its corresponding space point P in the ICS. Since the lens center O_c has the VCS coordinates $(x_d,\ y_d,\ z_d)$ as given in Eq. (18), the distance between P' and O_c, denoted as D, is calculated to be

$$D = \sqrt{(x'_P - x_d)^2 + (y'_P - y_d)^2 + (z'_P - z_d)^2}. \tag{26}$$

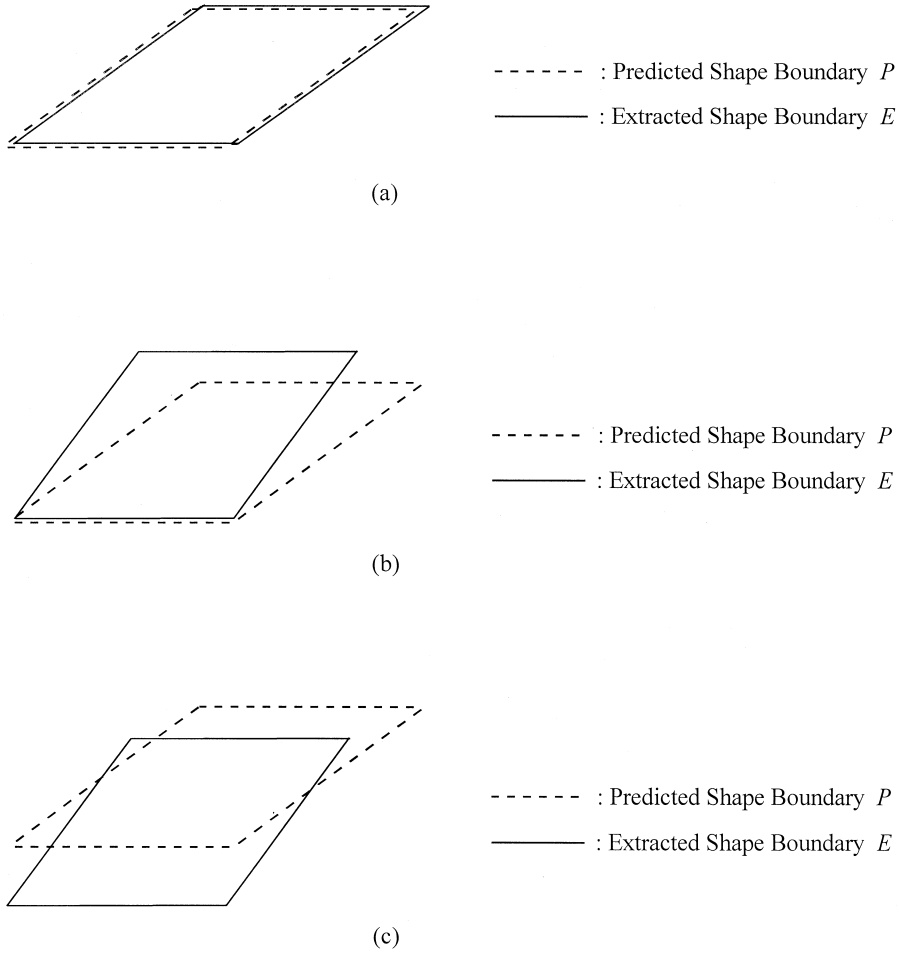

Fig. 7. Illustration of how to decide using the DWC correlation measure whether an object is an obstacle. (a) A non-obstacle is detected. (b) A static obstacle is detected. (c) A moving obstacle is detected.

Let (x_P, y_P, z_P) denote the VCS coordinates of the corresponding point P in the image.Then the following equation is satisfied:

$$\frac{x_P - x'_P}{x_d - x'_P} = \frac{y_P - y'_P}{y_d - y'_P} = \frac{z_P - z'_P}{z_d - z'_P} = \frac{f + D}{D} = K, \tag{27}$$

where f is the focus length and K is a constant. The VCS coordinates of point P can be solved accordingly to be

$$\begin{aligned} x_P &= K(x_d - x'_P) + x'_P, \\ y_P &= K(y_d - y'_P) + y'_P, \\ z_P &= K(z_d - z'_P) + z'_P. \end{aligned} \tag{28}$$

Since we assume that the height of the boundary point in the VCS is zero, we substitute $z'_P = 0$ into the above equations for further simplification. Using the transformation between the CCS and the VCS described in Eq. (18), we can get the CCS coordinates $(u_P, -f, w_P)$ of point P, and so the desired coordinates (u_P, w_P) of point P in the image.

2.3. Shape matching

To judge whether an object O is an obstacle in the current cycle, we extract its shape (represented by the shape boundary points) in the image of the previous cycle, and predict its shape in the image of the current cycle using coordinate transformation

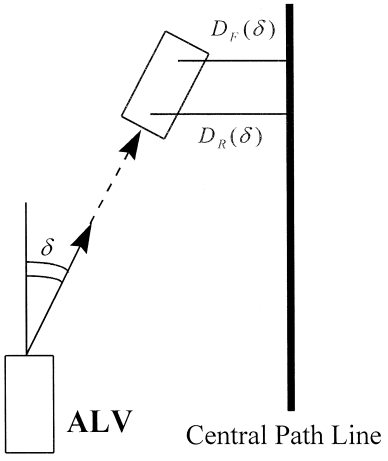

Fig. 8. Illustration of closeness distance measure $L(\delta)=1/\{1+[D_F(\delta)]^2+[D_R(\delta)]^2\}$.

techniques based on an assumption stated and derived previously. Let P denote the predicted shape boundary and E denote the extracted shape boundary of the object in the image of the current cycle. To match P and E, we overlap them and compute the measure of the DWC [34] to check the similarity between them. First, the minimum distance d_b of a boundary point b in P or E is defined to be the Euclidean distance between b and its closest point in the other shape. The weight w_b^k of b is defined to be

$$w_b^k=\begin{cases}1/\left(d_b^2+1\right) & \text{if } 0\le d_b\le k,\\ 0 & \text{otherwise,}\end{cases} \tag{29}$$

where k is a constant that limits the distance within which the closest point of b is searched for. Then the DWC is defined to be

$$\mathrm{DWC}^k(P,E)=\frac{1}{2}\left(\frac{1}{N_P}\sum_{i\in P}w_i^k+\frac{1}{N_E}\sum_{j\in E}w_j^k\right), \tag{30}$$

where N_P and N_E are the total numbers of the boundary points in P and E, respectively. It can be easily verified that $0\le \mathrm{DWC}^k(P,E)\le 1$. The value of $\mathrm{DWC}^k(P,E)$ is then checked to judge whether the object O is an obstacle. If the value is greater than a certain threshold value, say TH_1, where $0<\mathrm{TH\_1}<1$, it is decided that O is not an obstacle because P and E are strongly similar, as illustrated in Fig. 7(a). If the value is smaller than TH_1 and greater than another threshold value, say TH_2, where $0<\mathrm{TH\_2}<\mathrm{TH\_1}$, it is decided that O is a static obstacle because P and E are partially similar, as illustrated in Fig. 7(b). Finally, if the value is smaller than TH_2, it is decided that O is a moving obstacle because P and E are strongly dissimilar, as illustrated in Fig. 7(c).

3. Proposed obstacle avoidance method

3.1. Navigation point selection

If no obstacle appears on the road ahead, we drive the ALV to follow the central path line on the road using a closeness distance measure from the ALV to the central path line proposed by Cheng and Tsai [36]. The measure is defined as

$$L(\delta)=\frac{1}{1+\left[D_F(\delta)\right]^2+\left[D_R(\delta)\right]^2}, \tag{31}$$

where D_F and D_R are the corresponding distances from the front and the rear wheels of the ALV to the central path line after the ALV traverses a distance with the turn angle δ, as illustrated in Fig. 8. A larger value of L means that the ALV is closer to the

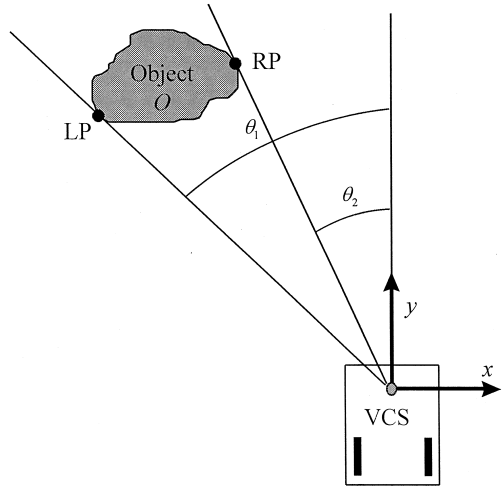

Fig. 9. Illustration of the definitions of the LP and the RP of one object.

path. It is easy to verify that $0 < L \leq 1$, and that $L = 1$ if and only if both the front wheels and the rear wheels of the ALV are located just right on the path.

