Construction of a Machine Guide Dog Using a Two-mirror Omni-camera and an Autonomous Vehicle[†]

Chih-Wei Huang[1] and Wen-Hsiang Tsai[1, 2, *]

[1] Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan
sakuraai1986.cs99g@nctu.edu.tw
[2] Department of Information Communication, Asia University, Taiwan
whtsai@cis.nctu.edu.tw

Abstract. A system for use as a machine guide dog composed of an autonomous vehicle and a two-mirror omni-camera for navigations on sidewalks to guide blind people is proposed. Methods for extracting 3D information from acquired omni-images to localize the vehicle using landmarks of curb lines, tree trunks, stop lines on roads, lawn corners, signboards, and traffic cones are proposed. The methods are based on a space-mapping scheme and three new space line detection techniques. Each space line detection technique can be applied directly on omni-images to compute the 3D locations of a specific type of space line in the landmark shapes. Good experimental results show the feasibility of the proposed system.

Keywords: machine guide dog, autonomous vehicle, landmark detection, vehicle localization, omni-images.

1 Introduction

There are millions of blind people in the world. Some of them use blind canes to walk on the road. However, blind canes can only detect obstacles at short distances. A better choice is to use guide dogs as shown in Fig. 1(a). However, guide dogs are very few; e.g., there are about 60,000 blind people but just about 30 guide dogs in Taiwan in 2012. So it is of great advantages if *machine guide dogs* can be designed for use by the blind. To implement a machine guide dog, one way is to use a vision-based autonomous vehicle which can navigate automatically in outdoor environments and keep watch over the camera's field of view (FOV) to avoid collisions with obstacles. In this study, we use a specially-designed omni-camera with two mirrors as the vision system on an autonomous vehicle for this purpose.

Localization is a critical issue in implementing a navigation system, by which a vehicle can move on correct paths. Willis and Helal [1] provided a navigation system for the blind using the RFID technology to identify building and room locations. Chen and Tsai [2] proposed an indoor autonomous vehicle system using ultrasonic sensors. In outdoor spaces, the GPS can be used as a localization system for the vehicle [3]. Also, Atiya and Hager [4] proposed a vision-based system which can localize a mobile robot in real time. To enhance localization accuracy, Lui and Jarvis [5] constructed an outdoor robot with a GPU-based omni-vision system possessing an automatic baseline selection capability. To detect landmarks in environments, Fu et al. [6] proposed a navigation

[†] This work was supported financially by the Ministry of Economic Affairs under Project No. MOEA 100-EC-17-A-02-S1-032 in Technology Development Program for Academia.

[*] To whom all correspondence should be sent.

system with embedded omni-vision for multi-object tracking. A vehicle which achieves self-localization by matching omni- images was proposed by Ishizuka et al. [7].

(a) (b) (c)

Fig. 1. Guide dogs and system configuration. (a) A guide dog. (b) View of system. (b) Architecture of camera system (upright).

The goal of this study is to navigate a vision-based machine guide dog on outdoor sidewalks automatically. To achieve this goal, the major task is *vehicle localization*. The strategy for vehicle localization proposed in this study is to detect as many types of landmarks as possible along paths on sidewalks. The operation of the proposed system includes two stages: *learning* and *navigation*. The learning stage includes primarily the task of training the vehicle to acquire the along-path information useful for later vehicle guidance in the navigation stage. A scheme for training a vehicle for outdoor navigation along sidewalks is proposed first. Then, new space line detection and localization techniques based on the space-mapping method [9] are proposed next. These techniques then are applied to detect natural landmarks of lawn corner and tree trunks as well as artificial landmarks of signboards, stop lines on roads, and traffic cones for vehicle localization. Also proposed are methods for dynamically adjusting the vehicle guidance scheme to overcome varying outdoor lighting conditions.

In the remainder of this paper, the proposed path learning process and navigation strategies will be presented in Sections 3 and 4, respectively. The proposed new space line detection techniques and their applications for landmark detection and localization will be described in Sections 4 and 5, respectively. Finally, some experimental results and conclusions will be given in Section 6.

2 Learning Stage

The purpose of the proposed learning process is to create a path consisting of nodes on a sidewalk to be visited by the vehicle toward a destination. At first, some landmarks are selected for vehicle localization. Then, the used camera system is calibrated. At last, some parameter information of each landmark is extracted and recorded in the path.

2.1 Learning of Selected Landmarks

When the vehicle navigates for a time period, mechanic errors will accumulate to cause imprecise odometer readings of the vehicle location and orientation. To solve this problem, Chou and Tsai [8] proposed methods for detecting light poles and hydrants as landmarks to localize the vehicle. In this study, we select additionally two types of *natural* landmarks, *tree trunk* and *lawn corner*, and three types of *artificial* landmarks, *signboard*, *traffic cone*, and *stop line on the road*. With these additional types of landmarks, more information along the path can be used for vehicle localization, and so the vehicle can be guided more reliably to the destination. We “learn” these landmarks

by driving the vehicle to get close to each of them and recording the *vehicle direction* with respect to the nearby curb line as well as the *vehicle location* with respect to the landmark.

