AUTOMATIC ENVIRONMENT LEARNING AND PATH GENERATION FOR INDOOR AUTONOMOUS LAND VEHICLE GUIDANCE USING COMPUTER VISION TECHNIQUES*

Feng-Min Pan (潘豐民)† and Wen-Hsiang Tsai (蔡文祥)§

†Institute of Computer and Information Science
Microelectronics and Information Science and Technology Research Center
National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China

摘 要

本論文提出一套使自動車能在建築走廊中自動作環境學習、路徑產生及自動導引的整合技術。我們運用電腦視覺技術在走廊內作自動車定位。使用到的環境特徵包括走廊中的踢線及角點。我們提出一種模式比對的方法來找出最可信的環境特徵配對，並利用此一環境特徵配對來作自動車定位。我們也提出一套環境學習的方法來記錄自動車在學習階段中所經過的環境特徵，以期達成環境學習的目的。此外並提出一套路徑產生的方法，找出一條能讓自動車作平穩航行的合理路徑。最後，我們製作一輛實際的自動車作測試，並達到了自動車在走廊中作平順且安全航行的目的。經過了多次成功的實驗，我們證明了此套方法的可行性。

關鍵字：環境學習，路徑產生，自動導引。

ABSTRACT

An integrated approach to automatic model learning and path generation for vision-based autonomous land vehicle (ALV) guidance in building corridors is proposed. Computer vision techniques are utilized to locate an ALV in corridors. Used environment features include baseline segments and corners on walls. The ALV location work is accomplished by a matching scheme for finding possible matching corner pairs or baseline segment pairs. Furthermore, strategies for model learning are proposed in order to acquire automatically the environment features that the ALV pass through in a learning stage. And techniques of path generation are also proposed to generate paths automatically for guiding the ALV in indoor environments. Finally, a real ALV was constructed as a testbed, and smooth and safe navigation sessions can be achieved. Lots of successful experiments confirm the feasibility of the proposed approach.

Key Words: Environment learning, Path generation, Automatic guidance.

1. Introduction

1.1 Motivation

Guidance of autonomous land vehicles (ALV's) by computer vision techniques has been intensively studied in recent years. This tendency is due to the fast development of computer vision techniques. Furthermore, the vision-based method, which is more similar to the human vision function, is a more intelligent and flexible way for ALV guidance. The goal of this research is to accomplish the works of environment model learning, path generation, and automatic guidance in an indoor environment using computer vision techniques. It is desired that without human involvement in the measurement of the environment in which the ALV will navigate, smooth paths can be generated and safe navigation can be achieved.

Several successful ALV systems have been established for various purposes. A reactive robotic control system which incorporates aspects of learning momentum to improve the system's ability to successfully navigate in dynamic environments was proposed by Arkin, et al. [1,2]. The system decompose actions into behaviors in order to produce rapid real-time sensory responses. Schema-based robotic control employed in such a system is one form of reactive control systems. They are multiple concurrent processes that operate in conjunction with associated perceptual schemas and contribute to the overa. The CMU mobile robot system[3,4] were equipped with multiple sensors, including range sensors, odometers, color TV cameras, etc. Outdoor range data from a laser range scanner is used for 3-D feature extraction , map building, and object recognition. Vision algorithms including region analysis and line tracking were successfully used in outdoor navigation.

To conduct this research work, three ALV's have been constructed by Tsai, et al. [5-9]. In Chang and Tsai[5], a line following approach was proposed. The baseline in a corridor which is the intersection of a wall and the ground is extracted and used. A method based on a principle similar to that of the cross ratio is used to find the baseline location with respect to

* This work was supported partially by NSC under Grant NSC 82-0404-E-009-157.

§ To whom all correspondence should be addressed.

the vehicle. The ALV follows long baselines to perform indoor navigation. In Ku and Tsai[6], a model-based navigation approach was proposed, and the corridor contour is used to match the model and the input pattern extracted from the video camera image. So, the global location of the ALV can be known. In Ke and Tsai[7], a corner tracking approach was proposed, in which corners in the building are stored as the model, and a method for rotating the camera to track corners was proposed. The idea is to select visible right-angle corners in indoor environments as guiding marks. A new guidance approach by model matching was proposed in Cheng and Tsai[8]. Two laser light sources were employed to reduce the processing time for computing vertical line positions by triangulation. A matching scheme using distance weight correlation is proposed. The vertical line position information was matched with the model to locate the ALV accurately. In Su and Tsai[9], an model-based navigation and collision avoidance in building corridors and elevators was proposed. The multiple corner position information was matched with the model to locate the ALV accurately, the reflex photoelectric sensors were used for obstacle avoidance and a radio equipment was employed to control the elevator operations of lifting up, lifting down, door closing, and door opening so that the ALV can accomplish elevator entering automatically.

An ALV navigating in an unknown environment in general must have the capabilities to explore the environment with its sensors, construct an abstract representation of the environment to generate a path to the goal, and furthermore, navigate smoothly and safely in the learned environment. Vision-based model matching is a reasonable approach to achieving this goal. Selecting stable environment features and developing effective methods to extract these features are the most important key points to model-based ALV guidance. In this study, the positions of the baseline segments and corners in indoor environments are selected for learning and guidance of ALV's in building corridors because they are abundant and easily visible in buildings and are convenient for use in reference models.