To find the turn angle of the front wheel to drive the ALV as close to the central path line as possible, a range of possible turn angles is searched. An angle is hypothesized each time, and the value of L is calculated accordingly. The angle that produces the maximal value of L is then used as the desired turn angle.

It should be mentioned that allowing the ALV a larger angle to turn in a session of turn drive does not mean that better navigation can be achieved. It may cause serious twisting. On the other hand, a smaller range of turn angles may bring the ALV slightly closer to the central path line. Hence, the largest angle allowing the ALV to turn is a tradeoff between smoothness of navigation and closeness to the central path line.

If obstacles appear on the road ahead, we compute a safe navigation point and drive the ALV toward this point for obstacle avoidance. The navigation point is selected as follows. For each boundary point of an obstacle, we compute its *corresponding angle* that is defined as the angle between the *y*-axis of the VCS and the line segment which connects the boundary point and the origin of the VCS. This angle is positive to the left with respect to the *y*-axis of the VCS. We define the *left extreme point LP* as the boundary point whose corresponding angle is the largest, and define the *right extreme point RP* as the boundary point whose corresponding angle is the smallest. Fig. 9 shows the LP and the RP of an obstacle *O*, where the corresponding angle θ_1 of the LP is the largest and the corresponding angle θ_2 of the RP is the smallest. Then the proposed navigation point selection method is illustrated in Fig. 10, where there are three obstacles O_1, O_2, and O_3 ahead of the ALV on the road, and LP_i and RP_i are the LP and

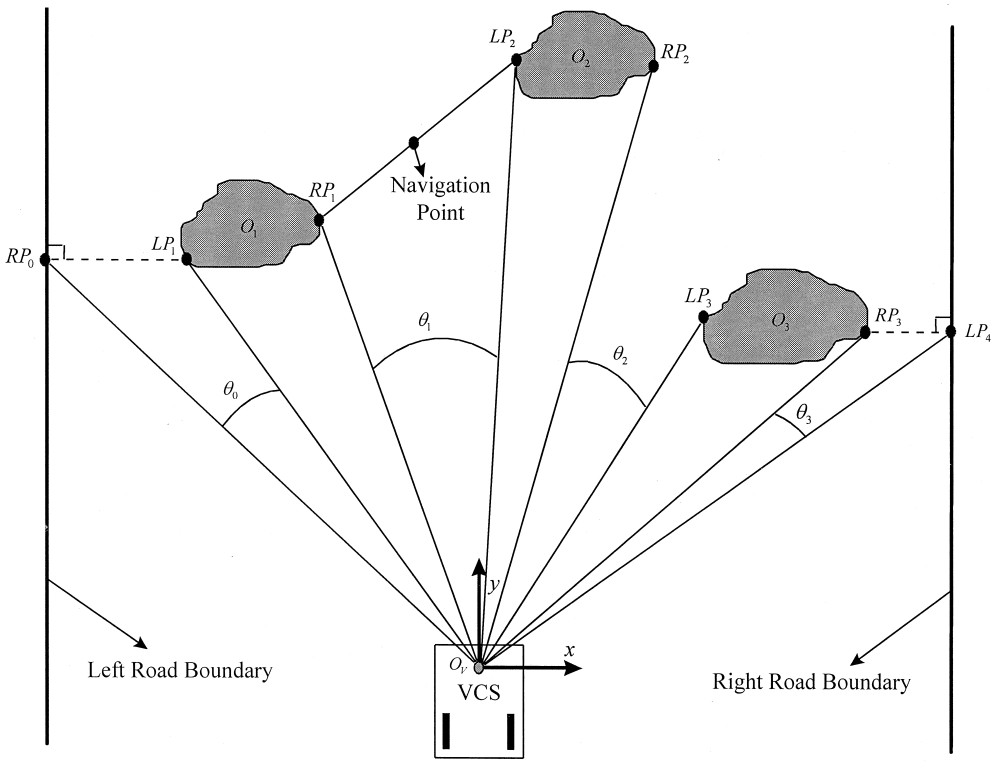

Fig. 10. Illustration of how the navigation point is chosen when obstacles appear on the road ahead.

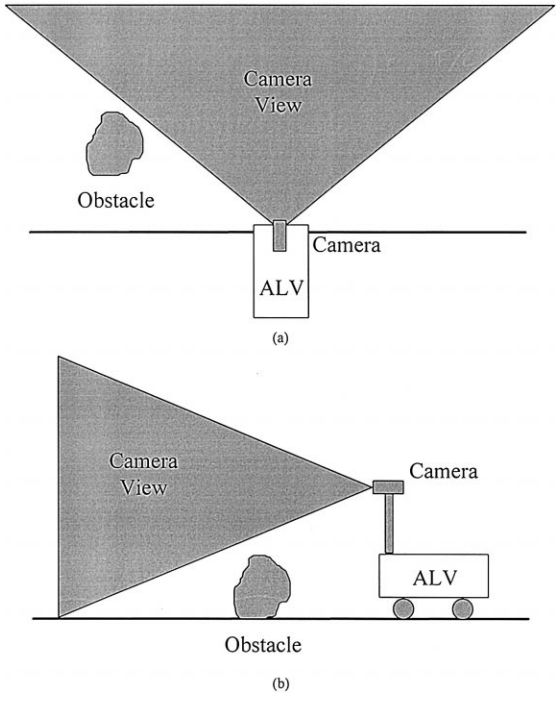

Fig. 11. Illustration of why obstacles disappear from the image due to the angle of the camera view when they are still ahead of the ALV. (a) Horizontal camera view. (b) Vertical camera view.

the RP of obstacle O_i, respectively, for $i = 1$, 2, and 3.

In the figure, RP_0 is the point on the left road boundary that is the closest to LP_1, and LP_4 is the point on the right road boundary that is the closest to RP_3. We then compute the angle between the line segment connecting RP_i and O_V (the origin of the VCS) and the line segment connecting LP_{i+1} and O_V, for $i = 0$, 1, 2, and 3. From all of the angles so computed, we find the largest one and let it be θ_k, $k = 0$, 1, 2, or 3. Then the middle point of the line segment connecting RP_k and LP_{k+1} is chosen as the navigation point. It can be seen from the figure that θ_1 is the largest angle, so the navigation point is set as the middle point of the line segment connecting RP_1 and LP_2. This way of choosing the navigation point can be further applied to the case that there are more obstacles than three on the road ahead.

If all of the obstacles ahead of the ALV appear in the image simultaneously in the current cycle, they are extracted from the image to find their LPs and RPs, and the navigation point in this cycle is computed using these LPs and RPs. In the next cycle, if all of the obstacles ahead of the ALV still appear in the image simultaneously, they are extracted and the navigation point in this cycle is computed in the same way as described above. But, if some of these obstacles, which are still ahead of the ALV, disappear from the image due to the angle of the camera

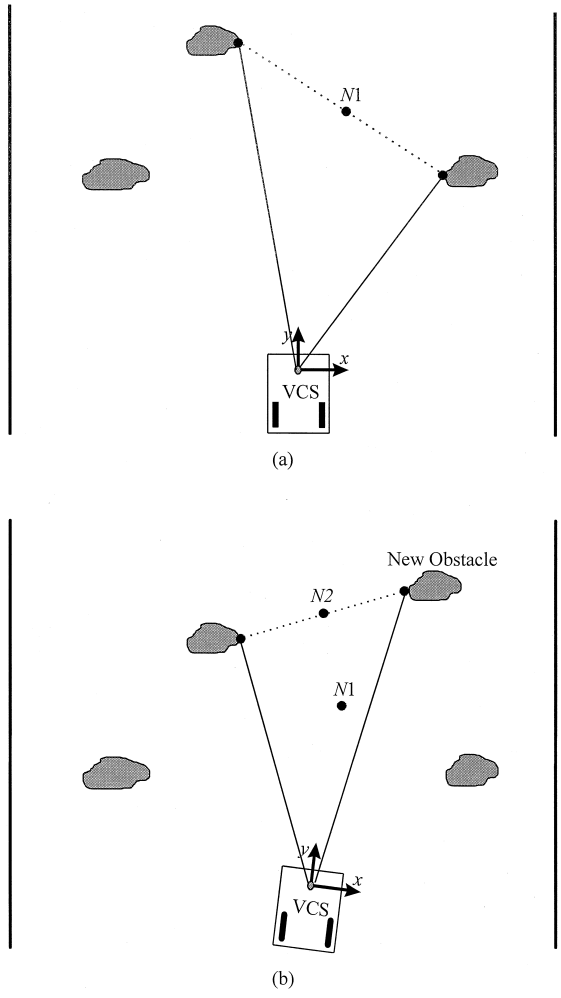

Fig. 12. Illustration of why the chosen navigation point varies during ALV navigation. (a) Point $N1$ is chosen as the navigation point. (b) Point $N2$ is chosen as the navigation point since a new obstacle appears ahead of the ALV after the ALV travels a certain distance.

view as shown in Fig. 11, they cannot be extracted from the image to find their LPs and RPs. At this moment, their LPs and RPs in this cycle are predicted using their extracted LPs and RPs in the previous cycle. The prediction process has been stated previously in Section 2.2. These predicted LPs and RPs together with the extracted LPs and RPs of the obstacles that appear in the image in this cycle are then used to compute the navigation point in this cycle.