2.2 Construction of Pano-mapping Table as Camera Calibration

The used camera system as illustrated in Figs. 1(b) and 1(c) consists of a perspective camera, a lens, and two reflective mirrors of different sizes, all integrated into a single structure. We call the big mirror *Mirror B*, and the small *Mirror S*, respectively. The camera system is slanted for an angle of γ to enlarge the imaged frontal scene portion.

To "calibrate" the camera system, we use the space-mapping method proposed by Jeng and Tsai [9] by creating a *pano-mapping table* to record the relations between the locations of image points and those of the corresponding world-space points. A light ray is assumed to go through each world-space point P with an elevation angle α and an azimuth angle θ, be reflected by the mirror of the camera system, and be projected onto the omni-image plane as a point p at coordinates (u, v), as illustrated in Fig. 2. The pano-mapping table, like the one shown in Table 1, specifies the relation between the coordinates (u, v) of the image point p and the azimuth-elevation angle pair (θ, α) of the corresponding world-space point P.

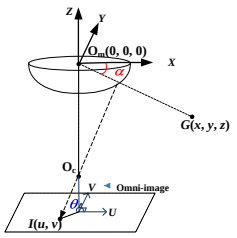


Fig. 2. Omni-imaging principle.

Table 1. A pano-mapping table used for the omni-camera.

	θ_1	θ_2	θ_3	…	θ_S
β_1	(u_{11}, v_{11})	(u_{21}, v_{21})	(u_{31}, v_{31})	…	(u_{S1}, v_{S1})
β_2	(u_{12}, v_{12})	(u_{22}, v_{22})	(u_{32}, v_{32})	…	(u_{S2}, v_{S2})
β_3	(u_{13}, v_{13})	(u_{23}, v_{23})	(u_{33}, v_{33})	…	(u_{S3}, v_{S3})
β_4	(u_{14}, v_{14})	(u_{24}, v_{24})	(u_{34}, v_{34})	…	(u_{S4}, v_{S4})
...	…	…	…	…	…
β_T	(u_{1T}, v_{1T})	(u_{2T}, v_{2T})	(u_{3T}, v_{3T})	…	(u_{ST}, v_{ST})

3 Navigation Stage

After learning the navigation path like that shown in Fig. 3, we use it to guide the vehicle. The path includes a series of nodes, through which the vehicle can move to the destination. More details are described in the following sections.

3.1 Navigation Strategy Adopted in This Study

The vehicle presumably could localize itself by the on-board odometer readings to conduct *node-based navigation*; that is, the three readings (P_x, P_y, P_θ) provided by the odometer might be used as the *vehicle pose* to identify the vehicle position and orientation in a global coordinate system (GCS) at each path node, and so can be utilized to navigate the vehicle correctly on the path. However, these odometer values are in general imprecise because of the mechanic errors accumulated during the navigation. Therefore, a *vehicle localization* process should be conducted at each node. The proposed strategy for this purpose includes two major steps: (1) adjust the erroneous vehicle orientation by the use of the detected curb line orientation; and (2) correct the vehicle position by the use of the estimated pre-selected landmark location. Note that the GCS is defined on the sidewalk for each navigation session with the start vehicle position as the origin, and the forward moving direction of the vehicle as the vertical axis.

3.2 Vehicle Localization by Selected Landmarks

To reduce the influence of accumulated mechanic errors, we conduct vehicle localization by the use of curb lines and landmarks as mentioned. Specifically, when the vehicle arrives at a node with inaccurate odometer readings (P_x'', P_y'', P_θ''), according to the first step of the above-mentioned vehicle localization strategy, we detect the straight curb line in the acquired omni-image, and compute its orientation θ' with respect to the moving direction of the vehicle [8], as illustrated in Fig. 4. Comparing θ' with the recorded curb line orientation θ in the learned path data, we compute the deviation of the vehicle orientation as $\Delta\theta = \theta' - \theta$ and adjust the vehicle orientation for the amount of $\Delta\theta$ to obtain a *calibrated* vehicle orientation P_θ' computed as $P_\theta' = P_\theta'' - \Delta\theta$.

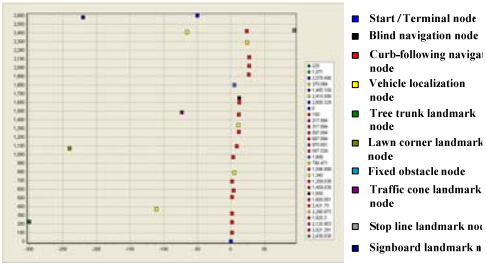


Fig. 3. Illustration of learned navigation path.

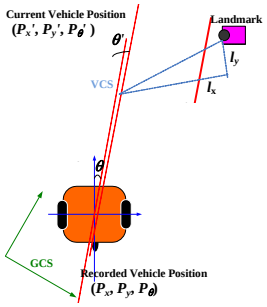


Fig. 4. Computing current vehicle location.