In the remainder of this part, we investigate the location of environment features, baseline segments and corners, in building corridors in Section 2. The ALV location and model learning techniques are described in Section 3. Techniques for path generation are described in Section 4. The ALV location and guidance techniques are described in Section 5. Image processing techniques and experimental results are described in Section 6. Conclusions and suggestions for further study can be found in Section 7.

2. Locating Environment Features for Model Learning and ALV Guidance

2.1 Coordinate Systems and Transformations

Several coordinate systems and coordinate transformations will be defined here for use in the following sections. The coordinate systems are shown in Figure 2. The camera coordinate system (GCS), 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 contact point of the front wheel of the vehicle and the ground. The x-axis and the y-axis are placed on the ground and are parallel to the short and long sides of the vehicle body, respectively. To describe the model, we need a third coordinate system, called the global coordinate system (GCS), which is denoted as x'-y'-z'. The x'-axis and y'-axis are defined to lie on the ground.

The GCS is assumed to be fixed all the time, while the VCS is moving with the vehicle during navigation. The location of the vehicle can be assured once the relation between the VCS and the GCS is found. Since the vehicle is on the ground all the time, the z-axis and the z'-axis can be ignored. That is, the relation between the 2-D coordinate systems x-y and x'-y' is sufficient for determining the position and orientation of the vehicle.

The transformation between the two 2-D coordinate systems x-y and x'-y' can be written as follows:

$$(x'\ y'\ 1) = (x\ y\ 1)\begin{bmatrix} cosw & sinw & 0 \\ -sinw & cosw & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x'_p & y'_p & 1 \end{bmatrix} \tag{2.1}$$

where (x'_p, y'_p) is the translation vector from the origin of x'-y' to the origin of x-y and w is the relative rotation angle of x-y with respect to x'-y', as shown in Figure 3. The vector (x'_p, y'_p) and the angle w determine the position and the direction of the vehicle in the GCS, respectively.

The transformation between the CCS and the VCS can be written in terms of homogenous coordinates[8] as

$$(u\ v\ w\ 1) = (x\ y\ z\ 1)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_d & -y_d & -z_d & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where

$$
\begin{aligned}
r_{11} &= \cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi, \\
r_{12} &= -\sin\theta\cos\phi, \\
r_{13} &= \sin\theta\sin\phi\cos\psi - \cos\theta\sin\psi, \\
r_{21} &= \sin\theta\cos\psi - \cos\theta\sin\phi\sin\psi, \\
r_{22} &= \cos\theta\cos\phi, \\
r_{23} &= -\cos\theta\sin\phi\cos\psi - \sin\theta\sin\psi, \\
r_{31} &= \cos\phi\sin\psi, \\
r_{32} &= \sin\phi, \\
r_{33} &= \cos\phi\cos\psi.
\end{aligned}
\tag{2.3}
$$

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

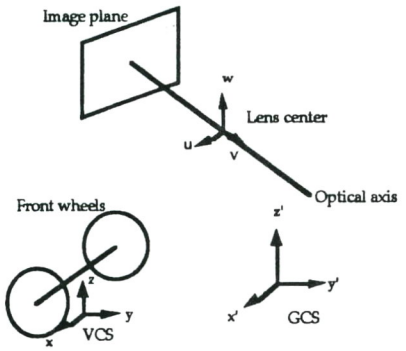


Figure 2: The camera coordinate system u-v-w, the vehicle coordinate system x-y-z, and the global coordinate system x'-y'-z'.

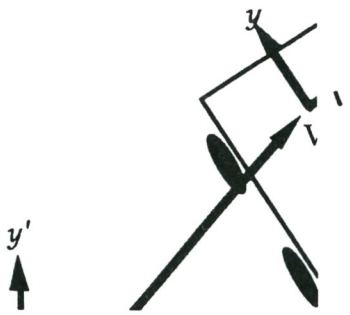


Figure 3: The relation between 2-D coordinate systems x-y and x'-y' represented by a translation vector (x_p', y_p') and a relation angle w.

2.2 Locating Baseline Segments and Corners

In this study, the features of baseline segments and corners in building environments are used to achieve the proposed approach to model learning, pathgeneration and ALV guidance. In this section, the geometric properties of the baseline, whose height is fixed, are used to calculate the VCS coordinates of thebaseline segments detected in an input image. The formula for calculating the VCS coordinates of a corner point which is located on a baseline is derived in the following.

As shown in Figure 4, after backprojecting a point P in the image into the VCS, we can get a line L which passes the lens center and P. The intersection point of the line L and the horizontal plane going through the baseline is the corresponding space point of P which is what we desire. Denote the point as P'.

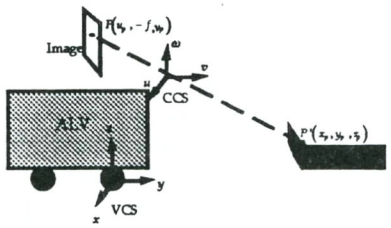


Figure 4: Configuration of the system for finding the back-projection point for an image pixel.