In subsequent navigation cycles, the navigation point is computed cyclically until no obstacle is ahead of the ALV. At this moment, the ALV heads back to the central path line using the line following scheme described previously, and the obstacle avoidance process is finished. Note that the chosen navigation point may vary during ALV navigation since new obstacles may appear in the image during navigation, as illustrated in Fig. 12. In (a), three obstacles are ahead of the ALV and point $N1$ is chosen as the navigation point. After the ALV travels a certain distance, a new obstacle appears ahead of the ALV and the chosen navigation point is changed to point $N2$ as shown in (b).

3.2. Turn angle computation

After the navigation point is chosen, the ALV turns an angle to approach this point for safe navigation. The turn angle computation is illustrated in Fig. 13, where P_n: (x_n, y_n) is the navigation point, l_T is the distance between P_n and O_V, R is the rotation radius, and δ is the turn angle of the front wheels we want to compute. From Eqs. (5) and (6), we can obtain the following equation:

$$\tan\delta = \frac{\cos\mu}{\sin\mu + \dfrac{l_T}{2d}}. \tag{32}$$

Since l_T, $\sin\mu$, and $\cos\mu$ can be solved by

$$l_T = \sqrt{x_n^2 + y_n^2}, \tag{33}$$

$$\sin\mu = \frac{y_n}{\sqrt{x_n^2 + y_n^2}}, \tag{34}$$

$$\cos\mu = \frac{x_n}{\sqrt{x_n^2 + y_n^2}}, \tag{35}$$

the desired turn angle of the front wheels is solved accordingly to be

$$\delta = \tan^{-1}\left(\frac{2\,dx_n}{x_n^2 + y_n^2 + 2\,dy_n}\right). \tag{36}$$

3.3. Precise ALV location estimation

The ALV keeps driving 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 certain distance, and the current ALV location cannot be found by vision-based information. To overcome this difficulty, the system uses the ALV control information to estimate the current ALV location. Let P_i denote the ALV location at the time instant of image taking, and P_i' denote the current ALV location after the ALV has travelled a certain distance. Then, as described in Section 2.1, the translation vector T from P_i to P_i' can be found by using the travelled distance S and the pan angle δ of the front wheels of the ALV from P_i to P_i', and the current ALV location P_i' can thus be estimated by using P_i and T.

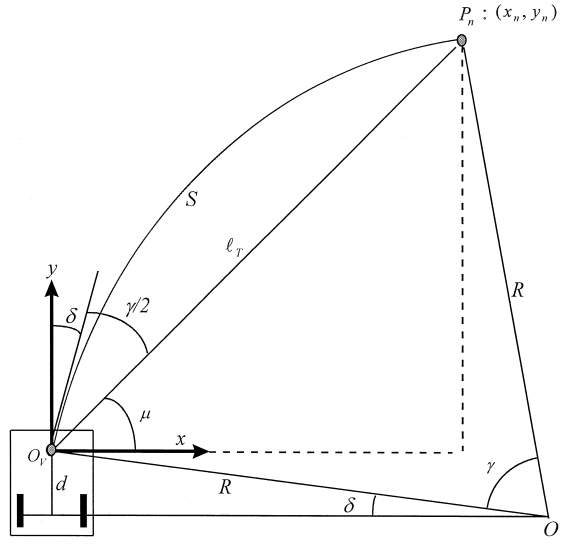

Fig. 13. Illustration of how the turn angle of the front wheels of the ALV is computed.

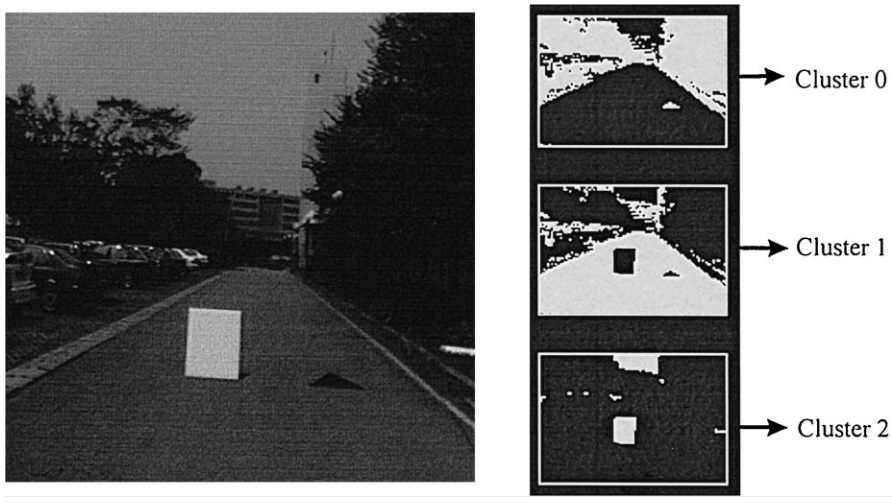

Fig. 14. A real color road image and its clustering result.

At the end of each navigation cycle, the actual current ALV location can be estimated as described above. The ALV then drives from the actual current ALV location toward the navigation point when obstacles appear on the road ahead, or toward the central path for line following when no obstacle appears on the road ahead.

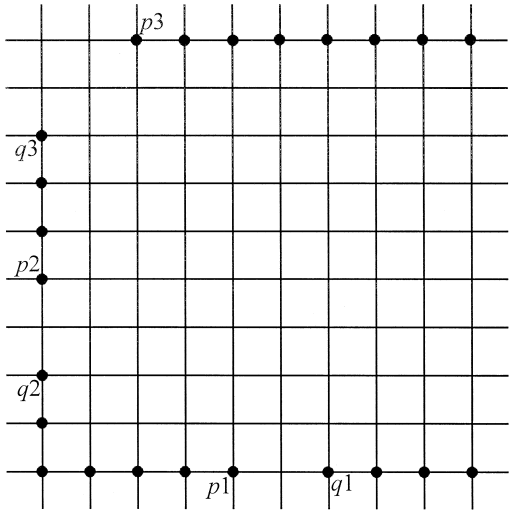

Fig. 15. A 24-connected component, where one broken pixel exists between $p1$ and $q1$ in the horizontal direction, another exists between $p2$ and $q2$ in the vertical direction, and a third one exists between $p3$ and $q3$ in the diagonal direction.