Afterward, we start to detect the landmark of the current node, if recorded, and obtain its position (l_x. l_y) with respect to the vehicle coordinate system (VCS) (the details described later). According to the learned landmark position (L_x, L_y) in the GCS recorded in the path data and the calibrated vehicle orientation P_θ', we compute the current vehicle position (P_x', P_y') by the following equations (see Fig. 4 for an illustration):

$$\begin{bmatrix} P_x' \\ P_y' \end{bmatrix} = \begin{bmatrix} L_x \\ L_y \end{bmatrix} + \begin{bmatrix} \cos P_\theta' & \sin P_\theta' \\ -\sin P_\theta' & \cos P_\theta' \end{bmatrix} \begin{bmatrix} l_x \\ l_y \end{bmatrix}. \quad (1)$$

Finally, we replace the odometer readings, (P_x'', P_y'', P_θ''), with the above corrected vehicle pose parameters (P_x', P_y', P_θ') and navigate the vehicle forward to the next node.

4 Natural Landmark Detection for Vehicle Localization

It is found in this study that the uses of space lines are sufficient to localize many types of landmarks. However, compared with the result of using images acquired by the traditional projective camera, the projection of a space line onto an *omni-image* taken by an omni-camera is not a line but a *conic-section curve* [10]. Wu and Tsai [10] proposed a method for detecting *directly* such curves in omni-images of an H-shaped landmark used in automatic helicopter landing. In this study, we propose instead new space line detection techniques based on the space-mapping method [9] using *pano-mapping tables*. An essence of the proposed techniques is to utilize the *space plane* which goes through the space line and the center of the mirror of the omni-camera, instead of trying to obtain directly the conic section curve in the omni-image. More details are described next.

4.1 Line Detection Using Pano-Mapping Table

Assume that a pano-mapping table has been set up for the omni-camera, and that a space line L to be detected is projected by *Mirror B* onto the omni-image with G being a point on L. A light ray going through G is projected by *Mirror B* onto the omni-image to become an image point I as shown in Fig. 5. The mirror center O_B and G together form a vector $V_G' = [G_x', G_y', G_z']^T$ where T means "transpose." The components of V_G' can be described in terms of the azimuth and elevation angles θ and α of the light ray as: $G_x' = \cos\alpha\times\cos\theta$, $G_y' = \cos\alpha\times\sin\theta$, $G_z' = \sin\alpha$. Also, as mentioned previously, to increase the frontal FOV, we have slanted the camera system up for the angle of γ. So, there is a transformation between the coordinates (X', Y', Z') of the original camera coordinate system (CCS) and the new coordinates (X, Y, Z) of the slanted CCS, described by:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\gamma) & -\sin(-\gamma) \\ 0 & \sin(-\gamma) & \cos(-\gamma) \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{2}$$

so that $V_G' = [G_x', G_y', G_z']^T$, after being slanted with the above transformation, becomes $V_G = [G_x, G_y, G_z]$ with $G_x = \cos\alpha\times\cos\theta$, $G_y = \cos\alpha\times\sin\theta\times\cos\gamma$ and $G_z = -\cos\alpha\times\sin\alpha + \sin\alpha\times\cos\gamma$. Then, as shown in Fig. 6, let I_L be the conic section curve resulting from projecting the space line L onto the omni-image, and Q be the space plane going through L and the mirror center O_B. Also assume that the coordinates (X, Y, Z) describe a point on Q, and that $N_Q = (l, m, n)$ describe the normal vector of Q. Because N_Q and V_G are perpendicular to each other, we have:

$$N_Q \cdot V_G = (l, m, n)\cdot(G_x, G_y, G_z) = l\times G_x + m\times G_y + n\times G_z = 0. \tag{3}$$

where "$\cdot$" denotes the inner product operation. By Eq. (2), we can rewrite Eq. (3) as:

$$l + m\times\frac{(\cos\alpha\times\sin\theta\times\cos\gamma + \sin\alpha\times\sin\gamma)}{(\cos\alpha\times\cos\theta)} + n\times\frac{(-\cos\alpha\times\sin\theta + \sin\alpha\times\cos\gamma)}{(\cos\alpha\times\cos\theta)} = 0. \tag{4}$$

However, Eq. (4) consists of three unknown parameters l, m, and n which describe the normal vector of Q. Assuming $n \neq 0$, we may divide Eq. (4) by n to get:

$$B + Aa_0 + a_1 = 0 \tag{5}$$

where

$$A = m/n;\ B = l/n;$$

$$a_0 = \frac{(\cos\alpha\times\sin\theta\times\cos\gamma + \sin\alpha\times\sin\gamma)}{(\cos\alpha\times\cos\theta)};\quad a_1 = \frac{(-\cos\alpha\times\sin\theta + \sin\alpha\times\cos\gamma)}{(\cos\alpha\times\cos\theta)}. \tag{6}$$

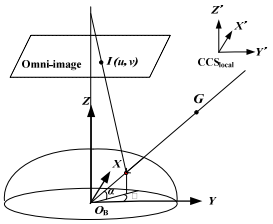


Fig. 5. A space point with elevation α & azimuth θ.

