The 13th IPPR Conference on
Computer Vision, Graphics and Image Processing

Obstacle Avoidance in Person Following for Vision-based Autonomous Land Vehicle Guidance Using Vehicle Location Estimation and Quadratic Pattern Classifier

Ching-Heng Ku (顧靜恆) and Wen-Hsiang Tsai(蔡文祥)
E-mail: a00chk00@nchc.gov.tw and whtsai@cis.nctu.edu.tw
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 300, Republic of China
Tel: (886)3-5720631 Fax: (886)3-5721490

ABSTRACT

An obstacle avoidance method for use in person following for vision-based autonomous land vehicle (ALV) guidance is proposed. This method is based on the use of vehicle location estimation and a quadratic pattern classifier, and aims to guide the ALV to follow a walking person in front by navigating along a derived collision-free path. Before generating the collision-free path, the person's location is obtained from extracted objects in the image by a person detection method. The object closest to a predicted person location is regarded as the followed person and the remaining objects are regarded as obstacles. The collision-free navigation path is designed for ALV guidance in such a way that the ALV not only can keep following the person but also can avoid collision with nearby obstacles. The navigation path results from a quadratic classifier that uses the vehicle and all of the objects in the image as input patterns. A turn angle is then computed to drive the ALV to follow the navigation path. Successful navigation sessions confirm the feasibility of the approach.

Keyword: obstacle avoidance, autonomous land vehicle guidance, person following, vehicle location estimation, quadratic pattern classifier

1. Introduction

In recent years, many approaches to autonomous land vehicle (ALV) guidance in indoor and outdoor environments have been developed. How to guide the ALV to navigate by following a walking person in a certain environment and avoid obstacles in the mean time is the major goal of this study.

In the study of obstacle avoidance, some vision-based navigation methods [1][2] for mobile robots with obstacle avoidance capability have been proposed. Ohya [1] used a model edge map for vehicle navigation on a planned path. Obstacles are detected by computing the difference between the edges estimated from the 3D environment model and the edges detected from the actual camera image. The navigation system developed by Lorigo, et al. [2] consists of three independent vision modules, an edge module, an RGB module, and an HSV module, for obstacle detection. The obstacle boundaries from the individual modules are combined into a single obstacle boundary which is converted to motor commands. Yang [3] used an adaptive-network-based fuzzy classifier to define 3D obstacle regions that must be avoided. Biewald [4] used a human-like concept and a more qualitative world model to plan routes. Ku and Tsai [5] used a quadratic classifier in pattern recognition for collision avoidance in ALV navigation in an unknown indoor environment. When a vision-based ALV navigates by following a person walking in front, the person has to be detected first from the image captured by a camera. Although the image of the person is also detected in some pedestrian tracking systems, the approaches [6-8] to detecting a pedestrian using the difference between two consecutive images cannot be used in a person following system. The image for pedestrian tracking is acquired using a stationary CCD camera. The difference between two consecutive images contains the information of the moving person. However, the camera used in person following is mounted on the ALV and moved along the path of the ALV. The moving person cannot be detected using the image difference information when the camera is not stationary.

On the other hand, after the relative position of the person to the ALV is calculated, a trajectory for the ALV need be generated to obtain the turn angle of the front wheels of the ALV. Some trajectory planning approaches [9-11] have been proposed. Munoz and Ollero [9] combined a kinematic visibility graph planning method, a path generation algorithm based on beta-spline curves, and a cubic spline speed profile definition technique to propose a smooth trajectory planning method for mobile robots. Shiller and Serate [10] proposed a trajectory planning method for computing the track forces and track speeds of planar tracked vehicles required to follow a given path at specified speeds on horizontal and inclined planes. Ku and Tsai [11] used sequential pattern recognition techniques for ALV smooth navigation by person following. These methods do not consider the existence of obstacles.

In this study, an obstacle avoidance method for use in person following for vision-based ALV guidance using vehicle location estimation and quadratic pattern classifier design is proposed. This method aims to guide the ALV to follow a walking person in front along a corridor with obstacles. This goal is achieved in this study by guiding the ALV to navigate along a derived collision-free path. First, the translation of the ALV location at the current sampling instant relative to that at the previous sampling instant is estimated using the ALV control

information. The translation is then used together with previous person locations with respect to the vehicle location at the previous sampling instant to predict the person location at the current sampling instant. After all of the objects appearing in the image are extracted, the object closest to the predicted person location is regarded as the followed person and the remaining objects are regarded as obstacles. A collision-free navigation path is then derived for ALV guidance in such a way that the ALV not only can keep following the person but also can avoid collision with nearby obstacles. The navigation path is derived by the use of the quadratic classifier that regards the vehicle and all of the objects in the image as input patterns. All patterns are categorized into two groups as the input to the quadratic classifier using a pattern generation method proposed in this study. A turn angle is then computed to drive the ALV to follow the navigation path. The main ideas of this study include the categorization of the locations of the person and obstacles without special marks, and the use of the quadratic classifier to generate a collision-free path for safe person following in the navigation.

A flowchart of the proposed obstacle avoidance method for person following is shown in Fig. 1. In this study, we focus on the steps of the detection of the locations of the person and obstacles, and the use of the quadratic classifier to generate a collision-free path. Some other details of the processes to implement the proposed method are presented in [5]. The details of the system flowchart are described as follows.

Step 1. *Image acquisition*: Capture the image of the front view of the vehicle with a wide-angle camera mounted on the vehicle.

Step 2. *Vehicle location estimation*: Estimate the translation of the ALV location at the current sampling instant relative to that at the previous sampling instant using the ALV control information. The estimation method is described in detail in Section 2.A.