The equation of the horizontal plane can be set to be $z = h$ by measuring height h of the baseline of the wall before navigation. Assume that point P in the image has the coordinates $(u_p, -f, v_p)$ in the CCS where (u_p, v_p) indicates its position in the image and f is the focal length. Using Eq. 2.2, we get the VCS coordinates (x_p, y_p, z_p) of point P in the image as

$$
\begin{aligned}
x_p &= u_p(\cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi) + f(\sin\theta\cos\phi) + \\
&\quad v_p(\sin\theta\sin\phi\cos\psi - \cos\theta\sin\psi) + x_d. \\
y_p &= u_p(\sin\theta\cos\psi - \cos\theta\sin\phi\sin\psi) - f(\cos\theta\cos\phi) - \\
&\quad v_p(\cos\theta\sin\phi\cos\psi + \sin\theta\sin\psi) + y_d. \\
z_p &= u_p(\cos\phi\sin\psi) - f\sin\phi + v_p(\cos\phi\cos\psi) + z_d.
\end{aligned}
$$

where (x_d, y_d, z_d) are the coordinates of the lens center in the VCS. On the other hand, the equation of line L is

$$
\frac{x - x_d}{x_p - x_d} = \frac{y - y_d}{y_p - y_d} = \frac{z - z_d}{z_p - z_d}. \tag{2.4}
$$

Since point P' is a corner point, by substituting $z = h$ into Eq. 2.4, the desired VCS coordinates (x'_p, y'_p, z'_p) of point P' can be solved to be:

$$x_{p'} = x_d + \frac{h - z_d}{z_p - z_d}(x_p - x_d),$$
$$y_{p'} = y_d + \frac{h - z_d}{z_p - z_d}(y_p - y_d),$$
$$z_{p'} = h. \qquad (2.5)$$

2.3 Matching Corners with Learned Model

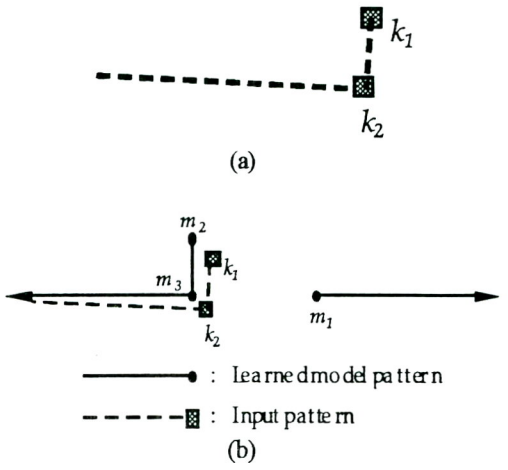
Applying the formulas derived in Section 2.2, the VCS coordinates of the corners detected from the input image can be calculated by Eq. 2.2, and the GCS coordinates of these corners can also be calculated according to the estimated ALV location (described in Section 5.2) by Eq. 2.1. Since the estimated ALV location is not very accurate due to machine errors, the GCS coordinates of the detected corners are not very accurate, either. Proposed in this section is a method that matches these detected corners with the learned model to get accurate GCS coordinates of these corners.

More specifically, let the estimated GCS coordinates of the input corners constitute an input pattern, which we denote by a corner set K = $\{k_1, k_2, \cdots, k_p\}$, where each corner $k_i = (ks_i, kl_i, kr_i)$ consists of its location (ks_i), the angle of its left line segment (kl_i), and the angle of its right line segment (kr_i). When a neighboring line segment of a corner does not exist, a pseudo angle is created for it and is set to 10 (>2 π). Within a reasonable angle difference tolerance range to the neighboring line segments of a corner, for each k_i a corresponding learned model pattern M can be extracted from the entire learned model. Furthermore, the distance tolerance range can be determined according to the inaccuracy of the mechanical devices. Now, consider any input corner k_i in K. There exists a corresponding set of corners in the learned model denoted by M = $\{m_1, m_2, \cdots, m_q\}$, where each corner $m_j = (ms_j, ml_j, mr_j)$, consisting of three elements (similar to (ks_i, kl_i, kr_i)), is possible to match k_i to form a pair of (k_i, m_j). Each corner m_j in M is within the area of a circle whose center is k_i and the radius is the distance tolerance.

Let M and K be two corner-type patterns to be matched in a two-dimensional space. The Cartesian product K × M which gives all possible match pairs is the set of all pairs of the form (k_i, m_j) where $k_i \in$K and $m_j \in$M. After M and K are superimposed, for each pair (k_i, m_j) in K × M, if the angle difference dl_{ij} between kl_i and ml_j is larger than a preselected value V that defines an angle difference limit within the tolerance range or if the angle difference dr_{ij} between kr_i and mr_j is larger than the value V, then we discard the pair (k_i, m_j) in K × M. Afterwards, for each pair (k_i, m_j) in K × M, let dc_{ij} denote the Euclidean distance between k_i in K and m_j in M. Then the correlation measure C_{ij} is defined to be

$$C_{ij} = \begin{cases} 1/(dc_{ij}^2 + 1) & \text{if } dc_{ij} < U, \\ 0 & \text{otherwise,} \end{cases} \qquad (2.6)$$

where U is an experimental constant value that defines a distance limit within which the closest feature corner m_j in M of k_i is searched for.



Possible match pairs :

$\{(k_1,m_1), (k_1,m_2), (k_1,m_3), (k_2,m_1), (k_2,m_2), (k_2,m_3)\}$

(c)

The result of the best match pair :

(k_2,m_3)

(d)

Figure 5: The steps of the corner matching procedure(Algorithm(1).
(a) Two corners are detected.
(b) The sensed pattern $\{k_1,k_2\}$ and the model pattern $\{m_1,m_2,m_3\}$.
(c) Possible match pairs.
(d) The matching result.