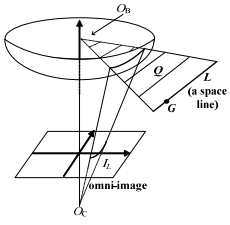


Fig. 6. A space line projected in omni-image.

Chou and Tsai [8] proposed a method to localize a vertical line with respect to the vehicle (actually, with respect to the CCS on the vehicle). The vertical line, called the L_Y *line* hereafter, is parallel to the *Y*-axis line in the GCS. See Fig. 7 for an illustration. In this study, we extend their method to localize two more specific lines: one parallel to the *X*-axis, called the L_X *line*; and the other parallel to the *Z*-axis, called the L_Z *line*. We now derive equations for use to localize these three types of lines.

The direction vector of L_Y is $D_Y = [0, 1, 0]^T$. Therefore, Eq. (3) leads to $0\times l + 1\times m + 0 \times n = 0$, ;or equivalently, $m = 0$, and so Eq. (5) can be reduced to be:

$$B = -a_1. \tag{7}$$

For our cases here, the direction vector of the L_X line is $D_X = [1, 0, 0]^T$. So, Eq. (3) leads to $1\times l + 0\times m + 0\times n = 0$, or equivalently, $l = 0$, and so Eq. (5) can be reduced to be:

$$A = -a_1/a_0. \tag{8}$$

About L_Z line detection, the direction vector of the L_Z line is $D_Z = [0, 0, 1]^T$. So, Eq. (3) leads to $0\times l + 0\times m + 1\times n = 0$, or equivalently, $n = 0$, and so Eq. (4) can be reduced to be:

$$l + m \times a_0 = 0. \tag{9}$$

or equivalently,

$$-a_0 = K \tag{10}$$

where

$$K = l/m. \tag{11}$$

For each case above, to detect the L_X, L_Y, or L_Z line in the omni-image M directly with the tilt angle γ of the camera system known in advance, the following steps are performed: 1) apply binarization and edge detection operations to M to obtain edge points in M; 2) set up a 1D Hough space H of the parameter A, B, or K; 3) for each edge point p with coordinates (u, v), look up the pano-mapping table of the omni-camera to obtain the azimuth-elevation angle pair (θ, α) of the world-space point P corresponding to p; 4) compute a_0 and a_1 according to Eqs. (6); 5) for each cell c in H with value A, B, or K, if Eq. (8), (7), or (10) is satisfied, then increment the count of c by one; 6) find the maximum cell count in H with value A_{max}, B_{max}, or K_{max} which, according to Eqs. (6) and (11), are equal to m/n, l/n, or l/m, respectively.

Note that in the above algorithm, we do not *really* detect the line L_X, L_Y, or L_Z (in the form of a conic section) but just its related parameter A, B, or K described by Eq. (8), (7), or (11), respectively.

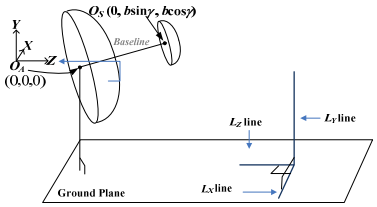


Fig.7. Three specific space lines.

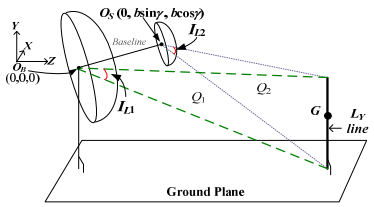


Fig. 8. An L_Y line projected onto two mirrors.

4.2 3D Data Computation Using Detected Space Lines

Based on the detected parameters of the three types of space lines described above, we can derive the 3D locations of each type of space line, as described subsequently.

(A) 3D data computation using an L_Y line

As shown in Fig. 8, an L_Y line is projected by *Mirrors B* and *S* onto the image plane to form lines I_{L1} and I_{L2}, respectively. The center O_B of *Mirror B* is assumed to be located at coordinates (0, 0, 0) in the CCS. The position of the center O_S of *Mirror S* may be described in terms of the slant angle γ of the camera system and the *baseline* value b between the two mirrors as (0, $b\sin\gamma$, $b\cos\gamma$) in the CCS. Let the two space planes going through L_Y and the centers of the two mirrors, O_B and O_S, respectively, be denoted as Q_1 and Q_2. Also, let G be a point on L_Y with coordinates (X, Y, Z), and the normal vector of Q_1 be described by $N_1 = [l_1, m_1, n_1]^T$. Denote the vector from the origin O_B at coordinates (0, 0, 0) to G at coordinates (X, Y, Z) as V_{G1} which is just $V_{G1} = [X, Y, Z]^T$. Then, since N_B and V_{G1} are perpendicular, we have $V_{G1} \cdot N_1 = 0$, leading the following equation:

$$l_1X + m_1Y + n_1Z = 0. \tag{12}$$

Furthermore, we know that the mirror center O_S is at coordinates (0, $b\sin\gamma$, $b\cos\gamma$). Also, suppose that the normal vector of Q_2 is denoted by $N_2 = [l_2, m_2, n_2]^T$. Since point G is on the L_Y line, it is also on Q_2, meaning that the vector V_{G2} from O_S to G is just $V_{G2} = [X, Y - b\sin\gamma, Z - b\cos\gamma]^T$. So, by a similar reasoning using the fact $V_{G2} \cdot N_2 = 0$, we get:

$$l_2X + m_2(Y - b\sin\gamma) + n_2(Z - b\cos\gamma) = 0. \tag{13}$$

Since the direction vector of L_Y is $D_Y = [0, 1, 0]^T$, meaning that $m_1 = m_2 = 0$, the above two space planes described by (12) and (13) can be reduced to be

$$B_1X + Z = 0;\ B_2X + (Z - b\cos\gamma) = 0, \tag{14}$$

with $B_1 = l_1/n_1$ and $B_2 = l_2/n_2$ which can be obtained by the 1D Hough transform process mentioned previously. Solving (14), we can obtain the following desired solutions for X and Z to specify the location of a point P on the L_Y line, or simply, just that of L_Y on the X-Z plane in the CCS (or equivalently, on the ground):

$$X = b\cos\gamma/(B_2 - B_1);\ Z = -B_1b\cos\gamma/(B_2 - B_1). \tag{15}$$

(B) 3D data computation using an L_X line

As shown in Fig. 9, the process for 3D computation using an L_X line is similar to the case of using an L_Y line. The two planes Q_3 and Q_4 may be described by:

$$l_3X + m_3Y + n_3Z = 0;\ l_4X + m_4(Y - b\sin\gamma) + n_4(Z - b\cos\gamma) = 0 \tag{16}$$

where $N_3 = [l_3, m_3, n_3]^T$ and $N_4 = [l_4, m_4, n_4]^T$ represent the normal vectors of Q_3 and Q_4, respectively. Also, the direction vector of the L_X line is $D_X = [1, 0, 0]^T$ so that $l_3 = l_4 = 0$, and so Eqs. (16) can be reduced to be

$$A_1Y + Z = 0;\ A_2(Y - b\sin\gamma) + (Z - b\cos\gamma) = 0 \tag{17}$$

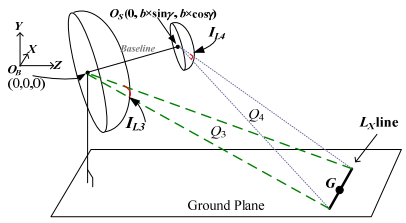


Fig. 9. An L_X line projected onto two mirrors.

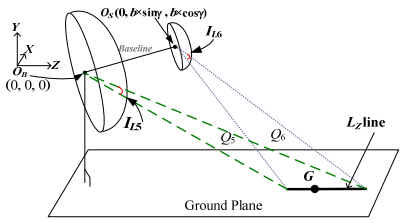


Fig. 10. An L_Z line projected onto two mirrors.

where $A_1 = m_3/n_3$ and $A_2 = m_4/n_4$ which can be obtained by the 1D Hough transform process mentioned previously. Eqs. (17) may be solved to get the values of Y and Z as follows to specify the location of a point G on the L_X line in the Y-Z plane of the CCS:

$$Y = (A_2 b\sin\gamma + b\cos\gamma)/(A_2 - A_1);\ Z = -A_1(A_2 b\sin\gamma + b\cos\gamma)/(A_2 - A_1). \quad (18)$$

(C) 3D data computation using an LZ line

Similarly, as shown in Fig. 10, Q_5 and Q_6 may be described by:

$$l_5X + m_5Y + n_5Z = 0;\ l_6X + m_6(Y - b\sin\gamma) + n_6(Z - b\cos\gamma) = 0 \quad (19)$$

where $N_5 = [l_5, m_5, n_5]^T$ and $N_6 = [l_6, m_6, n_6]^T$ represent the normal vectors of Q_5 and Q_6, respectively. Eqs. (19) are equivalent to

$$K_1X + Y = 0;\ K_2X + (Y - b\sin\gamma) = 0, \quad (20)$$

where $K_1 = l_5/m_5$ and $K_2 = l_6/m_6$. Also, the direction vector of the L_Z line is $D_Z = [0, 0, 1]^T$, meaning that $n_5 = n_6 = 0$, and so Eqs. (2) can be solved to get the values of X and Y as follows to specify the location of a point G on the L_Z line on the X-Y plane in the CCS:

$$X = (b\times\sin\gamma)/(K_2 - K_1);\ Y = -(K_1\times b\times\sin\gamma)/(K_1 - K_2). \quad (21)$$