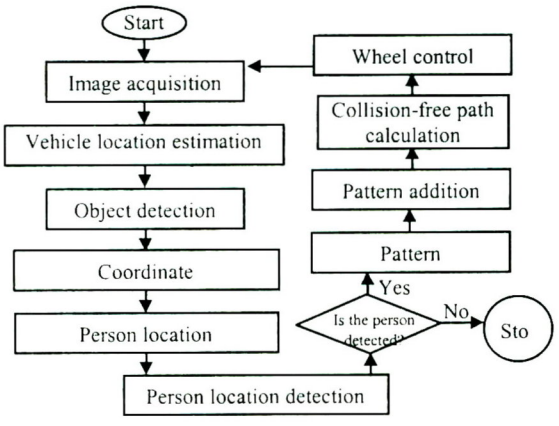


Fig. 1. System flowchart.

Step 3. *Object detection*: Detect the image points that compose the baselines of the objects, including those of the walls of the corridor, the followed person, and the obstacles that appear in the way of ALV navigation in the corridor. This step is conducted by the use of an obstacle detection algorithm introduced in [5]. The obstacle detection algorithm uses scan lines and image processing techniques such as local thresholding and region growing methods to detect obstacles. An originality of the algorithm is the concept that obstacles are assumed to lie on the ground and represented by baselines.

Step 4. *Person location prediction*: Predict the person location at the current sampling instant using the translation of the ALV together with the person location at the previous sampling instant. The prediction method is described in Section 2.B.

Step 5. *Person location detection*: Take the object closest to the predicted person location as the followed person and regard the remaining objects as obstacles. The person detection method is described in Section 2.C. Besides, check whether the followed person is detected or not. If yes, continue; otherwise, stop the vehicle.

Step 6. *Pattern generation*: Categorize the patterns representing the obstacles, the person, and the two sides of the vehicle body into two classes using a pattern generation algorithm introduced in Section 3.B. Add some extra points as patterns using a pattern addition algorithm described in [5] to take into consideration the width of the vehicle.

Step 7. *Collision-free path calculation*: Generate a collision-free path using a quadratic classifier described in Section 3.A. The path is just the decision boundary of the classifier designed with the input patterns generated in Step 6.

Step 8. *Wheel control*: Steer the ALV front wheels according to the turning angle derived by a method described in [5] based on the collision-free path generated in Step 7. In this way, the vehicle keeps its trajectory on a collision-free path continually.

Step 9. Go to Step 1.

In the process described above, the proposed system does not use any environment knowledge given in advance; instead the guidance of the ALV is based on local visual information only. At least five advantages are found in this approach. First, the locations of the person and obstacles can be discriminated. Second, the ALV can follow the walking person based on the collision-free path. Third, the derived quadratic path is more precise to match the kinematic trajectory of the vehicle than the linear path derived by most other methods. Fourth, by following the quadratic path it is easier to go through obstacles without collisions than other approaches using linear paths [5]. In this study, the generated quadratic collision-free path goes through the center of the ALV. This means that the ALV is located on the collision-free path in every cycle and need not navigate to approximate a collision-free path that is usually a linear one. Fifth and the last, both the current locations of the vehicle and the person are taken into consideration while generating the quadratic collision-free path.

In the remainder of this paper, the approach to obtaining the person position by the proposed person detection method is described in Section 2. The approach includes three steps, namely, the estimation of the vehicle location between two

sampling instants, the prediction of the person position, and the detection of the person position. In Section 3, the method for deriving the collision-free path is described. The method includes a pattern generation method and a path generation method. In Section 4, some experimental results are illustrated. Conclusions are included in the last section.

2. Proposed Detection Method of Person Location

A method for detecting the person location is proposed in this section. In this study, all obstacles are all assumed to lie on the ground, so the surfaces of obstacles will contact the ground at certain spots, which appear as line segments in most cases and will be called baselines in this study. When all the baselines of the objects, including the obstacles and the followed person, in a corridor are detected in an image, the ALV does not know which object is the person to follow. The proposed person detection method aims to discriminate obstacles from the followed person.

Because the moving direction and the speed of the person is unpredictable, the person location with respect to the VCS at the current sampling instant cannot be found only by vision-based information unless the cycle time is zero. But the person location at the current sampling instant can be estimated using the control-based information and the person location at the previous sampling instant. After all of the objects appearing in the image are extracted, the object closest to the estimated person location by the control-based ALV information is regarded as the followed person and the remaining objects are regarded as obstacles.

More specifically, as shown in Fig. 2, let P_i denote the person position and VCS_i denote the vehicle coordinate system at the beginning of cycle i during navigation. Assume that, at the beginning of cycle i, the x and y coordinates of the person position at P_{i-1} with respect to VCS_{i-1} have been found. We use the ALV control information to estimate the translation from VCS_{i-1} to VCS_i, which is then used to compute the x and y coordinates of the person position at P_{i-1} with respect to VCS_i. Also assume that, at the beginning of cycle i+1, the x and y coordinates of the person position at P_i with respect to VCS_i have been found. Again, the ALV control information is used to estimate the translation from VCS_i to VCS_{i+1}, which is then used to compute the x and y coordinates of the person positions at P_{i-1} and P_i with respect to VCS_{i+1}.

After P_{i-1} and P_i are found in cycle i+1, we then detect P_{i+1} as follows. We first predict the person position, called P'_{i+1}, at the beginning of cycle i+1 using P_{i-1} and P_i. The prediction process is illustrated in Fig. 2, where it is assumed that the translation from P_{i-1} to P_i is identical to that from P_i to P'_{i+1}. The predicted person position is reasonable because in the short duration between two cycles the person may be assumed to move straightforward with a constant speed in general. Since the x and y coordinates of the person positions at P_{i-1} and P_i with respect to VCS_{i+1} have been computed previously, the x and y coordinates of the person position at P'_{i+1} with respect to VCS_{i+1} can be solved accordingly. Then, the predicted person position P'_{i+1} is used to detect the real person position P_{i+1}. The way is to find the object closest to P'_{i+1} and take it as the detected person located at P_{i+1}.

