

Vision-Based Autonomous Vehicle Guidance for Indoor Security Patrolling by a SIFT-Based Vehicle-Localization Technique

Kuan-Chieh Chen and Wen-Hsiang Tsai, *Senior Member, IEEE*

Abstract—A novel method for guidance of vision-based autonomous vehicles for indoor security patrolling using scale-invariant feature transformation (SIFT) and vehicle localization techniques is proposed. Along-path objects to be monitored are used as landmarks for vehicle localization. The localization work is accomplished by three steps: SIFT-based object image feature matching, 2-D affine transformation using the Hough transform, and analytic 3-D space transformation. Object monitoring can be simultaneously achieved during the vehicle-localization process, and most planar-surfaced objects can be utilized in the process, greatly enhancing the applicability of the proposed method. Vehicle trajectory deviations from the planned path due to mechanic error accumulation are also estimated by setting up a calibration line on the monitored object image and applying the 3-D space transformation. Moreover, a path-correction technique is proposed to conduct a path adjustment and guide the vehicle to navigate to the next path node. Analysis of the accuracy of the vehicle-localization and path-correction results is finally included. The experimental results show that the proposed method, utilizing only a single view of each object, can guide the vehicle to navigate accurately and monitor objects successfully.

Index Terms—Autonomous vehicle, computer vision, guidance, landmarks, planar-surfaced objects, scale-invariant feature transformation (SIFT), security patrolling, vehicle localization, 3-D space transformation.

I. INTRODUCTION

IN RECENT years, due to fast developments of computer vision techniques, studies on vision-based autonomous vehicle navigation have high prominence because of their great potential in various applications [1]–[8]. Autonomous vehicles are becoming increasingly capable of performing a great variety of dangerous or dreary work to replace human beings

Manuscript received August 17, 2009; revised December 7, 2009 and February 18, 2010; accepted April 27, 2010. Date of publication June 7, 2010; date of current version September 17, 2010. This work was supported by the Ministry of Economic Affairs under Project MOEA 98-EC-17-A-02-S2-0047 in the Technology Development Program for Academia. The review of this paper was coordinated by Prof. S. Ci.

K.-C. Chen was with the Institute of Multimedia Engineering, College of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan. He is now with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan (e-mail: jackie.cs94g@nctu.edu.tw).

W.-H. Tsai is with the Department of Computer Science, College of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan, and also with the Department of Information Communication, Asia University, Taichung 41354, Taiwan (e-mail: whtsai@cis.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2010.2052079

in applications of interoffice document delivering, unmanned transportation, house cleaning, security patrolling, etc. The goal of this study is to develop an autonomous vehicle system for indoor security patrolling applications. The major issue in this study is to make the vehicle precisely navigate in its patrolling path. This may be accomplished by accurate *vehicle localization* in each navigation cycle.

Traditionally, an autonomous vehicle is equipped with an odometer to measure the current location of the vehicle with respect to a starting point. However, this scheme usually suffers from incremental mechanic errors that are caused by the vehicle wheel system. On the other hand, it is common to equip vehicles with visual cameras that provide more flexible visual information for navigation control. It is, thus, desired to develop an *automatic vision-based* vehicle-localization technique to overcome the mechanic error-accumulation problem. One way to achieve this goal is to utilize the features of artificial landmarks or natural scenes in the environment to locate the vehicle by feature matching. Chou and Tsai [9] proposed a method to utilize house corners to estimate vehicle locations. Okuma *et al.* [10] used a colored marker for camera pose estimation. Huang *et al.* [11] used a colored rectangular signboard to obtain the relative position of the vehicle by calculating the vanishing points in the signboard image. Wu and Tsai [12] used a circular shape that is attached on the ceiling for vehicle-location estimation using an upward-looking omni-camera equipped on the vehicle. Xu *et al.* [13] utilized parallel lines and corner points on ceilings as features for vehicle localization.