4.3 Tree Trunk Detection and Localization

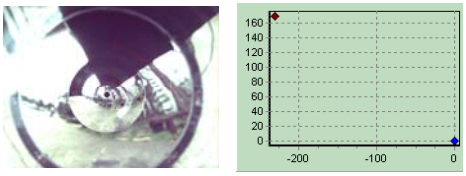
Before we can apply the formulas derived previously to conduct localization of a tree trunk, we have to solve the varying lighting problem occurring during image acquisition, which often causes failures of tree trunk detection. For this, we binarize the input image by moment-preserving thresholding [11] and conduct image segmentation to obtain a group G of candidate feature points. To ensure goodness of the result, we apply principal component analysis (PCA) to G. Specifically, as shown in Fig. 11 we compute the following data: the center C of the points in G; the height h of C; the eigenvalue pair (λ_1, λ_1) of the covariance matrix of the points in G; the eigenvectors $e_1 = [u_1, v_1]^T$ and $e_2 = [u_2, v_2]^T$ corresponding to λ_1 and λ_2, respectively; the length ratio η of G in terms of the two eigenvalues: $\eta = \lambda_1/\lambda_2$; and the orientation ω of G: $\omega = \tan^{-1}(v_1/u_1)$. Then, we utilize the three parameters h, ω, and η to describe the shape of the tree trunk, and check the correctness of the extracted tree trunk points by matching these computed parameter values against those "learned" in the learning phase and recorded in the path data — if the resulting match measure is within a preset tolerance, then the extracted feature point group G are regarded correct and used for vehicle localization; otherwise, the vehicle is moved a little bit around to conduct a repetition of the above process.

To localize a tree trunk, we regard the tree trunk axis as a vertical space line and apply the previously-proposed L_Y line localization technique to compute its location with respect to the vehicle. Specifically, we extract the centers C_1 and C_2 of the two groups G_1 and G_2 of tree trunk feature points appearing in the image regions of *Mirrors S* and *B*, respectively, in the input omni-image, and regard them as projections of a single space point G on the L_Y line, as mentioned in Section 4.1. More detailed steps include: 1) compute the coordinates (u_1, v_1) and (u_2, v_2) of centers C_1 and C_2, respectively; 2) use (u_1, v_1) and (u_2, v_2) to look up the pano-mapping table to get the respective elevation-azimuth angle pairs (α_1, θ_1) and (α_2, θ_2) of C_1 and C_2; 3) use Eqs. (6) and (7) to compute the two parameters B_1 and B_2; 4) use Eqs. (15) to compute the location of the tree trunk axis

described by X and Z with respect to the vehicle. An example of tree trunk detection and corresponding vehicle localization results is shown in Fig. 12.

(a) (b)

Fig.11. Principal component analysis for tree trunk detection. (a) Principal components, e_1 and e_2. (b) Orientation ω.



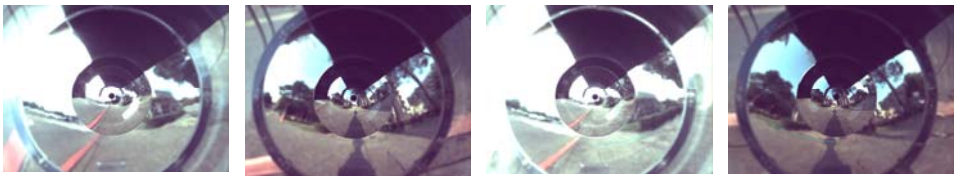
(a) (b)

Fig. 12. Tree truck detection and localization. (a) Extracted tree axis. (b) Computed tree location (red spot) with respect to vehicle.

4.4 Lawn Corner Detection and Localization

Because the lawn corner is too obscure to be recognized in the omni-image, the proposed lawn corner detection process is divided into two stages. When the vehicle arrives at a proper position for the detection work, a space line forming one side of the corner, appearing as a horizontal space line L_1 on the ground, is detected and localized firstly. Then, the vehicle is guided to turn left, and the other side of the corner, appearing to be another horizontal line L_2 perpendicular to L_1, is detected and localized as well. The two space lines L_1 and L_2 then are drawn in the input omni-image to cross each other to form a corner. Finally, we compute the 3D data of the corner as the localization result.

The details of this process for estimating the coordinates (X_g, Y_g, Z_g) of the lawn corner in the GCS (global coordinate system) include: 1) acquire an omni-image I_1 of the lawn; 2) apply image thresholding and edge detection operations to I_1 to obtain an edge point image I_1' of the lawn boundaries; 2) apply the previously-described 1D Hough transform to I_1' to detect a side boundary L_1 of the lawn as an L_X line and compute its corresponding parameters A_1 and A_2; 3) use Eqs. (18) to compute the values of Y and Z as $Y = -h_1$ and $Z = d_1$ which specify the height of the lawn with respect to the CCS and the distance of the lawn boundary L_1 to the vehicle, respectively; 4) turn the vehicle for an angle of 90° and conduct the above steps to obtain two other values of Y and Z as $Y = -h_2$ and $Z = d_2$ for the other lawn boundary L_2; 5) assign values to the coordinates (X_g, Y_g, Z_g) of the lawn corner by $X_g = d_2$, $Y_g = -(h_1 + h_2)/2$, and $Z_g = d_1$. An example of the results of lawn detection and localization using the above process is shown in Fig.13.



(a) (b) (c) (d)

Fig. 13. Results of lawn corner detection and localization. (a) and (b) Lawn images taken in two perpendicular directions. (c) and (d) Lawn boundary detection results.

5 Artificial Landmark Detection for Vehicle Localization

The artificial landmarks we use in this study include curb line, signboard, stop line on roads, and traffic cones. Their uses for vehicle localization are described now.