4. Image processing techniques and experimental results

4.1. Image processing techniques

We use an ISODATA clustering algorithm based on an initial-center-choosing technique [35], which can solve the problem caused by great changes of intensity in navigations, to divide the road image into three clusters according to their intensity values: (1) cluster-0: dark area, like trees and the tested black

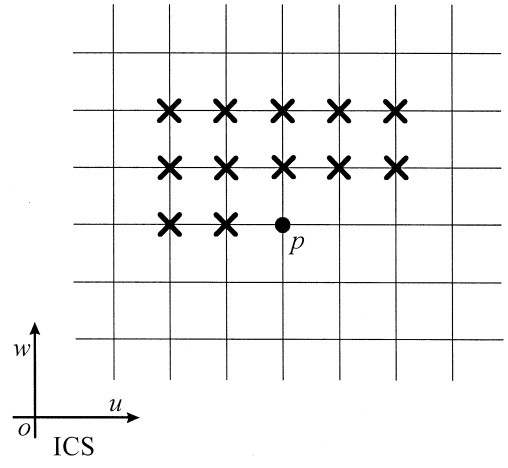

Fig. 16. A point p and its candidate 24-neighbors, which are represented by the $\times$ points.

board on the road; (2) cluster-1: gray area, coming from the main body of the road; and (3) cluster-2: bright area, like the sky and the tested white boards on the road. A real road image and its clustering

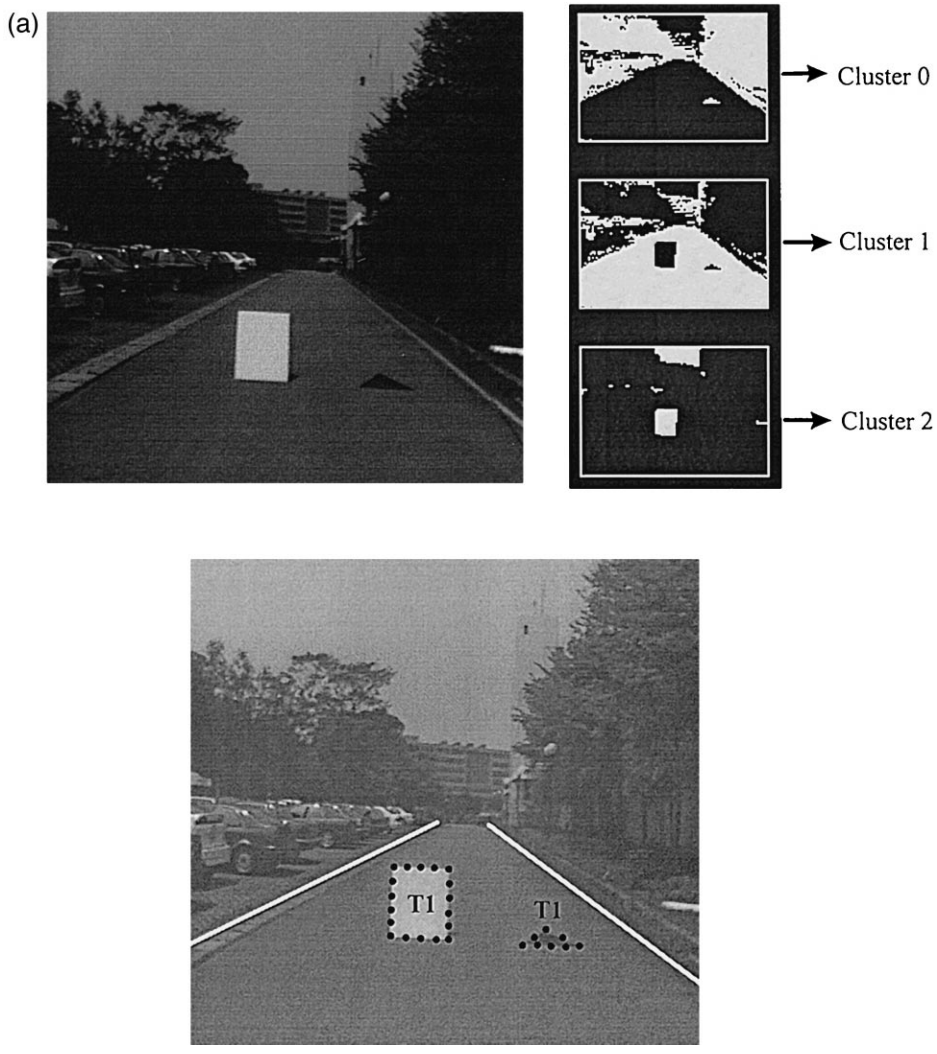

Fig. 17. A sequence of real road images, their clustering results, and the extracted and predicted boundary points of some tested objects on the road ahead, which illustrate the obstacle detection and avoidance processes when the ALV navigates along a road. (a) Two type-1 objects newly appear in the image in this cycle, and whether they are obstacles will be judged in the next cycle. (b) The two type-1 objects in (a) become type-2 objects that are judged to be obstacles or not in this cycle, where the white board is decided to be an obstacle and the black board is decided to be a non-obstacle. (c) The two type-2 objects in (b) become type-3 objects that have been decided to be obstacles or non-obstacles in the previous cycle, and two additional type-1 objects newly appear in the image that will be judged to be obstacles or not in the next cycle. (d) The two type-1 objects in (c) become type-2 objects that are decided to be non-obstacles in this cycle. Note that in this figure the two type-3 objects in (c) disappear from the image though they are still ahead of the ALV. (e) The two type-2 objects in (d) become type-3 objects and no new object appears in the image in this cycle, while the ALV has reached the navigation point and begins to head back to the central path line. (f) One type-3 object in (e) disappears from the image, and another type-3 object in (e) remains in the image in this cycle, while the ALV keeps heading back to the central path line. (g) One type-1 object newly appears in the image when the ALV navigates on the central path line, and whether it is an obstacle will be judged in the next cycle. Note that in this figure the object partially blends into the road. (h) The type-1 object in (g) becomes type-2 object that is decided to be an obstacle in this cycle.

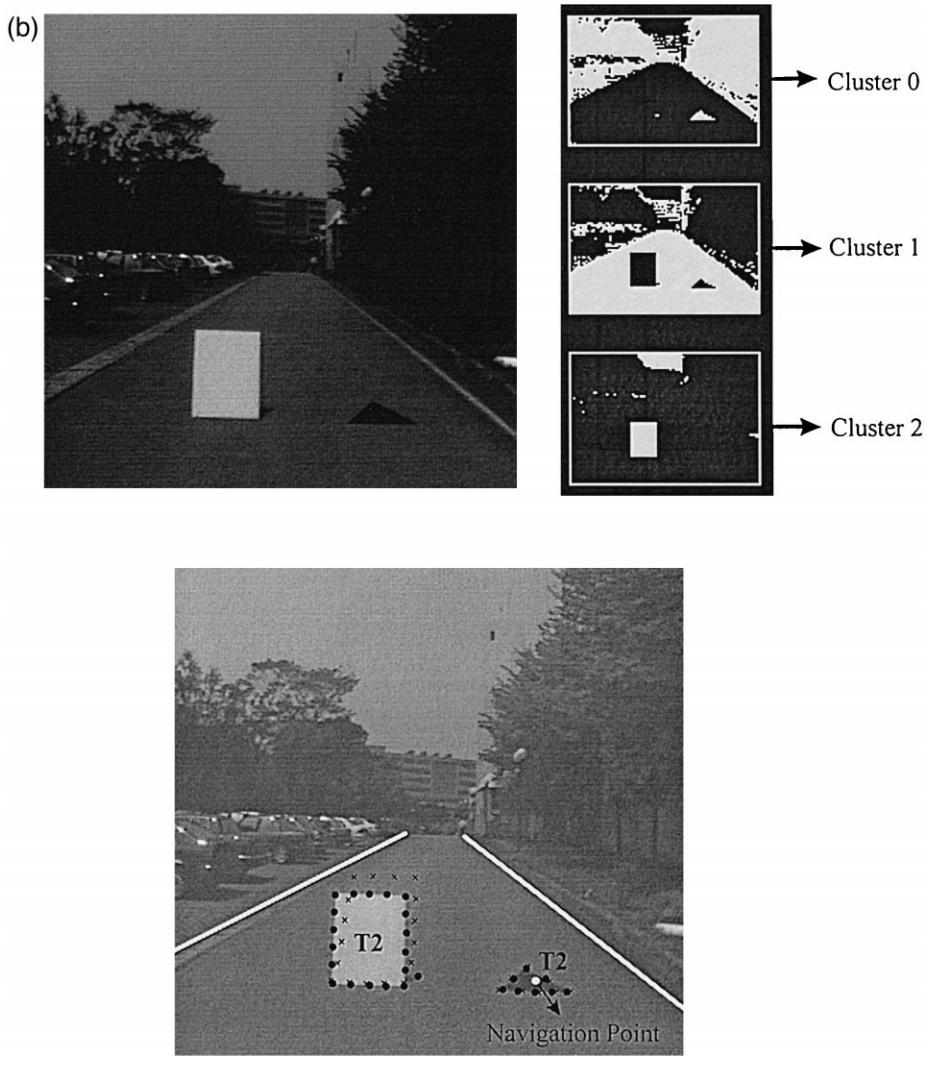

Fig. 17 (continued).

result are shown in Fig. 14, where the tested black and white boards on the road are classified into cluster-0 and cluster-2 areas, respectively.

We then extract the road surface, which is the cluster-1 feature, from the binary image to find the ALV location and the left and right boundaries on the road [35]. Next, we extract the boundary points of the objects in the area bounded by the two lines representing the two road boundaries from the binary image [39]. The process is described as follows. First, we use the *Sobel* operators, which have the advantage of both differencing and smoothing effect, to find the positions of the boundary points in the image.