An example of the matching result is shown in Figure 5. Two corners are detected in the image, and are shown in Figure 5(a). Note that only corners are drawn with small circles. Within an tolerance error range, the input pattern and the learned model pattern are superimposed as shown in Figure 5(b). The possible match pairs are shown in Figure 5(c). Note that each corner in the input pattern is assigned to a model pattern. The result of matching is shown in Figure 5(d). Each corner in the input pattern is now assigned to a single corner in the model pattern.

2.4 Matching Baseline Segments with the Learned Model

As mentioned in Section 2.3, we can get the VCS coordinates and the GCS coordinates of the baseline segments detected from the input image by Eq. 2.2, and Eq. 2.1. Similarly, the GCS coordinates of the detected baseline segments are not very accurate owing to machine errors. Therefore, what proposed in this section is a method that matches these detected baseline segments with the learned model to get the accurate GCS coordinates of these baseline segments.

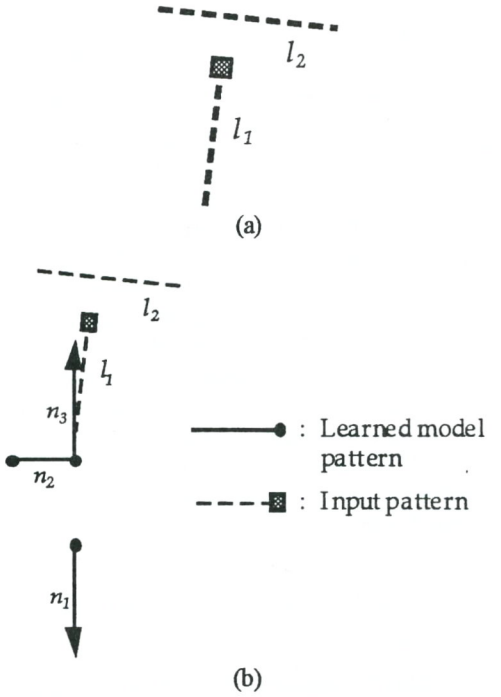
More specifically, let the estimated GCS coordinates of the input baseline segments constitute an input pattern, which we denote by a line set L = $\{l_1, l_2, \cdots, l_p\}$, where each $l_i = (ls_i, la_i)$ denotes the location (ls_i) and the angle (la_i) of a line. Now, consider a line segment l_i in L. The line segments in the learned model pattern N is denoted by a line set N = $\{n_1, n_2, \cdots, n_q\}$, where each $n_j = (ns_j, na_j)$, consisting of two elements (similar to (ls_i, la_i)), is possible to match l_i to form a pair of (l_i, n_j).

As mentioned in Section 2.3, let N and L be two line-type patterns to be matched in a two-dimensional space. After N and L are superimposed, for each pair (l_i, n_j) in L × N, if the angle difference da_{ij} between la_i and na_j is larger than the experimental value V that defines an angle difference limit within the tolerance range , then we discard the pair (l_i, n_j) in L × N. Afterwards, for each pair (l_i, n_j) in L × N, we define the distance d_{ij} of a middle point k_i in l_i to be the distance from k_i to the line segment n_j in N. And define the correlation measure L_{ij} to be

$$L_{ij} = \begin{cases} 1/(d_{ij}^2 + 1) & \text{if } d_{ij} < U; \\ 0 & \text{otherwise,} \end{cases} \quad (2.7)$$

where U is an experimental constant value that defines a distance limit within which the closest line segment n_j in N of k_i in l_i is searched for.

An example of the matching result is shown in Figure 6. Two baseline segments are detected in the input image, and are shown in Figure 6(a). Note that only baseline segments are drawn with small circles. Within tolerance error range, the input pattern and the learned model pattern are superimposed as shown in Figure 6(b). The possible match pairs are shown in Figure 6(c). Note that each baseline segment in the input pattern is assigned to a model pattern. The result of matching is shown in Figure 6(d). Each baseline segment in the input pattern is now assigned to a single baseline segment in the model pattern.



Possible match pairs :

$\{ (l_1,n_1), (l_1,n_2), (l_1,n_3), (l_2,n_1), (l_2,n_2), (l_2,n_3) \}$

(c)

The result of the best match pair :

(l_1,n_3)

(d)

Figure 6: The steps of the baseline segment matching procedure (Algorithm 2). (a) Two baseline segments are detected. (b) The sensed pattern $\{l_1,l_2\}$ and the model pattern $\{n_1,n_2,n_3\}$. (c) Possible match pairs. (d) The matching result.

3. Strategies for Model Learning

3.1 Locating ALV by Matching Result

The idea of our approach is illustrated in Figure 7, which says that if we can select some features such that the VCS information and GCS information can be

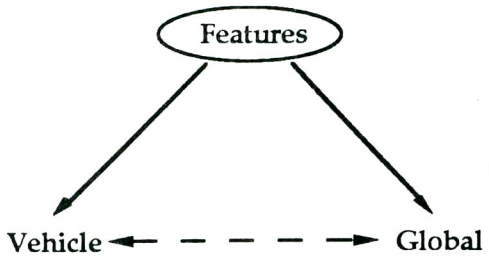


Figure 7: Principle of ALV location.

known, then the relation between the VCS (vehicle) and GCS (world) can be determined. For this, we select the position of a corner as the feature to find the position of the ALV, and select the position of a baseline segment as the feature for finding the ALV slant angle. Recall that the matching steps begin with finding the VCS coordinates of features and end with matching the features with the learned model to find the GCS coordinates of these features. So the position of the feature with respect to the VCS and with respect to the GCS can be known, and this will be discussed later.