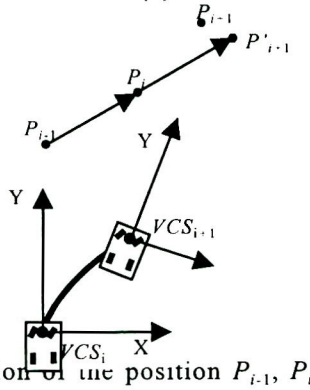


Fig. 2. Illustration of the position P_{i-1}, P_i, P_{i+1}, and P'_{i+1}.

The details of the above process are described in the following. We first state the estimation process of the translation from VCS_i to VCS_{i+1} in Section 2.A. Then, in Section 2.B, the derivation process of the coordinates of the person position at P_{i-1} and P_i with respect to VCS_{i+1} is introduced, followed by a description of the prediction process of the person position at P'_{i+1}. Finally, the detection process of the person position at P_{i+1} is introduced in Section 2.C.

A. Estimation of Translation from VCS_i to VCS_{i+1}

The translation from VCS_i to VCS_{i+1} is estimated using the ALV control information, namely, the speed and the turn angle of the ALV, and the time interval between two sampling instants. These values can be obtained from the feedback information of the ALV and then used to estimate the kinematic trajectory of the ALV from VCS_i to VCS_{i+1}. Accordingly, the translation and the rotation of the two VCS's can be obtained. The details are as follows.

Consider the simple kinematic model for an automobile with front and rear tires [12], as shown in Fig. 3. The rear tires are aligned with the car body sides while the front tires are allowed to spin about the vertical axes. Besides, the center position between the two front wheels is treated as the origin of the VCS. As shown in Fig. 3, the vehicle moves a distance S from location A to location B by turning an angle of ϕ, where A and B represent the origin of the VCS_i and that of the VCS_{i+1}, respectively. We assume that the vehicle speed v and the navigation time interval t of the vehicle are both known in advance. Hence, the navigation distance S is a constant and can be computed by $S = vt$.

The translation specified by the values x and y in the VCS_i is acquired by the following equations [5]:

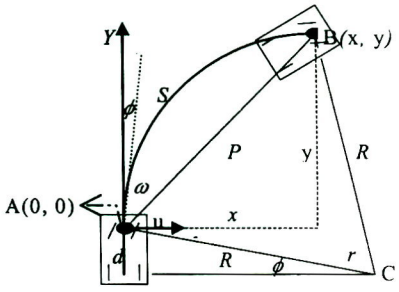


Fig. 3. Analysis of the path S, through which the vehicle navigates from location A to B by turning an angle of ϕ.

$$x = P\cos u,\ y = P\sin u,$$
$$P = R\sqrt{2(1-\cos r)},\ R = \frac{d}{\sin\phi},\ r = \frac{S}{R},$$
$$u = \frac{\pi}{2} - \phi - \frac{r}{2}, \tag{1}$$

where R is the rotation radius, d is the distance between the front wheels and the rear wheels, r and P are the corresponding angle and the secant line of S, respectively, and ϕ is the turning angle of the front wheels. According to Equations (1), the translation (x, y) from VCS_i to VCS_{i+1} is computed to be as follows:

$$x = \frac{d}{\sin\phi}\sqrt{2\left(1-\cos\frac{vt\sin\phi}{d}\right)}\cos\left(\frac{\pi}{2}-\phi-\frac{vt\sin\phi}{2d}\right), \tag{2}$$

$$y = \frac{d}{\sin\phi}\sqrt{2\left(1-\cos\frac{vt\sin\phi}{d}\right)}\sin\left(\frac{\pi}{2}-\phi-\frac{vt\sin\phi}{2d}\right). \tag{3}$$

Next, we state the derivation process of the rotation angle, denoted as φ, from VCS_i to VCS_{i+1}. Since the direction vector of the vehicle head in the VCS is that of the Y-axis of the VCS, the angle φ can be obtained by calculating the angle between the direction vector of the vehicle head at the position A and that at the position B, as shown in Fig. 3. Thus, the direction vector v_3 of the vehicle head at the position A with respect to VCS_i can be written as

$$v_3 = (0, 1). \tag{4}$$

In addition, as shown in Fig. 4, the direction vector v_4 of the vehicle head at the position B with respect to VCS_i can be derived from the direction vector v_2 of the front wheels of the vehicle at the position B in the following way. First, since the vector v_2 is the tangent of the trajectory of the ALV, it can be obtained to be as follows [14]:

$$v_2 = (d + y,\ d\cot\phi - x). \tag{5}$$

Then, because the angle between vectors v_2 and v_4 is the turn angle ϕ of the vehicle, the vector v_4 can be calculated to be as follows:

$$\vec{v_4} = (x_4, y_4) = \vec{v_2}\cdot\begin{bmatrix}\cos\phi & \sin\phi\\ -\sin\phi & \cos\phi\end{bmatrix} \tag{6}$$
$$= (y\cos\phi + x\sin\phi,\ d\sin\phi + y\sin\phi - x\cos\phi + d\cot\phi\cos\phi).$$

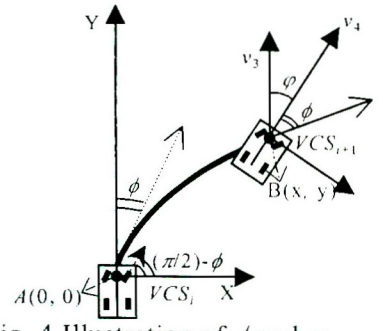


Fig. 4 Illustration of ϕ and φ.