Second, we scan the image to label the object boundary pixels to find *24-connected* components. Each component represents one specific object shape. The 24-connected component allows single broken points on its boundary in all directions including the horizontal, vertical, and diagonal directions as illustrated in Fig. 15, where broken pixels exist between $p1$ and $q1$ in the horizontal direction, between $p2$ and $q2$ in the vertical direction, and between $p3$ and $q3$ in the diagonal direction, and all shape points are

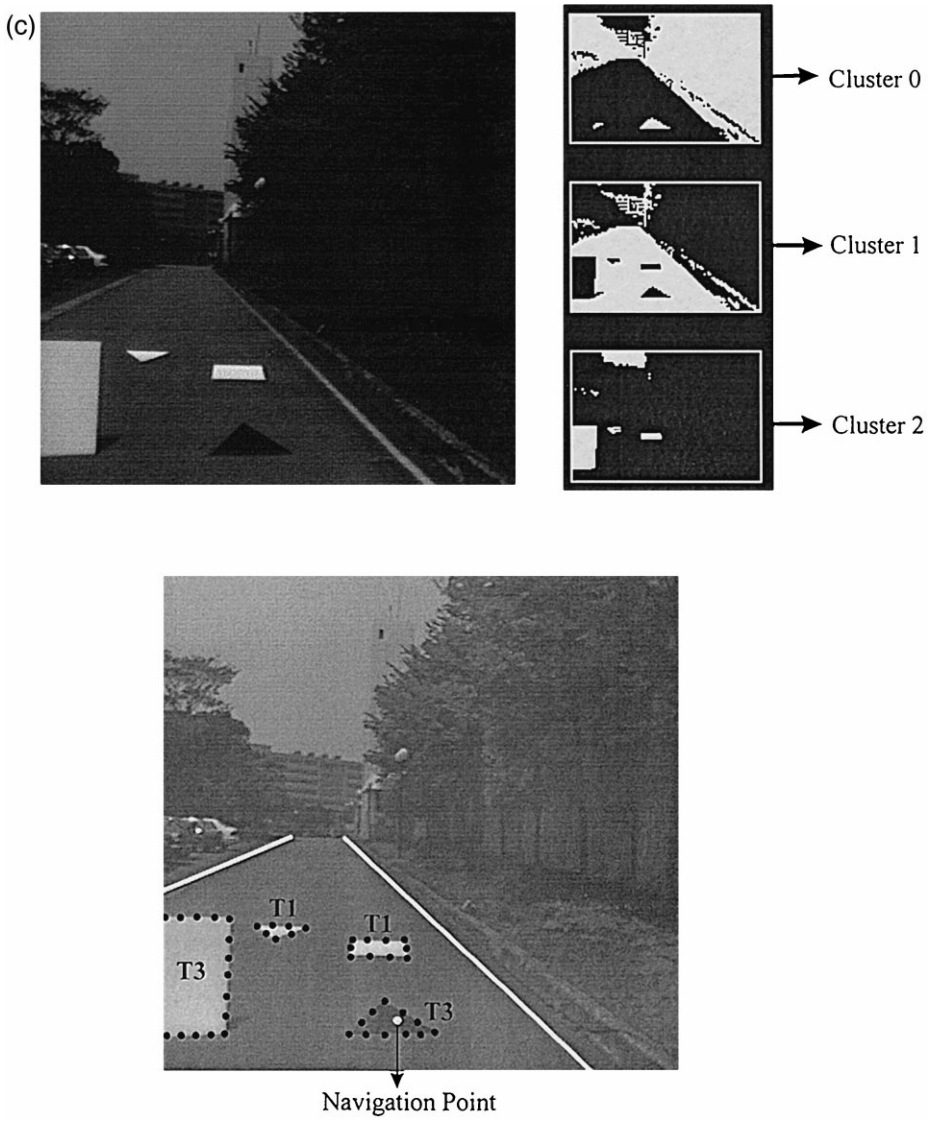

Fig. 17 (continued).

still connected. Before the labeling procedure, we first define the *candidate 24-neighbors* of some point p that have the following properties: (1) for each candidate 24-neighbor q of p, p and q are 24-connected; (2) if the u coordinate (the horizontal coordinate in the ICS) of q is smaller than that of p, then the w coordinate (the vertical coordinate in the ICS) of q is greater than or equal to that of p; and (3) if the u coordinate of q is greater than or equal to that of p, then the w coordinate of q is greater than that of p.

Fig. 16 shows a point p and its candidate 24-neighbors, which are labeled with ''x''. The labeling procedure of the 24-connected component, which is extended from that of the 8-connected component described in Ref. [39], is stated as follows.

Scan the input image pixel by pixel from left to right and from top to bottom. The nature of the scanning sequence ensures that when some pixel is examined, its candidate 24-neighbors have been examined. Let p denote the pixel examined currently in the scanning process. If p is not a boundary point,

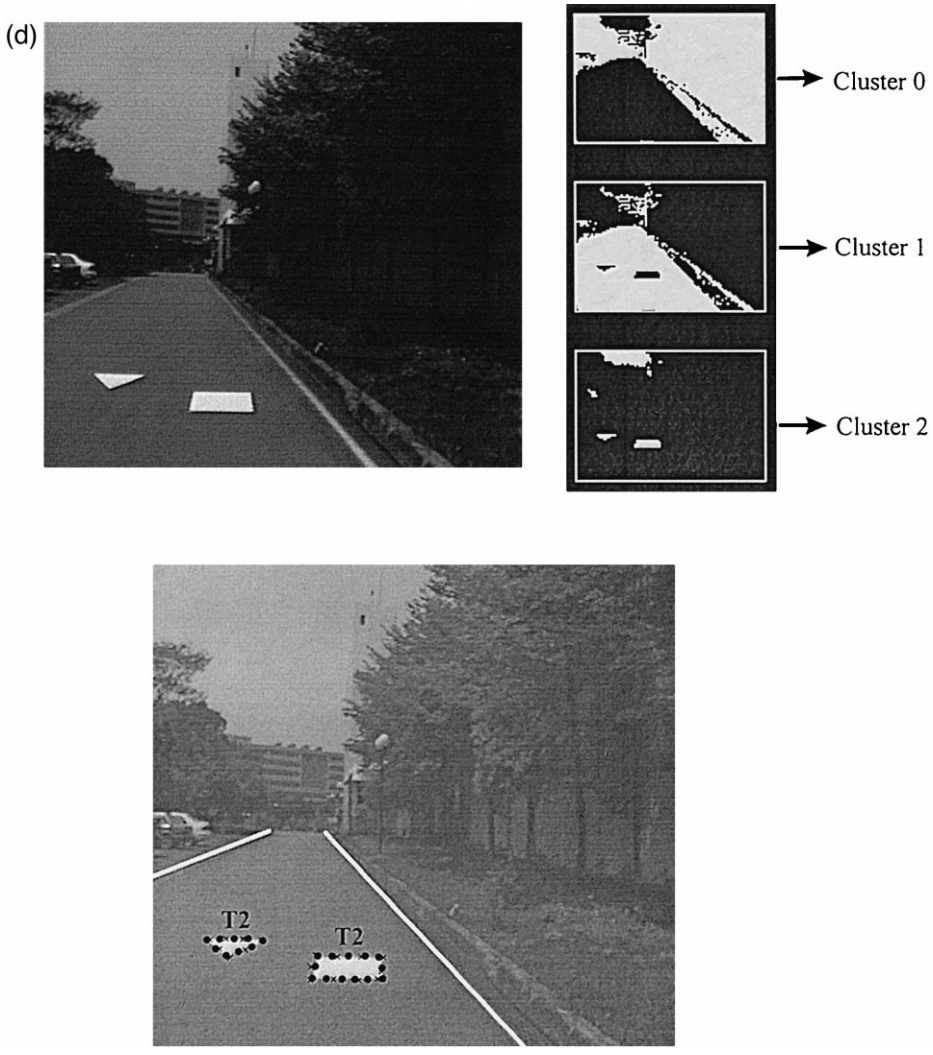

Fig. 17 (continued).

simply move p to the next scanning position. If p is a boundary point, check its candidate 24-neighbors for labeling p. If all the neighbors are not boundary points, decide that a new 24-connected component is encountered and a new label is assigned to p. If some of the neighbors are boundary points (have been labeled), assign the label of any one of these boundary points to p and make a note that the labels of these boundary points are equivalent. Then move p to the next scanning position and examine p in the same way. At the end of the scan, all boundary points have been labeled, but some of these labels may be equivalent.