3.1.1 Locating ALV by Matched Corners

3.1.1.1 Determination of ALV slant angle

As mentioned in Section 2.1, the ALV location is described by the ALV slant angle ω and the ALV position (x,y). To derive the ALV location in the GCS, we rewrite Eq. 2.1 as follows:

$$\begin{aligned} x' &= x\cos\omega - y\sin\omega + x'_p, \\ y' &= x\sin\omega + y\cos\omega + x'_p. \end{aligned} \quad (3.1)$$

In our approach, the ALV slant angle ω will be solved first, and the ALV position (x'_p, y'_p) is solved accordingly.

After a corner detected from the input image is matched with the learned model, the baseline segment which intersects this corner is used to determine the slant angle of the ALV. Assume that the VCS coordinates of the two points are computed to be (x_1,y_1) and (x_2,y_2) (ignore the z coordinates), then the slope m_1 of the baseline segment in the VCS can be computed by

$$m_1 = \frac{y_1 - y_2}{x_1 - x_2}.$$

Because the GCS coordinates of each corner and both of its neighboring line segments have been stored in the learned model, if the corner is matched with the *i*th corner in the learned model, then we can find the matched line segment which is the left or right line segment of the corner in the learned model and the GCS coordinates of the two end points of this line segment can be determined. Assume that the GCS coordinates of the two end points are (x'_1,y'_1) and (x'_2,y'_2) (ignore the z' coordinates), then the slope m_2 of the baseline segment in the GCS can be solved to be

$$m_2 = \frac{y'_1 - y'_2}{x'_1 - x'_2}.$$

After the slopes of the baseline segment in the VCS and in the GCS are determined, the slant angle of the ALV can be derived. This is illustrated in Figure 8. The angle θ_1 between the baseline segment and the positive x-axis of the VCS and the angle θ_2 between the baseline segment and the positive x'-axis of the GCS can be derived to be

$$\theta_1 = \tan^{-1} m_1,$$

$$\theta_2 = \tan^{-1} m_2,$$

respectively. Then the slant angle ω of the ALV can be solved to be

$$\omega = \theta_2 - \theta_1.$$

3.1.1.2 Determination of ALV global position

The position of a detected corner in the VCS and in the GCS are used to determined the ALV position. Note that the VCS coordinates (x,y) of the corner can be determined as discussed in Sec. 2.2. After matching with the model as discussed in Sec. 2.3, the GCS coordinates (x',y') of the corner are also determined, and the ALV slant angle ω has been solved as mentioned above. By substituting ω, (x,y), and (x',y') into Eq. 3.1, the position of the ALV can be determined to be

$$x'_p = x' - x\cos\omega + y\sin\omega,$$
$$y'_p = y' - x\sin\omega - y\cos\omega.$$

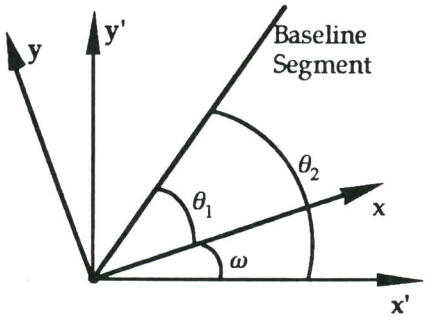


Figure 8: The slopes of baseline segment in the GCS and in the VCS are used to solve the slant angle of ALV.

3.1.2 Locating ALV by Matched Baseline Segments

3.1.2.1 Determination of ALV slant angle

Similarly, as mentioned in Section 3.2.1.1, after the slopes of the baseline segment in the VCS and in the GCS are determined, the slant angle of the ALV can be derived. The angle θ_1 between the baseline segment and the positive x-axis of the VCS and the angle θ_2 between the baseline segment and the positive x'-axis of the GCS can be derived to be

$$\theta_1 = tan^{-1}m_1,$$
$$\theta_2 = tan^{-1}m_2,$$

respectively. Then the slant angle ω of the ALV can be solved to be

$$\omega = \theta_2 - \theta_1.$$

3.1.2.2 Determination of ALV global position

The position of a detected baseline segment in the VCS and in the GCS are used to determined the ALV position. As shown in Figure 9, assume that we have found the slant angle ω of the ALV at A, and after a cycle time the slant angle ω' of the ALV at B is also known. Then, the difference of the two slant angles $\Delta\omega$ can be determined to be

$$\Delta\omega = \omega' - \omega$$

and the actual turn angle of the front wheels δ can be computed to be

$$\delta = sin^{-1}\left(\frac{d\Delta\omega}{S}\right)$$

where d is the distance between the front wheels and the rear wheels.

The distance S can be detected from the odometer in this cycle. Once the values of δ and S, and the location (x,y) of the ALV at A are known, the current actual location (x',y') of the ALV at B can be determined and the detail is described in Section 5.2.

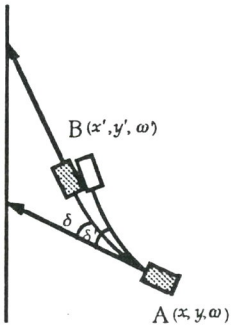


Figure 9: Illustration of determination of ALV global position, where δ' denotes the desired turn angle of front wheel and δ denotes the actual turn angle of front wheel.