Hence, the angle φ between the two direction vectors v_3 and v_4 can be calculated according to the following equation:

$$v_3 \cdot v_4 = |v_3|\cdot|v_4|\cdot\cos\varphi. \tag{7}$$

By substituting (4) and (6) into (7), the angle φ can be derived to be as follows:

$$\varphi = \operatorname{acos}\left(\frac{v_3\cdot v_4}{|v_3|\cdot|v_4|}\right) = \operatorname{acos}\left(\frac{y_4}{\sqrt{x_4^2+y_4^2}}\right) \tag{8}$$
$$= \operatorname{acos}\left(\frac{d\sin\phi + y\sin\phi - x\cos\phi + d\cot\phi\cos\phi}{\sqrt{(y\cos\phi + x\sin\phi)^2 + (d\sin\phi + y\sin\phi - x\cos\phi + d\cot\phi\cos\phi)^2}}\right).$$

B. Prediction of Person Position at P'_{i+1}

In this section, we describe the derivation process of the coordinates of the person position at P_{i-1} and P_i with respect to VCS_{i+1}, followed by a description of the prediction process of the person position at P'_{i+1}. As shown in Fig. 5, let x_{i-1}^i and y_{i-1}^i denote the coordinates of the person position at P_{i-1} with respect to VCS_i. Similarly, let x_i^i and y_i^i represent the coordinates of the person position at P_i with respect to VCS_i. Assume that the coordinates $(x_{i-1}^i,\ y_{i-1}^i)$ and $(x_i^i,\ y_i^i)$ with respect to VCS_i have been obtained. The translation (x, y) and the rotation angle φ from VCS_i to VCS_{i+1}, obtained in the previous section, can be used to calculate the coordinates $(x_{i-1}^{i+1},\ y_{i-1}^{i+1})$ and $(x_i^{i+1},\ y_i^{i+1})$ of the person position at P_{i-1} and P_i with respect to VCS_{i+1}, respectively, by the following equations:

$$[x_{i-1}^{i+1}\ \ y_{i-1}^{i+1}\ \ 1] = [x_{i-1}^{i}\ \ y_{i-1}^{i}\ \ 1]\cdot\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -x & -y & 1\end{bmatrix}\cdot\begin{bmatrix}\cos\varphi & \sin\varphi & 0\\ -\sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{bmatrix}, \tag{9}$$

$$[x_i^{i+1}\ \ y_i^{i+1}\ \ 1] = [x_i^{i}\ \ y_i^{i}\ \ 1]\cdot\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -x & -y & 1\end{bmatrix}\cdot\begin{bmatrix}\cos\varphi & \sin\varphi & 0\\ -\sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{bmatrix}. \tag{10}$$

In (9), the center of VCS_i is translated to that of VCS_{i+1} so that the coordinates of P_{i-1} with respect to VCS_{i+1} have to decrease the translation (x, y). In addition, the direction of VCS_i is rotated to be the same as that of VCS_{i+1} according to the angle φ. Hence, the coordinates $(x_{i-1}^{i+1},\ y_{i-1}^{i+1})$ of P_{i-1} with respect to VCS_{i+1} can be obtained in (9). In the mean time, the coordinates $(x_i^{i+1},\ y_i^{i+1})$ of P_i with respect to VCS_{i+1} is obtained in (10).

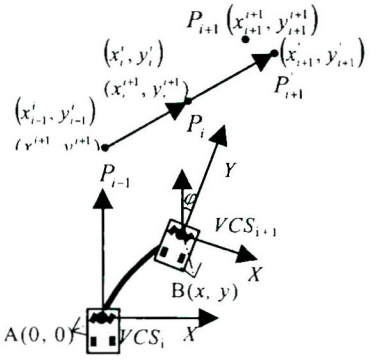


Fig. 5 Illustration of the coordinates of P_{i-1}, P_i and P'_{i+1} with respect to VCS_i and VCS_{i+1}.

Since we assume that the translation from P_{i-1} to P_i is identical to that from P_i to P'_{i+1}, the coordinates (x'_{i+1}, y'_{i+1}) of the predicted person position at P'_{i+1} with respect to VCS_{i+1} can be calculated to be as follows:

$$x'_{i+1} = 2x_i^{i+1} - x_{i-1}^{i+1}, \tag{11}$$

$$y'_{i+1} = 2y_i^{i+1} - y_{i-1}^{i+1}. \tag{12}$$

C. Detection of Person Position at P_{i+1}

When the predicted person position P'_{i+1} is obtained, the person position P_{i+1} can be detected, as mentioned previously, to be the object closest to P'_{i+1}. However, each candidate object is necessarily just a point in shape; it is a group of points composing a baseline segment. Therefore, a more sophisticate method is adopted in this study, based on the use of the mean and the variance of the detected baselines. The mean D of the baseline of an object represents the position of the object and its value is obtained by the following equation:

$$D = \frac{1}{m}\sum_{i=1}^{m} M_i = \frac{1}{m}\sum_{i=1}^{m}[x_i \quad y_i]^T = [x_D \quad y_D]^T, \tag{13}$$

where m is the number of the points that compose a baseline of an object and $M = [x \quad y]^T$ represents the coordinates of those points from the top view of the VCS. The calculated mean of the object is used to compute the Euclidean distance to the position P'_{i+1}. Since the coordinates x'_{i+1} and y'_{i+1} of the position P'_{i+1} have been obtained, the Euclidean distance H between the mean D and the position P'_{i+1} can be obtained by the following equation:

$$H = \sqrt{(x_D - x'_{i+1})^2 + (y_D - y'_{i+1})^2}. \tag{14}$$

The obtained distance H is taken as one of features to detect the person position P_{i+1}. As shown in Fig. 6, if an object is the person that we want to detect, its distance H will be smaller than a given threshold value T_d. On the other hand, the variance K of the points that compose a baseline of an object is taken as another feature and its value can be obtained by the following equation:

$$K = \frac{1}{m}\sum_{i=1}^{m}\sqrt{(M_i - D)^T (M_i - D)}. \tag{15}$$

Since the length of the baseline of a person is smaller than that of a wall and an obstacle in general, the variance K of the person will be smaller than a given threshold value T. Hence, the proposed method for detecting the position P_{i+1} includes two steps and is described in the following. In the first step, we check whether the variance K and the distance H of an object are smaller than the thresholds value T and T_d, respectively. The conditions may be written as follows:

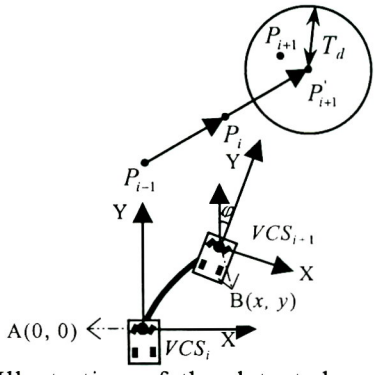


Fig. 6. Illustration of the detected person position P_{i+1} and the distance threshold value T_d.

$$K \le T \text{ and} \tag{16}$$

$$H \le T_d. \tag{17}$$

When all detected objects are tested using the two conditions (16) and (17), three situations might happen. The first situation is that no object satisfies both of the two conditions. It means that the ALV misses the followed person. In this case, the ALV is stopped. The second situation is that just one object satisfies both of the two conditions. The object is then taken as the followed person at P_{i+1}. The last situation is that more than one object satisfies both of the two conditions. Then, the object with the smallest distance H is chosen to be the person to follow. That is, in the second step, when the distance H between P'_{i+1} and an object P is the smallest, the object P is regarded as the followed person at P_{i+1} and the remaining objects are regarded as obstacles. If the object with the smallest distance is an obstacle instead of the followed person, the followed person can not be detected around the area that is predicted using the proposed method in Section 2-B. Hence, the ALV will stop in the next cycle and restart to detect an moving object as the followed person.

3. Proposed Collision-free Path Generation Method for Obstacle Avoidance in Person Following

In this section, the guidance method for obstacle avoidance in person following is proposed. The goal of this method is to generate a collision-free path that passes through the location of the followed person and the middle point between the two front wheels of the vehicle without hitting the obstacles. This goal is achieved using the proposed pattern generation method and the path generation method described in the following.

In the proposed path generation method, the decision boundary of the quadratic classifier is treated as the generated collision-free path. When the quadratic classifier is applied, there need two classes of patterns for use as the input. On the other hand, the proposed pattern generation method provides a systematic way to categorize all patterns into two groups L and R. These patterns include the baseline of obstacles, two sides of the vehicle body, and two series of artificial patterns generated on the left-hand and the right-hand sides of the person location. The pattern generation method is proposed in Section 3.A.In Section 3.B, the formula of the

quadratic classifier is described.

A. Pattern Generation for Quadratic Classifier Design

In this section, a pattern generation method is proposed to categorize all patterns into two groups L and R, which include two kinds of obstacle patterns O^L and O^R, two kinds of vehicle patterns V^L and V^R, and two kinds of person patterns P^L and P^R in the VCS. The details of the processing step are described in the following.

In this study, all detected objects except the followed person are treated as obstacles. The baseline of an obstacle is categorized into one of two sets O^L and O^R according to a derived reference line L1 in the VCS. As shown in Fig. 7, the reference line L1 passes through the person's position P and the origin of the VCS. With the coordinates x_{i+1}^{i+1} and y_{i+1}^{i+1} of the person's position obtained in Section 2.C, the formula of the reference line L1 can be derived to be as follows:

$$\mathrm{L1} = y - \frac{y_{i+1}^{i+1}}{x_{i+1}^{i+1}} x = 0. \tag{18}$$

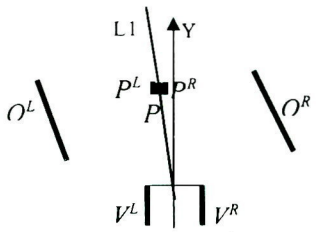


Fig. 7. Two kinds of obstacles O^L and O^R, two kinds of vehicle patterns V^L and V^R, and two kinds of person patterns P^L and P^R in the VCS.

If a point (x, y) is located on the left-hand side of the line L1, the value of L1 with respect to (x, y) is smaller than zero. Hence, if the mean of the baseline of an obstacle is on the left-hand side of the reference line L1, i.e., if L1 < 0, the baseline of the obstacle is assigned to a component of the set O^L. Here, the mean position of the baseline points of an obstacle represents the position of the obstacle. On the contrary, if the mean position of the baseline points of the obstacle is on the right-hand side of L1, i.e., if L1 > 0, the baseline of the obstacle is assigned to a component of the set O^R. That is, the baseline B_i of an obstacle can be categorized by the following rule:

$$\text{assign } B_i \text{ to } \begin{cases} O^L, & \text{if } \mathrm{L1}(D(B_i)) < 0; \\ O^R, & \text{if } \mathrm{L1}(D(B_i)) > 0. \end{cases} \tag{19}$$

where $D(B_i)$ represents the mean of the baseline B_i of the obstacle.

In the following, we state the proposed method for generating two sets P^L and P^R of patterns to represent the left-hand side and the right-hand side of the person position, respectively. The scheme makes the decision boundary pass through the person's position so that the collision-free path can be used as a navigation path in person following. As shown in Fig. 7, the person's position P is taken to be the mean of the baseline of the person. Each of the two generated pattern sets P^L and P^R are designed to be composed of a series of points that are located on the left-hand or the right-hand sides, respectively, of the position P. The distance between the pattern set P^L and the position P, and the pattern set P^R and the position P are both 30cm. The length of P^L or P^R is also 30cm.