Finally, the algorithm of Warshall [40], which can save much computing time, is employed to find the equivalent classes from these labels, and a unique label is assigned to each class. The image is scanned again to replace each label by the label assigned to its equivalent class. This yields a set of 24-connected components, each of which represents one specific object shape composed of the boundary points that have the same label. Fig. 17(a) shows a real road

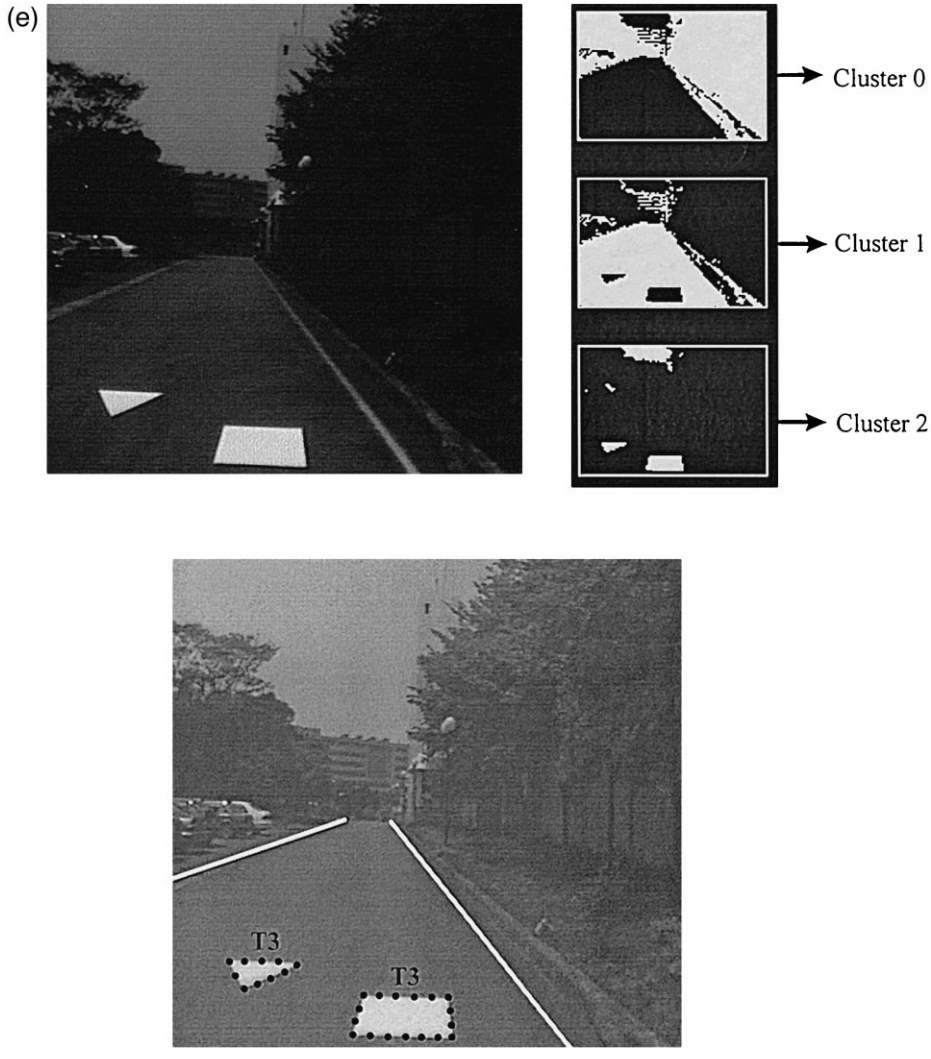

Fig. 17 (continued).

scene and the extracted boundary points, represented by the black circles, of two objects on the road.

4.2. Experimental results

Based on the proposed approach and algorithms, the prototype ALV constructed for this study is tested and found to be able to navigate safely and smoothly along part of the campus road in National Chiao Tung University. The ALV could follow the central path line when no obstacle appears on the road ahead. The ALV could decide whether the tested objects on the road ahead are obstacles during navigation, and could drive toward a safe navigation point to avoid collision with the detected obstacles. A lot of successful navigation tests confirm the feasibility of the approach. The average cycle time is about 1.0 s, and the average speed is 170 cm/s or 6.2 km/h.

Fig. 17 shows a sequence of real road images, their clustering results, and the extracted and predicted boundary points of some tested objects on the road ahead, which illustrate the obstacle detection and avoidance processes when the ALV navigates

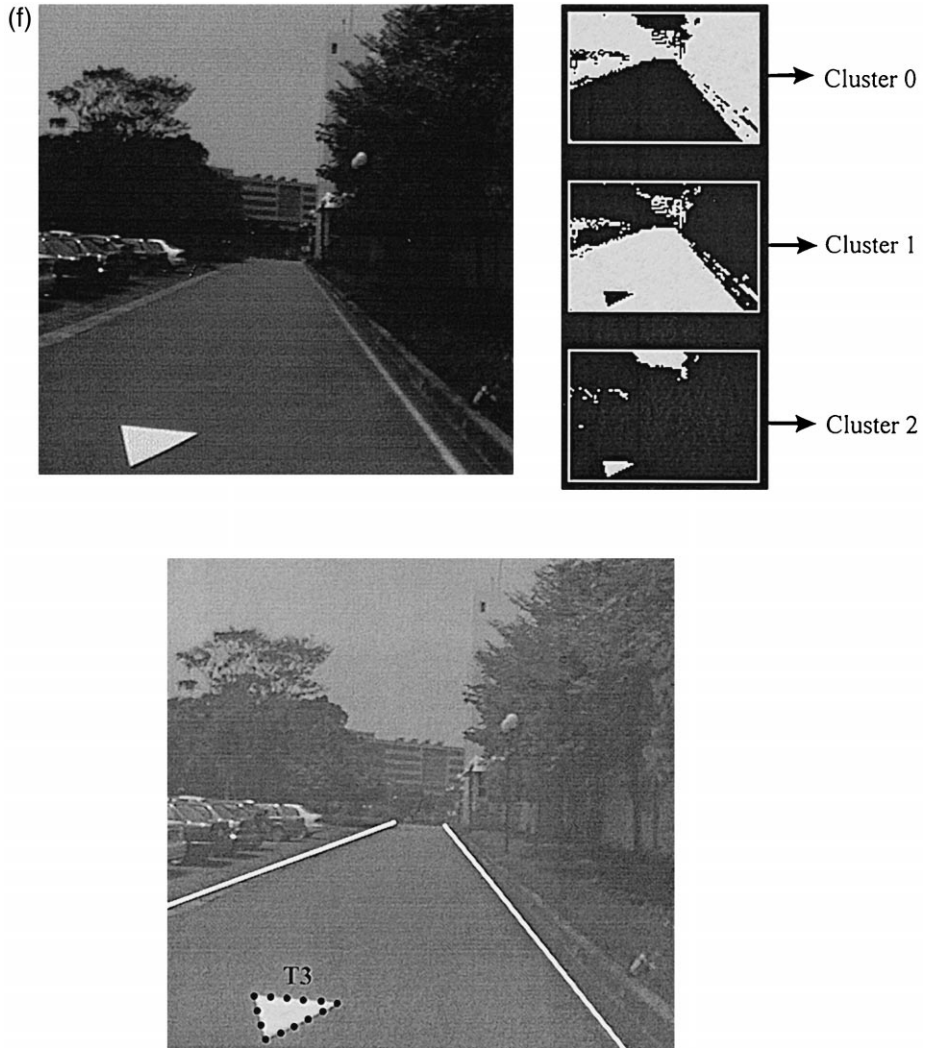

Fig. 17 (continued).

along a road. In the figure, the road boundaries are represented by the white lines, the extracted and predicted boundary points of objects are represented by the symbols of ‘‘ • ’’ and ‘‘**x**’’, respectively, the navigation points are represented by the white circles, and type-*i* objects, for $i = 1$, 2, or 3, are represented by T*i*’s.

In Fig. 17(a), two boards newly appear in the image, and the white board is classified into cluster-2 area and the black board is classified into cluster-0 area. As defined previously, they are type-1 objects in this cycle, and will be judged to be obstacles or not in the next cycle. In (b), the two boards become type-2 objects that will be judged to be obstacles or not in this cycle. After the shape matching process, it is decided that the white board is an obstacle and that the black board is not an obstacle. Then, the detected obstacle (the white board) is used to derive the navigation point, and a turn angle is computed to drive the ALV toward this point for safe navigation.