3.2 Adjustment of Global Model by Learned Local Model

Applying the approach proposed in Section 3.2, the actual ALV location can be determined. The local environment features can also be derived by the strategies described in Section 2.2, so we can adjust the learned global model by this current learned model.

Once the actual ALV location is determined, we can recompute the more precise global coordinates of the local features detected from the input image. For example, as shown in Figure 10, some features, such as f_1, f_2, f_3, f_4, f_5, and f_6, can be obtained. Furthermore, there are some features that do not match any feature in the learned global model, and we want to add these features to the global model. The adjustment of the global model means to insert knowledge about neighboring environment features into the ALV. For example, as shown in Figure 11, some features, such as f_2, f_3, f_4, f_5, and f_6, that do

not match any feature in the global model are found by using the strategies which described in Section 2.3, and we want to add these features to the global model in order to achieve the objective of ALV learning.

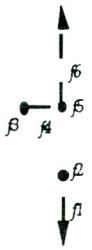


Figure 10: the accurate coordinates of these environment features detected from the input image.

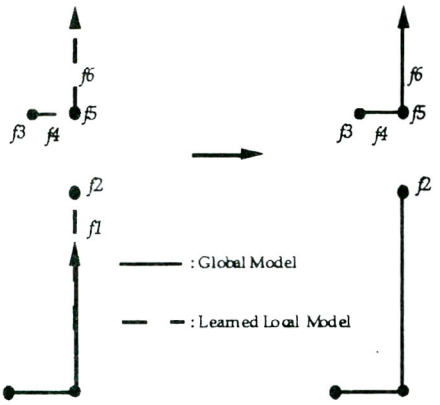


Figure 11: Adjustment of global model by learned local model.

4. Strategies for Path Generation

After learning and before ALV navigation, we have to generate a path so that the ALV can navigate along it. In this Section, two strategies are proposed to determine straight paths and circular paths that are followed by the ALV.

4.1 Determination of Straight Paths

Based on the baseline segments determined by the method described in Section 4.1 and the rough path that the ALV passes through in the learning phase, we may find some longer straight paths for future ALV navigation. More specifically, the rough path constitutes an input pattern, which we denote by a node set N = $\{n_1, n_2, \cdots, n_p\}$, where each node n_i denotes a location in this rough path. For example, the relation between the slant angle of the ALV and the angle of a baseline segment is shown in Figure 12, where the x-axis denotes the cycles in the learning phase and the y-axis denotes the angles which determine the directions of the ALV or the baseline segments in the GCS. The proposed method is described as follows.

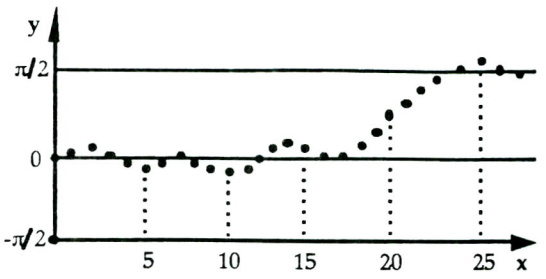


Figure 12: The relation between the slant angle of the ALV and the angle of a baseline segment in the model.

First, an average value A is computed by taking the sum of the angles of successive 4 nodes in N and divide it by 4. Next, we find out all the baseline segments such that the angle difference between the angle of each baseline and the average value A is smaller than a tolerance range V.

After the candidate baseline segments are found out, we select the one to which the distance from the average location of the 4 successive nodes in N is smaller than a tolerance range U, and decide accordingly a path segment such that the distance from this path segment to the selected baseline segment is the average distance from these nodes to the baseline segment, as shown in Figure 12.

At last, the processed nodes are represented by the path segments that we store in a set P_S, and we use these path segments to be the desired straight paths. An example of the result of determination of straight paths is shown in Figure 13.

4.2 Determination of circular paths

After the straight paths in P_S are determined, it is necessary to find the circular paths in a turning area for ALV guidance. Without loss of generality, we consider that a circular path exist between two successive straight paths whose directions are different. We will now derive a circular path from two successive straight paths.

As shown in Figure 14, let any two successive straight paths which are determined in Section 4.2 be denoted by *fl* and *rl*,. The intersection point Q of *fl* and *rl* can be solved. Then a line *cl* as shown in the figure from *fl* and *rl* can also be found. Furthermore, let P be a point which is moved in *cl* from the point Q to a certain location in order to meet the condition that the distance from P to *fl* and the distance from P to *rl* are both a fixed constant. Once this condition is satisfied, the point P is the center of a circle whose radius R is known in advance. Let the point on *fl*

onto which is projected be denoted by fp and the point on *rl* onto which is projected be denoted by rp. After the two points, fp and rp, are found, we can acquire an arc whose radius is the distance from the point fp to the point rp. This arc then is the desired circular path. An example of the result of determination of circular paths is shown in Figure 15.

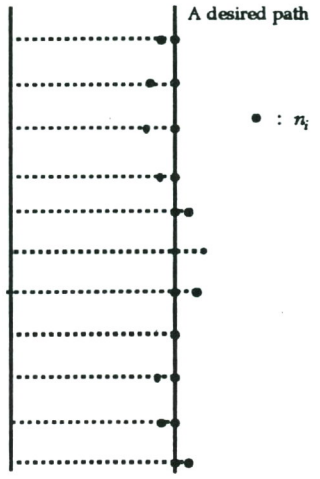


Figure 13: The desired path is decided by some nodes in N.