Besides the pattern sets O^L, O^R, P^L and P^R, as shown in Fig. 7, we regard the points of the projection of the left-hand and the right-hand sides of the vehicle body in the x-y plane of the VCS as two sets of patterns V^L and V^R, respectively. This scheme makes the collision-free path to go through the origin of the VCS, i.e., to go through the middle point between the two front wheels of the vehicle.

Now, we describe how to associate separately each of the pattern sets V^L and V^R with the person pattern sets P^L and P^R, and the obstacle pattern sets O^L and O^R for finding the collision-free path. The scheme for categorizing the six kinds of patterns O^L, O^R, P^L, P^R, V^L, and V^R into two groups L and R is based on the following rule:

$$L = O^L \cup V^L \cup P^L, \tag{20}$$

and

$$R = O^R \cup V^R \cup P^R. \tag{21}$$

More specifically, the left vehicle pattern set V^L is associated with the left person pattern set P^L and the obstacle pattern set O^L. On the other hand, the right vehicle pattern set V^R is associated with the right person pattern set P^R and the obstacle pattern set O^R. In this way, the two groups L and R are obtained, which are then used as the input to the classification designed by the way described in the next section.

B. Quadratic Classifier for Path Generation

From the result of the pattern generation processes to be described in Sections 3.A, we can obtain two pattern groups L and R, each of which includes patterns representing obstacles, corridor walls on one side of the vehicle, the two sides of the vehicle body, and that of the person. In this study, the quadratic classifier is used to generate a quadratic decision boundary $h(X)$ between L and R. The decision boundary is taken as a collision-free path that the ALV follows to achieve safe navigation in person following. Since the generated path is limited to go through the middle point between the two front wheels of the vehicle (i.e., through the origin of the VCS) in this study, as shown in Fig. 8, the decision boundary $h(X)$ will approach the person's location without hitting the obstacles. The vehicle can thus safely follow the person by navigation along the derived collision-free path $h(X)$. The details for generating the decision boundary are described as follows.

As shown in Fig. 8, let O^L and O^R be two groups of patterns representing obstacles, P^L and P^R be those representing the followed person, and V^L and V^R be those representing the vehicle body sides. In this study, O^L, P^L and V^L belong to the group L, and O^R, P^R and V^R belong to the group R. Each pattern in the pattern group consists of x and y values in the

VCS. We denote the coordinates of the ith pattern in L as $[x_i^L \quad y_i^L]^T$ and those of the jth pattern in R as $[x_j^R \quad y_j^R]^T$.

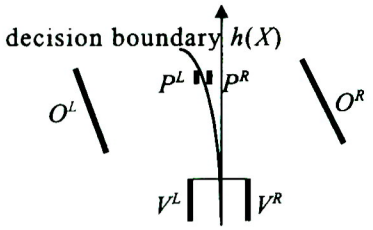


Fig. 8. A two-dimensional decision boundary, $h(X)$, passing through the origin of the VCS (i.e., through the middle point of the two vehicle wheels) and the person's location, is found by the quadratic classifier.

According to the theory of pattern recognition [13], we can find a quadratic decision boundary between the patterns of two classes to form a quadratic classifier. A general representation of the quadratic classifier is written as follows:

$$h(X)=X^TQX+V^TX+v_0, \qquad Q=\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}, \quad V=\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{22}$$

where v_0 is a constant, and the vector $X=[x_1 \quad x_2]^T$ specifies a pattern of L or R. If $h(X) < 0$, it means that X belongs to L; if $h(X) > 0$, it means that X belongs to R.

Since the collision-free path is limited to go through the middle point between the two front wheels of the vehicle, i.e., the origin of the VCS, the value of v_0 in (22) can be set as zero. The formula of the decision boundary $h(X)$ in Equation (22) can be represented as follows [5]:

$$\begin{aligned} h(X) &= X^TQX+V^TX=\sum_{i=1}^{2}\sum_{i=1}^{2}q_{ij}x_ix_j+\sum_{i=1}^{2}v_ix_i \\ &=\sum_{i=1}^{3}\alpha_iy_i+\sum_{i=1}^{2}v_ix_i=A^TY+V^TX \\ &=[\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad v_1 \quad v_2][y_1 \quad y_2 \quad y_3 \quad x_1 \quad x_2]^T, \end{aligned} \tag{23}$$

where

$$A=[\alpha_1 \quad \alpha_2 \quad \alpha_3]^T=[q_{11} \quad q_{12}+q_{21} \quad q_{22}], \qquad X=[x_1 \quad x_2]^T,$$

and $Y=[y_1 \quad y_2 \quad y_3]^T=[x_1^2 \quad x_1x_2 \quad x_2^2]^T$.

We can use the technique for designing a linear classifier to find the coefficients (α_1, α_2, α_3, v_1, v_2) as follows.

Let $M=[Y^T \quad X^T]^T=[y_1 \quad y_2 \quad y_3 \quad x_1 \quad x_2]^T$. The values of matrix M come from those of the patterns of L and R. Matrix M for L and R will be denoted as M^L and M^R, respectively. Substitute the values of x_1 and x_2 in matrix M by those of the ith point of L, $x_1=x_i^L$ and $x_2=y_i^L$, and denote the resulting matrix M by M_i^L. Similarly, substitute the values of x_1 and x_2 in matrix M by those of the ith point of R, $x_1=x_i^R$ and $x_2=y_i^R$, and denote the resulting matrix M by M_i^R.