In (c), the two boards in (b) become type-3 objects that have been decided to be obstacles or non-obstacles in the previous cycle, and two additional white boards (type-1 objects) newly appear in

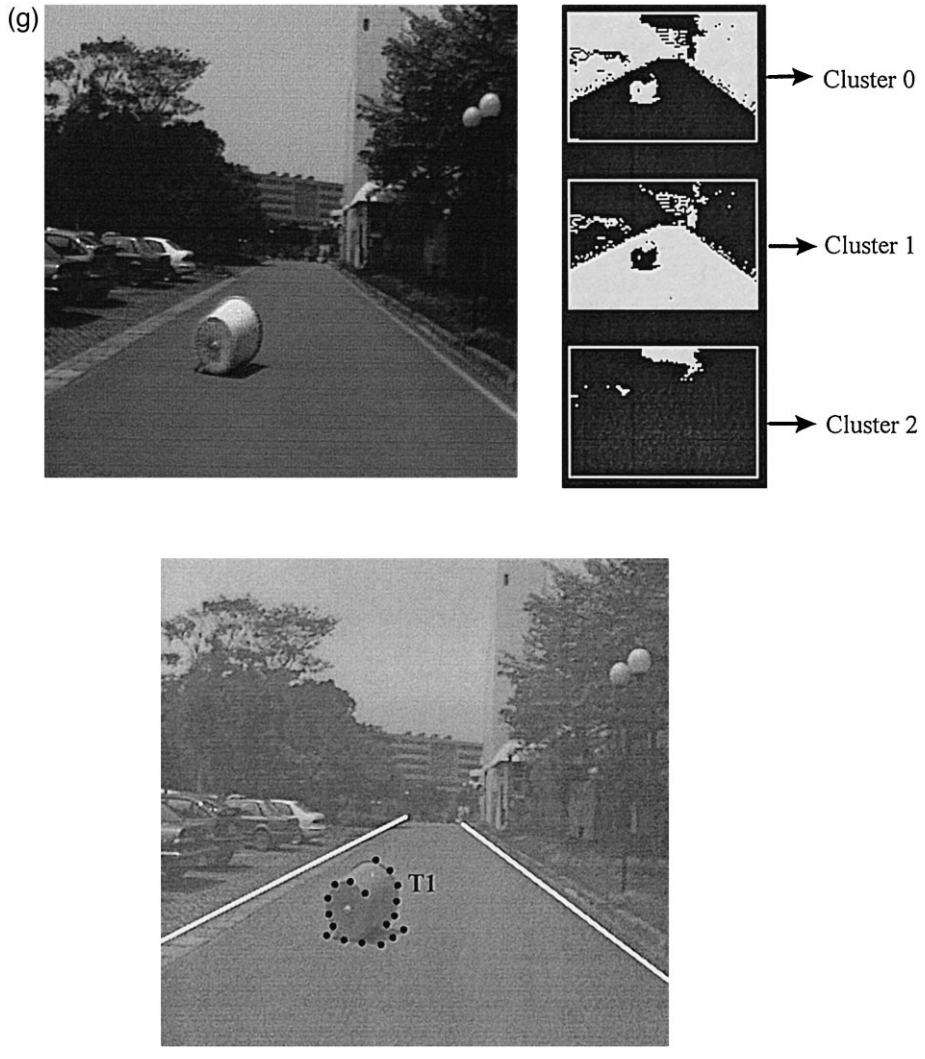

Fig. 17 (continued).

the image that will be judged to be obstacles or not in the next cycle. Since the previous obstacle in (b) still remains in the image in this cycle, it is extracted and used to derive the navigation point in this cycle, and the ALV keeps driving toward the navigation point. In (d), the two type-3 objects in (c) disappear from the image, and the two white boards in (d) become type-2 objects that are judged and decided to be non-obstacles in this cycle. Hence, no obstacle appears in the image in this cycle. But, due to the angle of the camera view as described in Section 3.1, the previous obstacle in (c) is still ahead of the ALV though it disappears from the image. At this moment, we predict the location of this hidden obstacle with respect to the ALV, which is then used to derive the navigation point that is also invisible in the image. The ALV keeps driving toward the navigation point.

In (e), the two white boards in (d) become type-3 objects and the ALV *has reached* the navigation point. At this moment, no obstacle is ahead of the ALV, and the ALV begins to head back to the central path line. In (f), one type-3 object in (e) disappear from the image and another type-3 object

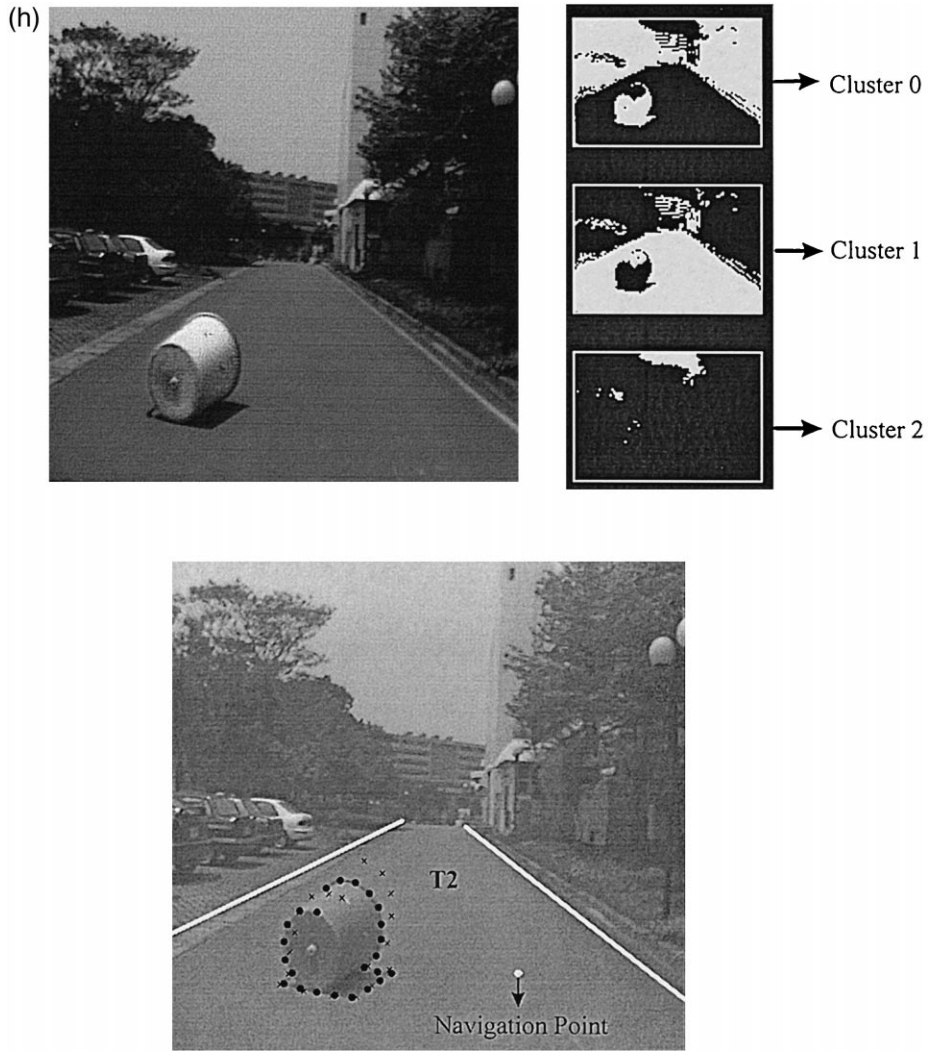

Fig. 17 (continued).

in (e) remains in the image, and the ALV keeps heading back to the central path line. Finally, as shown in (g), the ALV navigates on the central path line after navigation for several cycles. In (g), one plastic bucket newly appears on the road ahead, which will be judged to be obstacle or not in the next cycle. It can be seen from the cluster-1 area in the clustering result that the bucket partially blends into the road. In (h), the bucket is decided to be an obstacle after the shape matching process, where the bucket also blends into the road in this cycle. Then, a navigation point is derived, and a turn angle is computed to drive the ALV toward the navigation point. The obstacle avoidance process for ALV navigation described above is performed in the same way in subsequent navigation cycles.