Most of the aforementioned methods can only deal with landmarks with *specific* shapes or in *ideal* backgrounds, resulting in unreasonable restrictions on the environment in which the vehicle can navigate. Another approach to vehicle localization [14]–[17], [26] is to represent vehicle locations as a *visual path* that consists of a set of reference node images and a topological graph. Each node in the graph corresponds to a vehicle location, and each between-node link represents a path segment through which the vehicle can navigate. The vehicle-localization process is performed by finding the reference node image that is most similar to the current location view and then accordingly estimating the relative vehicle position. By repeating this process, the vehicle may be guided to follow the visual path. However, such a *continuous* image-matching process acquires the along-path environmental images very frequently, even at nonnode spots, and are therefore very time consuming, resulting in slow navigation speeds, in general.

In this paper, a vision-based method for indoor security patrolling is proposed, which conducts vehicle localization using the image features of along-path fixed objects to be monitored. Such a type of object is abundant in indoor environments and, thus, can be used as a landmark for vehicle localization. An advantage of this method of *localization by along-path objects* is that vehicle-location estimation and object monitoring can be conducted *simultaneously*.

More specifically, the security patrolling process as proposed is divided into two stages: *learning* and *navigation*. In the learning stage, the vehicle is instructed to learn relevant information of a preselected path and the along-path objects to be monitored. In the navigation stage, the vehicle moves along the learned path and visits each *node* where a concerned object is located. Since the position of an object in an image of it will be just *close to*, instead of *exactly at*, the position of the object found in a reference image acquired in the learning stage, we adopt the scale-invariant feature transformation (SIFT) technique proposed by Lowe [18]–[20] for image feature extraction and matching. Accordingly, *any* planar-surfaced object can be utilized as a landmark, leading to the great improvement on the flexibility to choose a landmark.

After the image-matching step is conducted, an affine transformation between the image taken in the learning stage and that taken in the navigation stage is estimated from the matching result using the Hough transform. Then, a line called the *calibration line* is set up in the object image to compute the deviation of the vehicle trajectory from its correct path by utilizing the found affine transformation and an *analytic* 3-D transformation technique proposed in this paper. Finally, a path-correction technique is proposed to guide the vehicle to navigate more accurately to the next object to be monitored. Only single views of the monitored objects are needed for vehicle localization and object monitoring. With reasonable between-node distances, the proposed method can guide the vehicle to smoothly navigate specified paths without hitting the walls in the environment, as demonstrated by the experiments conducted in this study.

In the remainder of this paper, the problem dealt with in this paper and the advantages of the proposed method are described in Section II. The detail of the proposed analytic 3-D transformation technique is described in Section III. The proposed path-correction technique utilizing the 3-D transformation result is presented in Section IV. The proposed object image-matching scheme based on the SIFT to find the transformation between two images is given in Section V. Some experimental results are described in Section VI, followed by conclusions in Section VII.

II. IDEA OF THE PROPOSED METHOD

The vehicle used as a test bed in this study is shown in Fig. 1(a). An odometer in the vehicle provides three parameters of the vehicle location in each navigation cycle—one parameter for the vehicle orientation and the other two for the x - and y -coordinates of the vehicle position in the real-world space. A pan-tilt-zoom (PTZ) camera is equipped on the vehicle for image acquisition, as shown in Fig. 1(b).

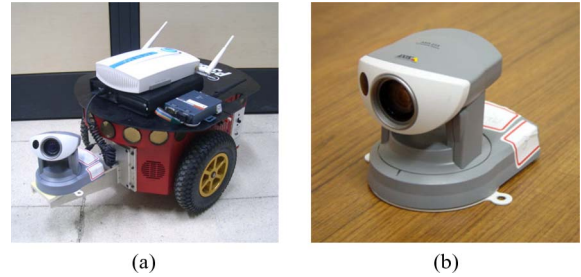


Fig. 1. Autonomous vehicle system used in this study. (a) Vehicle. (b) PTZ camera equipped on the vehicle.

A. Problem Definition and Advantages of the Proposed Method

The problem investigated in this paper is defined here. A vehicle navigates in an indoor environment along a preselected path on which a number of objects are to be monitored. These objects are utilized for vehicle localization, each with at least a *planar* surface facing the vehicle. Such objects may be posters, paintings, boxes, etc. They need not be manmade; instead, even wall planes with granite surfaces, for example, are also appropriate. The vehicle visits each object in a navigation session to monitor the existence of the object, uses the features existing in the object's surface image to find the deviation of the vehicle from the planned trajectory, and corrects its pose (including its position and orientation) before moving to the next node, where another object is to be monitored. This way, several advantages can be identified, as described in the following.