Accordingly, the nonlinear solutions of the coefficients for the quadratic classifier whose input design patterns come from the points of L and R are equal to the linear solutions of the coefficients for the linear classifier whose input design patterns come from the points of M^L and M^R. So, the five coefficients in Equation (23) can be solved [5] and the result is represented as follows:

$$[\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad v_1 \quad v_2]^T=\left[\frac{1}{2}K_L+\frac{1}{2}K_R\right]^{-1}(D_R-D_L). \tag{24}$$

where D_L and D_R, and K_L and K_R are the means and variances of M^L and M^R, respectively, and are computed as follows:

$$\begin{aligned} D_L &= \frac{1}{m}\sum_{i=1}^{m}M_i^L=\frac{1}{m}\sum_{i=1}^{m}\left[(x_i^L)^2 \quad x_i^Ly_i^L \quad (y_i^L)^2 \quad x_i^L \quad y_i^L\right]^T, \\ D_R &= \frac{1}{n}\sum_{i=1}^{n}M_i^R=\frac{1}{n}\sum_{i=1}^{n}\left[(x_i^R)^2 \quad x_i^Ry_i^R \quad (y_i^R)^2 \quad x_i^R \quad y_i^R\right]^T, \\ K_L &= \frac{1}{m}\sum_{i=1}^{m}(M_i^L-D_L)(M_i^L-D_L)^T, \; K_R=\frac{1}{n}\sum_{i=1}^{n}(M_i^R-D_R)(M_i^R-D_R)^T, \end{aligned} \tag{25}$$

and m and n are the numbers of points of L and R, respectively.

By substituting Equation (25) into (24), the five coefficients of $h(X)$ in Equation (23) are obtained so that the quadratic decision boundary $h(X)$ is generated. The ALV can navigate along the derived collision-free path $h(X)$ to achieve safe navigation by person following.

4. Experimental Results

The proposed method has been implemented on a real ALV to follow a walking person, as shown in Fig. 9. The ALV, as shown in Fig. 10, is four-wheeled with two motors controlling the front and rear wheels, respectively. The width and the length of the ALV are 40 cm and 120 cm, respectively. The length between the front wheels and the rear wheels of the ALV is 82 cm. The structure of the vehicle system is introduced in detail in [5]. In our experiments, images taken by the image frame grabber are 512 × 486 pixels in resolution. The velocity of the vehicle is 31.75 cm per second that is equivalent to 1.14 km per hour. This velocity is acceptable in many applications in indoor environments. In addition, the vehicle modifies the turning angle every 1.5 seconds. Because the speed of the person has to be smaller than that of the ALV, the maximum speed of the person is about 31.75 cm/sec. Besides, in the experiment, one situation that the followed person does not appear in the captured image may cause the instability of the ALV. This case is discussed and solved using a proposed method in [14].

An example of the captured image is shown in Fig. 11, in which the detected baselines are drawn as white lines. Figures 12 through 16 show some experimental results in a real indoor corridor environment. Each of Figs. 12 and 13 include two parts (a) and (b). Part (a) is a real image captured by a wide-angle camera. The white lines in (a) are composed of the patterns of the detected obstacles. Part (b) shows the x-y plane of the VCS. All the patterns representing the obstacles, the followed

person, and the vehicle in the VCS are shown as white points. The generated collision-free path that goes through the origin of the VCS and the person position is also shown in (b). According to the generated path, a modification of the turning angle can be computed using the path following method described in [5].

As shown in Fig. 12, two walls and the followed person are detected and the collision-free path is correctly generated. As shown in Fig. 13, the generated collision-free goes through the way between the wall and the obstacle. If the distance between the followed person and obstacles is smaller than a width of the vehicle body, the person patterns are not treated as input to generate a collision-free path. As shown in Fig. 5.14, since the distance between the followed person and a wall is smaller than the width of the vehicle body, the generated collision-free path does not pass through the location of the followed person so that the vehicle will avoid collision with the wall when it navigates along the path. An experimental result showing the ALV successively following a walking person without collision in the corridor is shown in Fig. 15. The sequence of the captured images is shown in Fig. 15(a). The corresponding x-y planes of the VCS with respect to the captured images are illustrated in Fig. 15(b). Fig. 16 includes another sequence of experimental images showing the ALV following a person in the corridor.

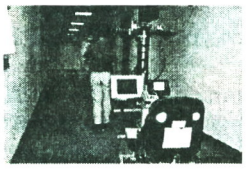


Fig. 9. The vehicle is guided automatically to follow a person who walks in front of the vehicle.

Fig. 10. The prototype ALV used in this study.

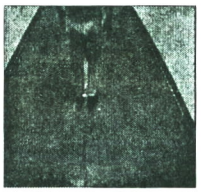
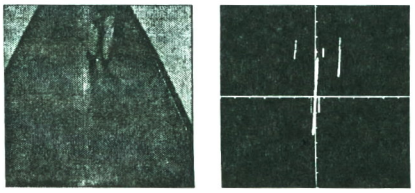
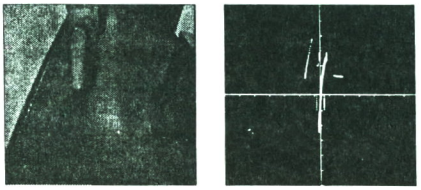


Fig. 11. An example of baseline detection results.



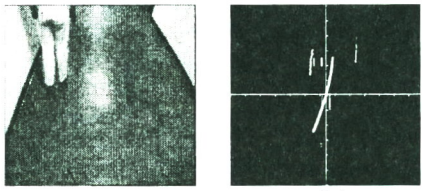
(a) (b)

Fig. 12. An experimental result of the generated collision-free path. (a) Detection result of walls and the followed person. (b) Top view of obstacles and a generated path.