5. Conclusions

A vision-based approach to obstacle detection and avoidance for ALV navigation in outdoor road environments has been proposed. Several techniques have been integrated in this study to provide a reliable

navigation scheme. Vision-based and control-based kinematic models have been combined such that fault-tolerant ALV navigation can be achieved. Backprojection and projection principles have been used to predict the boundary points of objects in the next cycle. The DWC correlation measure has been employed to judge whether an object is an obstacle. A connected component labeling algorithm and Warshall's algorithm have been implemented to extract effectively the boundary points of objects on the road ahead with less computing time. When obstacles disappear from the image but they are still ahead of the ALV, their positions with respect to the ALV can be predicted and used further to derive the navigation point. The safe navigation point on the road is chosen appropriately during navigation for obstacle avoidance. A sequence of real road images has been used in experiments to test the proposed obstacle detection and avoidance method along a road. Successful navigation results confirm the effectiveness of the proposed approach. Future research directions may focus on recognition and representation of general objects on roads, path planning, and environment sensing and learning, etc.

Acknowledgements

This work was supported by National Science Council, Republic of China under Grant NSC87-2213-E009-001.

References

[1] C. Thorpe, Outdoor visual navigation for autonomous robots, Robotics and Autonomous Systems 7 (1991) 85–98.
[2] S. Singh, P. Keller, Obstacle Detection for High Speed Autonomous Navigation, Proc. IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, April 1991, pp. 2798–2805.
[3] C. Thorpe, M. Hebert, T. Kanade, S. Shafer, Toward autonomous driving: the CMU navlab: Part I. Perception, IEEE Expert 6 (3) (1991) 31–42.
[4] C. Thorpe, M. Hebert, T. Kanade, S. Shafer, Toward autonomous driving: the CMU navlab: Part II. Architecture and systems, IEEE Expert 6 (3) (1991) 44–52.
[5] J.D. Crisman, C.E. Thorpe, SCARF: a color vision system that tracks roads and intersections, IEEE Transactions on Robotics and Automation 9 (1) (1993) 49–58.
[6] Y. Goto, A. Stentz, The CMU system for mobile robot navigation, Proc. IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 1987, pp. 99–105.
[7] C. Thorpe, M.H. Hebert, T. Kanade, S.A. Shafer, Vision and navigation for Carnegie-Mellon NAVLAB, IEEE Trans. on Pattern Analysis and Machine Intelligence 10 (3) (1988) 362–373.
[8] J.D. Crisman, C.E. Thorpe, UNSCARF, A Color Vision System for the Detection of Unstructured Roads, Proc. IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, April 1991, pp. 2496–2501.
[9] M.A. Turk, D.G. Morgenthaler, K.D. Germban, M. Marra, VITS — a vision system for autonomous land vehicle navigation, IEEE Trans. on Pattern Analysis and Machine Intelligence 10 (3) (1988) 342–361.
[10] A.M. Waxman, J.J. Lemoigne, L.S. Davis, B. Srinivasan, T.R. Kushner, E. Liang, T. Siddalingaiah, A visual navigation system for autonomous land vehicles, IEEE Journal of Robotics and Automation RA-3 (2) (1987) 124–141.
[11] K.E. Olin, D.Y. Tseng, Autonomous cross-country navigation, IEEE Expert 6 (3) (1991) 16–30.
[12] L.S. Davis, Visual navigation at the University of Maryland, Robotics and Autonomous Systems 7 (1991) 99–111.
[13] D. Kuan, G. Phipps, A. Hsueh, Autonomous robotic vehicle road following, IEEE Trans. on Pattern Analysis and Machine Intelligence 10 (4) (1988) 648–658.
[14] M. Schwarzinger, T. Zielke, D. Noll, M. Brauchmann, W.V. Seelen, Vision-Based Car-Following: Detection, Tracking, and Identification, Proc. of the Intelligent Vehicles '92 Symposium, Detroit, USA, Jun. 1992, pp. 24–29.
[15] F. Thomanek, E.D. Dickmanns, 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.
[16] M. Cappello, M. Campani, A. Succi, Detection of Lane Boundaries, Intersections and Obstacles, Proc. of the Intelligent Vehicles '94 Symposium, Paris, France, Oct. 1994, pp. 284–289.
[17] U. Regensburger, 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.
[18] B. Ulmer, VITA-An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic, Proc. of the Intelligent Vehicles '92 Symposium, Detroit, USA, Jun. 1992, pp. 36–41.
[19] W. Efenberger, Q.H. Ta, L. Tsinas, V. Graefe, Automatic Recognition of Vehicles Approaching from Behind, Proc. of the Intelligent Vehicles '92 Symposium, Detroit, USA, Jun. 1992, pp. 57–62.
[20] S. Hirata, Y. Shirai, M. Asada, Scene Interpretation Using 3-D Information Extracted from Monocular Color Images, Proc. of the 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, USA, Jul. 1992, pp. 1603–1610.
[21] M. Schmid, An Approach to Model-Based 3-D Recognition of Vehicles in Real Time by Machine Vision, Proc. of the 1994 IEEE/RSJ/GI International Conference on Intelligent

Robots and Systems, Munich, Germany, Sep. 1994, pp. 2064–2071.
[22] B. Roberts, B. Bhanu, Inertial navigation sensor integrated motion analysis for autonomous vehicle navigation, Journal of Robotic Systems 9 (6) (1992) 817–842.
[23] B. Heisele, W. Ritter, Obstacle Detection Based on Color Blob Flow, Proc. of the Intelligent Vehicles '95 Symposium, Detroit, MI, USA, Sep. 1995, pp. 282–286.
[24] W. Enkelmann, Obstacle detection by evaluation of optical flow fields from image sequences, Image and Vision Computing 9 (3) (1991) 160–167.
[25] S.M. Smith, J.M. Brady, A scene segmenter; visual tracking of moving vehicles, Engineering Applications of Artificial Intelligence 7 (2) (1994) 191–204.
[26] Z. Zhang, R. Weiss, A.R. Hanson, Obstacle detection based on qualitative and quantitative 3D reconstruction, IEEE Trans. on Pattern Analysis and Machine Intelligence 19 (1) (1997) 15–26.
[27] E. Grosso, M. Tistarelli, Active/dynamic stereo vision, IEEE Trans. on Pattern Analysis and Machine Intelligence 17 (9) (1995) 868–879.
[28] J.L. Bruyelle, J.G. Postaire, Direct range measurement by linear stereovision for real-time obstacle detection in road traffic, Robotics and Autonomous Systems 11 (1993) 261–268.
[29] N. Kehtarnavaz, N.C. Griswold, J.S. Lee, Visual control of an autonomous vehicle (BART) — the vehicle-following problem, IEEE Trans. on Vehicular Technology 40 (3) (1991) 654–662.
[30] Y.F. Wan, F. Cabestaing, J.C. Burie, A New Edge Detector for Obstacle Detection with a Linear Stereo Vision System, Proc. of the Intelligent Vehicles '95 Symposium, Detroit, MI, USA, Sep. 1995, pp. 130–135.
[31] M.E. Brauckmann, C. Goerick, J. Grob, T. Zielke, Towards All Around Automatic Visual Obstacle Sensing for Cars, Proc. of the Intelligent Vehicles '94 Symposium, Paris, France, Oct. 1994, pp. 79–84.
[32] J.L. Bruyelle, J.G. Postaire, Disparity Analysis for Real Time Obstacle Detection, Proc. of the Intelligent Vehicles '92 Symposium, Detroit, USA, Jun. 1992, pp. 51–56.
[33] M. Xie, L. Trassoudaine, J. Alizon, J. Gallice, Road obstacle detection and tracking by an active and intelligent sensing strategy, Machine Vision and Applications 7 (1994) 165–177.
[34] T.J. Fan, W.H. Tsai, Automatic chinese seal identification, Computer Vision, Graphics, Image Processing 25 (1984) 311–330.
[35] K.H. Chen, W.H. Tsai, Vision-based autonomous land vehicle guidance in outdoor road environments using combined line and road following techniques, Journal of Robotic Systems 14 (10) (1997) 711–728.
[36] S.D. Cheng, W.H. Tsai, Model-based guidance of autonomous land vehicle in indoor environments by structured light using vertical line information, Journal of Electrical Engineering 34 (6) (1991) 441–452.
[37] L.L. Wang, P.Y. Ku, W.H. Tsai, Model-based guidance by the longest common subsequence algorithm for indoor autonomous vehicle navigation using computer vision, Automation in Construction 2 (1993) 123–137.
[38] Y.M. Su, 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 17 (1) (1994) 63–73.
[39] R.C. Gonzalez, Richard E. Wood, Digit Image Processing, Addison-Wesley Publishing, Reading, MA, USA, 1992.
[40] S. Warshall, A theorem on boolean matrices, Journal of the ACM 9 (1) (1962) 11–12.