5.1 Proposed Technique for Curb Line Following

To conduct vehicle navigations on sidewalks, Chou and Tsai [8] proposed a technique to localize curb lines with respect to the vehicle. In this study, we propose a new method to localize the curb line by the use of the projection of the curb line onto the image region of *Mirror B* in the omni-image. More specifically, we get the feature points of the curb line using its color information. Then, we detect the two boundary lines of the curb which has a certain width. In the resulting edge-point image, we use the previously-proposed L_Z line detection method to find the two boundary lines. Then, we choose the inner boundary line of the curb to compute its location with respect to the vehicle. By the parameter K_1 obtained by the previously-described 1D Hough transform and the height h of the center of *Mirror B*, we can know from Eqs. (20) that $X = -Y/K_1$. Also, it is obvious that $Y = -h$, so we can get the following data of the curb line:

$$X = h/K_1;\ Y = -h. \tag{22}$$

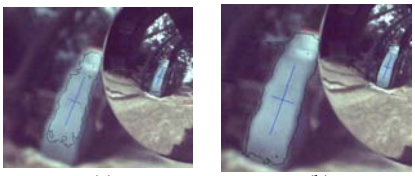
An example of the experimental results of curb detection and corresponding vehicle location estimation using the proposed localization method are shown in Fig. 14.

(a) (b)

Fig. 14. An example of curb line detection and localization results. (a) Curb line segmentation result. (b) Computed curb line location (yellow spot) with respect to vehicle position (blue spot).

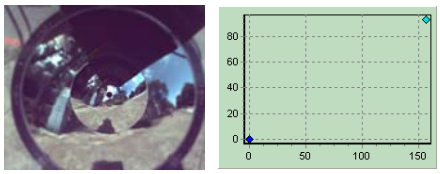
5.2 Signboard Detection and Localization

The idea of the proposed signboard detection method is to extract the signboard contour and apply the same technique as that used for tree trunk detection described previously. Due to the obvious signboard color, we use the HSI color model to extract the signboard shape from an image. Besides, varying lighting conditions often influence the hue and saturation features of the HSI colors of the landmark. Based on learned signboard contour information, a dynamic color thresholding scheme is proposed in this study to adjust the saturation threshold value S_{th} to be within a fixed range $[S_0, S_1]$ for the purpose of guaranteeing consistent signboard contour segmentation, where S_0 and S_1 are learned in advance in different lighting conditions in the learning stage. An experimental result of signboard segmentation by dynamic thresholding is shown in Fig. 15.



(a) (b)

Fig. 15. Results of signboard segmentation. (a) Result using fixed threshold. (b) Result using proposed dynamic thresholding.



(a) (b)

Fig. 16. Signboard localization. (a) Result of extracting L_Y line of signboard (b) Computed signboard position (green spot).

After the signboard is segmented successfully, by regarding the vertical signboard axis as an L_Y line perpendicular to the ground, we apply the method we use for detecting

and localizing the tree trunk described previously in Sec. 4 to compute the position of the signboard axis for vehicle localization. The details are omitted due to the page limit. An example of the results of such vehicle localization using a signboard is shown in Fig. 16.

5.3 Stop Line Detection and Localization

Besides landmarks on the *sidewalk*, we may also use those on the *road* for vehicle localization. One of the landmarks commonly seen on the road is the *stop line* which is used in this study as well for vehicle localization. Because the stop line on the road has obvious color information (mostly white), we also utilize the HSI color model to extract it. Then, we detect its boundaries in the input omni-image. Finally, we detect two parallel L_X lines and one perpendicular L_Z line in the edge-point image as illustrated in Fig. 17(a), using the previously-proposed 1D Hough transform, to obtain the parameter information of the entire boundary shape for use in vehicle localization. An experimental result is shown in Figs. 17(b) and 17(c).

(a) (b) (c)

Fig. 17. Result of stop line detection and localization. (a) Result of extracting stop line boundaries as L_X and L_Z lines. (b) Computed positions of stop line (yellow and red spots).

5.4 Traffic Cone Detection and Localization

When engineering works are conducted on sidewalks, the workers usually put traffic cones near the working area to warn people. For this type of situation, we propose to detect traffic cones and use them as landmarks. The proposed method for traffic cone detection is similar to that for stop line detection. But here we detect one L_Z line and one L_X line to carry out the detection of the traffic cone base which is of the shape of a square. After detecting the boundary lines of the traffic cone base, we apply the previously-mentioned 1D Hough transform to compute two parameters, A_1 and K_1, for use in computing its location. Besides, we also utilize traffic cone corner to compute a L_Y line by which we can draw a vertical line to illustrate the position of the traffic cone in the omni-image. An experimental result of detecting the traffic cone using the proposed method is given in Fig. 18.