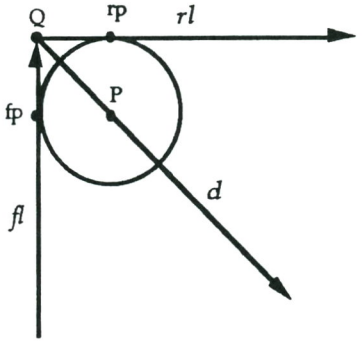


Figure 14: The overview of the determination of a circular path.

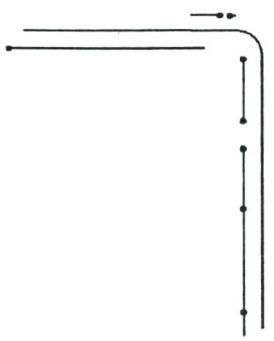


Figure 15: An example of the result of path generation.

5. Strategies for ALV Guidance

5.1 Wheel adjustment for straight path navigation

In general, a criterion for adjusting the driving wheel direction δ to a straight path is required so as to keep the vehicle close to a desired path in each navigation session. This assures safe and smooth ALV navigation. The method is described as follows. As shown in Figure 18, given a reasonable moving distance S and the turn angle of the ALV front wheels, the location of the ALV can be determined as discussed above. Given a straight path P_S, we define $D^F_{P_s}(\delta)$ to be the distance from the ALV to the given path P_S after the ALV traverses a distance S with the turn angle δ, as shown in Eq. 5.4. So the value of $D^F_{P_s}(\delta)$ is determined by the turn angle δ. A measure L_{P_s} of the ALV to the given path is defined to be

$$L_{P_s} = D^F_{P_s}(\delta) + D^B_{P_s}(\delta). \qquad (5.1)$$

To find the turn angle of the front wheel to drive the ALV as close to the path as possible, a range of possible turn angles are searched. An angle is hypothesized each time, and the value of L_{P_s} is calculated accordingly. The angle that produces the minimal value of L_{P_s} is then used as the turn angle for safe navigation.

5.2 Wheel adjustment for circular path navigation

Similarly, a criterion for adjusting the driving wheel direction δ to a circular path is also required so as to keep the vehicle close to a desired path in each navigation session. We propose a method for this as follows. As shown in Figure 17, given a reasonable moving distance S and the turn angle of the ALV front wheels, the location of the ALV can be determined as discussed above. Given a circular path P_C, we define $D^F_{P_c}(\delta)$ to be the distance from the ALV to the given path P_C after the ALV traverses a distance S with the turn angle δ. A measure L_{P_c} of the ALV to the given path is also defined to be

$$L_{P_c} = D^F_{P_c}(\delta) + D^B_{P_c}(\delta). \qquad (5.2)$$

Similarly, we find the turn angle δ that produces the minimal value L_{P_c}. Then we drive the ALV as close and smooth to the path as possible according to δ.

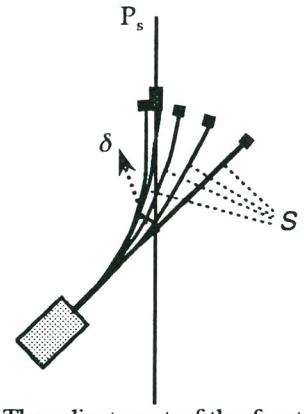


Figure 16: The adjustment of the front wheels in a straight path.

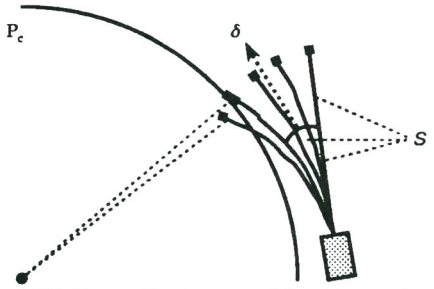


Figure 17: The adjustment of the front wheels in a circular path.

6. Image Processing and Experimental Results

6.1 Extraction of Baseline segments and Corners

The purpose of image processing in this study is mainly to extract corner points and baseline segments going through the points, and to compute the equations of the baseline segments.

When an input image is processed, the first thing is to extract the candidate pixels on the black basebands of the wall. A threshold value is preselected to find candidate pixels in column scanning. In practice, however, strong lighting near the windows will cause the failure of scanning if only a fixed threshold value is used for thresholding the entire input image. So, we divide the image into eight vertical strips and decide for each strip a threshold value T_i computed to be the average gray value G_i of the strip substracted by an experimental value E, which is described as follows:

$$T_i = G_i - E.$$

At last, an example of baseline segment extraction and corner extraction results is shown in Figure 18.

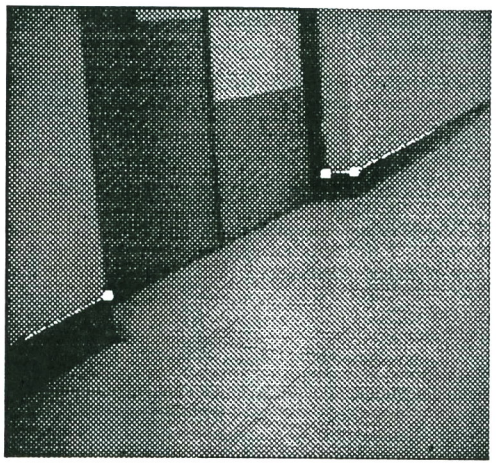


Figure 18: An example of extraction result of corners and baseline segments.