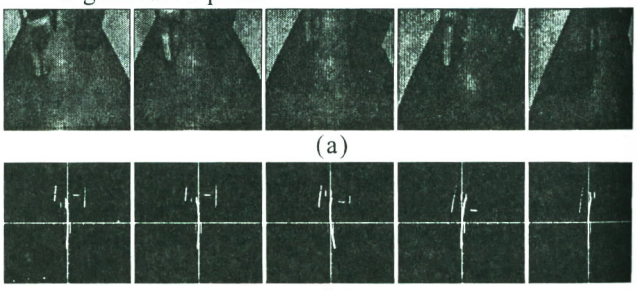
(a) (b)

Fig. 13. Another experimental result of generated collision-free path. (a) Detection result of a wall, the followed person, and an obstacle on the ground of the corridor. (b) Top view of obstacles and a generated path.



(a) (b)

Fig. 14. An experimental result shows that the distance between the followed person and a wall is smaller than the width of the vehicle body. (a) Detection result of walls and the followed person. (b) Top view of obstacles and a generated path.



(a)

(b)

Fig. 15. A sequence of experimental images in the corridor of a real indoor environment. (a) Captured images. (b) Top views of obstacles and generated collision-free paths with respect to the captured images.

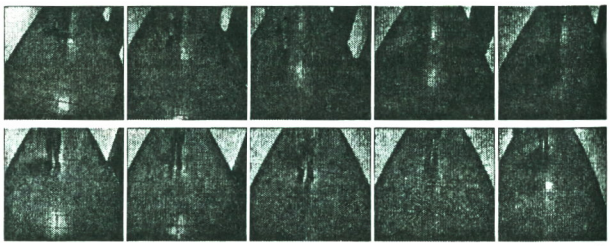


Fig. 16. A sequence of experimental images in the corridor of a real indoor environment.

5. Conclusions

In this study, an obstacle avoidance method for use in person following for vision-based ALV guidance has been proposed. The approach is based on the use of vehicle location estimation and the quadratic pattern classifier, and enables the ALV to follow safely a walking person through corridor environments with obstacles by navigating along a derived collision-free path. In this approach, the estimated translation of the vehicle location from the previous sampling intant to the current one is used to predict the person location at the current sampling instant. The object closest to the predicted person location is regarded as the followed person and the remaining objects are regarded as obstacles. After obtaining the person's location from extracted objects by the proposed person detection method, a collision-free path can be generated by a path generation method. The navigation path is generated

from a quadratic classifier that uses the vehicle and all of the objects in the image as input patterns. All patterns are categorized into two classes to be the input to the classifier using a pattern generation method. The collision-free navigation path is designed for ALV guidance in such a way that the ALV not only can keep following the person but also can avoid collision with nearby obstacles. This approach has been implemented on a real ALV. Successful and safe navigation sessions confirm the feasibility of the approach.

References

[1] A. Ohya, A. Kosaka and A. Kak, "Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing," *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems*, Vol. 2, pp. 704-711, Grenoble, France, September 1997.

[2] L. M. Lorigo, R. A. Brooks and W. E. L. Grimsou, "Visually-guided obstacle avoidance in unstructured environments," *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems*, Vol. 1, pp. 373-379, Grenoble, France, September 1997.

[3] Y. G. Yang and G. K. Lee, "Path planning using an adaptive-network-based fuzzy classifier algorithm," *13th International Conference for Computers and Their Applications*, pp. 326-329, Honolulu, HI, U.S.A., March 1998.

[4] R. Biewald, "Real-time navigation and obstacle avoidance for non-holonomic mobile robots using a human-like conception and neural parallel computing," *International Workshop on Parallel Processing by Cellular Automata and Array*, pp. 232-240, Berlin, Germany, September 1996.

[5] C. H. Ku and W. H. Tsai, "Obstacle avoidance for autonomous land vehicle navigation in indoor environments by quadratic classifier," *IEEE Transactions on Systems, man, and Cybernetics – Part B: Cybernetics*, Vol. 29, No. 3, pp. 416-426, June 1999.

[6] O. Masoud and N. P. Papanikolopoulos, "Robust pedestrian tracking using a model-based approach," *Proceedings of IEEE Conference on Intelligent Transportation Systems*, Boston, MA, U.S.A.. 9-12 Nov., pp. 338-343, 1997.

[7] J. Denzler and H. Niemann, "Real-time pedestrian tracking in natural scenes," *Proceedings of 7th International Conference on Computer Analysis of Images and Patterns*, Berlin, Germany, 10-12 Sept., pp. 42-49, 1997.

[8] L. Trassoudaine, S. Jouannin, J. Alizon and J. Gallice, "Tracking systems for intelligent road vehicles," *International Journal of Systems Science*, Vol. 27, No. 8, pp. 731-743, 1996.

[9] V. F. Munoz and A. Ollero, "Smooth trajectory planning method for mobile robots," Proceedings of Conference on Computational Engineering in Systems Applications, pp. 700-705, Lille, France, 9-12 July, 1996.

[10] Z. Shiller and W. Serate, "Trajectory planning of tracked vehicles," *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, Vol. 117, No.4, pp. 619-624, 1995.

[11] C. H. Ku and W. H. Tsai, "Smooth vision-based autonomous land vehicle navigation in indoor environments by person following using sequential pattern recognition," *Journal of Robotic Systems*, Vol. 16, No. 5, pp. 249-262, May 1999.

[12] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *IEEE Transaction on Automatic Control*, Vol. 38, No. 5, pp. 700-716, 1993.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Second edition, Academic Press, San Diego, U.S.A., 1990.

[14] C. H. Ku and W. H. Tsai, "Robust trajectory planning in person following for vision-based autonomous land vehicle guidance by visual field model and visual contact constraints," *Proceedings of 1999 Conference on Computer Vision, Graphics, and Image Processing*, Taipei, Taiwan, Republic of China, pp. 771-780, August 1999.