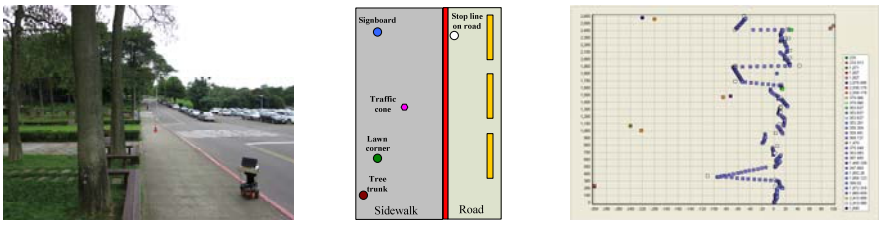
(a) (b)

Fig. 18. Result of traffic cone localization. (a) Result of extracting L_X, L_Y, and L_Z lines of traffic cone (b) Computed position of traffic cone (yellow spot) with respect to vehicle.

6 Experimental Results

The experimental environment was a sidewalk in National Chiao Tung University as shown in Figs. 19(a) and 19(b). In each navigation session, the vehicle started from an identical spot on the sidewalk just like in the learning process and navigated along the recorded navigation path nodes mainly by the curb line following technique. Then, the vehicle detected the pre-learned landmarks and localized their positions to adjust its odometer readings at each visited path node until reaching the appointed terminal node. Many successful navigation sessions have been conducted. A path map with recorded vehicle positions at the visited nodes of one navigation session is shown in Fig. 19(c).

Furthermore, we have tested the precision of vehicle localization in the experiments. We computed the errors between the actual positions of the landmarks measured manually and the positions of the landmarks computed by the proposed localization techniques for eight navigation sessions. The average error percentage of the estimated landmark position is 7.52% of an average landmark distance of about 200cm, which shows that the precision of the proposed system is satisfactory for real applications, considering the width of the sidewalk which is about 400 cm.



(a) (b) (c)

Fig. 19. Experimental environment and a path map. (a) A view of environment. (b) An illustration of environment. (c) A path map resulting from a navigation session.

7 Conclusions

Construction of a machine guide dog using a two-mirror omni-camera and an autonomous vehicle has been proposed, for which several methods have been proposed: 1) by the use of a learning interface designed in this study, a trainer can guide the vehicle to navigate on a sidewalk and construct a navigation path conveniently; 2) two new space line detection techniques based on the space mapping method have been proposed; 3) several landmark detection techniques have been proposed for conducting vehicle navigation; and 4) to conduct the landmark detection works more effectively in the outdoor environment, techniques for dynamic threshold adjustments have also been proposed. Good landmark detection results and successful navigation sessions on a sidewalks show the feasibility of the proposed methods. Future researches may be directed to detecting pedestrians or bike riders; designing a camera with a smaller size; recognizing traffic signals to go through road crossings; etc.

References

1. S. Willis and S. Helal, “RFID information grid for blind navigation and wayfinding,” *Proceedings of the 9th IEEE International Symposium on Wearable Computers*, pp. 34–37, Washington, DC, USA, Oct. 2005.
2. M. F. Chen and W. H. Tsai, "Automatic learning and guidance for indoor autonomous vehicle navigation by ultrasonic signal analysis and fuzzy control techniques," *Proceedings of 2009 Workshop on Image Processing, Computer Graphics, and Multimedia Technologies*, *Nat’l Computer Symposium*, pp. 473–482, Taipei, Taiwan, Nov. 2009.

3. E. Abbot and D. Powell, "Land-vehicle navigation using GPS," *Proc. of IEEE*, vol. 87, pp. 145–162, 1999.
4. Sami Atiya and Gregory D. Hager, "Real-time vision-based robot localization," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 785–800, Dec. 1993.
5. W. Lui and R. Jarvis, "Eye-Full Tower: A GPU-based variable multibaseline omnidirectional stereovision system with automatic baseline selection for outdoor mobile robot navigation," *Robotics and Autonomous Systems*, pp. 747-761, vol. 58, no. 6, Apr. 2010.
6. H. Fu, Z. Cao, and X. Cao, "Embedded omni-vision navigator based on multi-object tracking," *Machine Vision and Applications*, pp. 349-358, vol. 22, no. 2, 2011.
7. D. Ishizuka, A. Yamashita, R. Kawanishi, T. Kaneko, and H. Asama, "Self-localizaion of mobile robot equipped with omnidirectional camera using image matching and 3D-2D edge matching," *Proceedings of IEEE Int'l Conference on Computer Vision Workshops*, Barcelona, Spain, pp. 272-279, Nov. 6-13, 2011.
8. Y. H. Chou and W. H. Tsai, "Guidance of a vision-based autonomous vehicle on sidewalks for use as a machine guide dog," *Proceeding of 2011 Conference on Computer Vision, Graphic, and Image Processing*, Chiayi, Taiwan, 2011.
9. S. W. Jeng and W. H. Tsai, "Using pano-mapping tables to unwarping of omni-images into panoramic and perspective-view Images," *Proceeding of IET Image Processing*, vol. 1, no. 2, pp. 149–155, June 2007.
10. C. J. Wu and W. H. Tsai, "An omni-vision based localization method for automatic helicopter landing assistance on standard helipads," *Proceedings of 2nd Int'l Conference on Computer and Automation Engineering*, vol. 3, pp. 327–332, Singapore, 2010.
11. W. H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377-393, 1985.