6.2 Experimental Results

ALV learning experiments were performed in a building corridor in National Chiao Tung University. And the environment model is used for our experiments. Figure 19 shows the positions of the features in the environment. Each corner is marked by a small black spot. An experimental learning session starts from a corner of the corridor and continues by driving the ALV through a left-turn corner of the corridor. Figure 20, shows the trace of a learning session, and each block dot in the figure represents a vehicle location.

In the later stage of a learning session, a path can be produced by the path generation method. One learned model is shown in Figure 20 and the result of path generation is shown in Figure 15. The trace of a navigation session is shown in Figure 21..

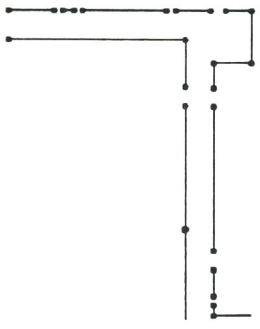


Figure 19: The positions of environment features in the experimental environment.

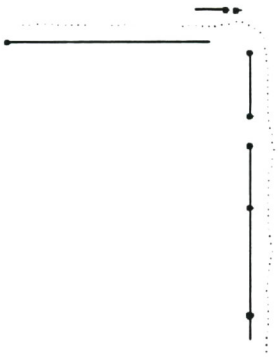


Figure 20: The original learned model

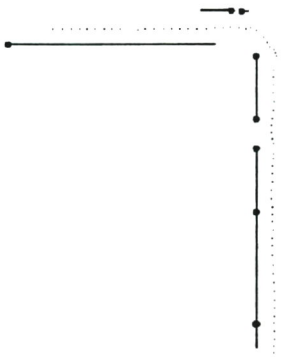


Figure 21: The trace of a navigation session.

7. Conclusions

An integrated approach have been proposed for ALV learning and ALV navigation in indoor environments by computer vision techniques. This approach has been implemented on a prototype ALV and satisfactory results have been obtained. A lot of successful navigation sessions in real time confirm the effectiveness of this approach. The main contributions of this study are as follows. First, a computer vision approach is proposed to locate an ALV by the use of visible baseline segments and corners. We use the location of an ALV and several coordinate transformations to acquire the environment features that the ALV pass through. Second, a technique of path generation has been used to generate a feasible path for guiding the ALV in indoor environments automatically. Third, in order to meet the varying speeds of the ALV due to the varying weights of the on-board loading persons, we use an odometer to measure the actual distance in a process cycle. Forth, smooth and safe navigation has been achieved by the use of the strategies for ALV turn wheel angle adjustment. Finally, the proposed approach is more convenient and flexible to users because no human involvement is needed in the measurement of the environment in which the ALV will navigate. Lots of successful navigation experiments confirm the effectiveness of the proposed approach.

References

[1] R. J. Clark, R. C. Arkin, A. Ram, ``Learning momentum: on-line performance enhancement for reactive systems,'' *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, France, May 1992.

[2] Ronald. C. Arkin, ``Motor schema-based mobile robot navigation,'' *International Journal of Robotics Research*, Vol. 8, No. 4, pp. 92--112, August 1989.

[3] C. Thrope, M. H. Hebert, T. Kanade, and S. Shafer, ``Vision and navigation for {Carnegie-Mellon NAVLAB},'' *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 362--373, May 1988.

[4] Y. Goto and A. Stentz, ``The CMU system for mobil robot navigation,'' *Proc. IEEE Int. Conf. on Robotics and Automation*, Releigh, N. C., U.S.A., pp. 99--105, 1987.

[5] M. S. Chang, P. Y. Ku., L. L. Wang, and W. H. Tsai, ``Indoor autonomous land vehicle guidance by baseline following using computer vision techniques,'' *Proceedings of Workshop on Computer Vision, Graphics and Image Processing*, Taipei, Taiwan, Republic of China, pp. 111--119, Aug. 1989.

[6] P. Y. Ku and W. H. Tsai, ``Model-based guidance of autonomous land vehicle for indoor navigation,'' *Proceedings of Workshop on Computer Vision, Graphics and Image Processing*, Taipei, Taiwan, Republic of China, pp. 165--174, Aug. 1989.

[7] W. J. Ke and W. H. Tsai, ``Indoor autonomous land vehicle guidance by corner tracking using computer vision,'' *Proceedings of Workshop on Computer Vision, Graphics and Image Processing*, Tainan, Taiwan, Republic of China, pp. 133--139, Aug. 1991.

[8] S. D. Cheng and W. H. Tsai, ``Model-based guidance of autonomous land vehicles in indoor environments by structured light using vertical line information,'' in Proceedings of Workshop on Computer Vision, Graphics and Image Processing, Tainan, Taiwan, Republic of China, pp. 340--345, Aug. 1991.

[9] Y. M. Su and W. H. Tsai, "Autonomous land vehicle guidance for navigation in building corridors and elevators by computer vision, radio, and photoelectric Sensing Techniques," Proc. of 1992 Conf. on Computer Vision, Graphics, and Image Processing, Nantou, Taiwan, R.O.C., 1992.

[10] L. L. Wang and W. H. Tsai, ``Car safety driving aided by 3-d image analysis techniques,'' Tech. Rep. TR-MIST-86-004, MIST Technical Report, National Chiao Tung University, Hsinchu, Taiwan, ROC., 1986.