- 1) The proposed method is more *flexible* than traditional ones, which require landmarks with special shapes (circles, rectangles, etc.).
- 2) The proposed method utilizes single views of fixed object scenes for vehicle localization, which is simpler than many visual path-following techniques [14]–[17], [26] utilizing contents of the entire environmental views.
- 3) Object monitoring and vehicle localization may be *simultaneously* conducted to save time and speed up navigation.
- 4) Objects with planar surfaces are abundant in indoor environments so that the localization work can be *frequently* carried out along the navigation path to guide the vehicle more effectively.
- 5) *Frequent* vehicle localization is advantageous as an aid for *path correction* during navigation, overcoming the mechanical error-accumulation problem that is often encountered in autonomous vehicle guidance.

B. Main Idea of Vehicle Localization by Planar-Surfaced Objects

The main idea of using a planar-surfaced object for vehicle localization, as proposed in this paper, is to set up a *reference coordinate system* (RCS) on a line, which is called the *calibration line*, on the object surface with one end of the line as the origin of the system and compute the vehicle location (including its position and orientation) with respect to this coordinate system. Also, the calibration line is selected to be parallel to the environment floor and is used as an axis of the RCS.

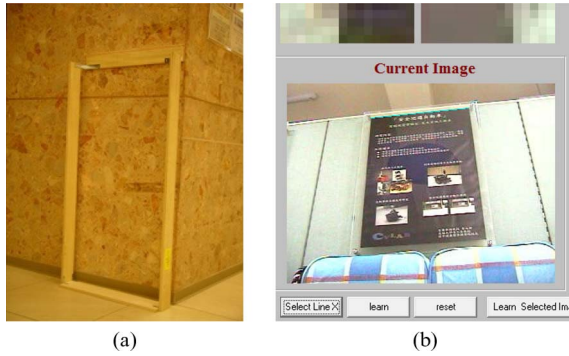


Fig. 2. Two methods for creating a calibration line. (a) Using a line stand affixed with a line structure. (b) Manually drawing a virtual line in the image (the green one on the top of the poster).

To set up the calibration line, it is noted that a manmade object often has rectangular surface shapes, on which a *real* line can be detected for this purpose. For an object that does not have line features on its planar surface, for example, a granite-surfaced wall, it is proposed in this paper to create for it a *virtual* (physically nonexistent) calibration line by two ways: 1) putting right under the object a “line stand” affixed with a line structure (e.g., a straight steel wire or a line on the surface of the stand) and then automatically detecting the line in the image or 2) *drawing* in the object image a line that is *parallel* to the environment floor using any clue of the image features. See Fig. 2 for an illustration.

With the allowance of a virtual line as an axis of the RCS, the flexibility of choosing objects for use as vehicle-localization landmarks is greatly raised. This is another advantage of the proposed method.

C. Sketch of the Proposed Navigation Process

In the navigation stage, the vehicle moves from one node to another according to the path that is learned in the learning stage. Vehicle localization at a node along the path is implemented to include three stages: SIFT-based feature matching, 2-D affine transformation, and analytic 3-D space transformation. The first stage is to match the *SIFT feature vectors* of two monitored object images, i.e., one taken in the navigation stage and the other taken in the learning stage. The second stage is to find an affine transformation from the matching results using the Hough transform [21]. In addition, the third stage is to find the vehicle location based on a new technique of analytic 3-D space transformation proposed in this paper. More details are described in the following, where the two images mentioned above taken in learning and navigation stages are denoted as I_L and I_N , respectively.

Algorithm 1: autonomous vehicle navigation process.

Stage 1—SIFT-based feature matching for finding similar feature point pairs.

- 1) Apply the SIFT to I_N to obtain a set F_N of SIFT feature vectors, and retrieve the corresponding feature vector set F_L of I_L , which was obtained in the learning stage.
- 2) Take every pair of *similar* feature vectors, i.e., one from F_L and the other from F_N , to define a group of four

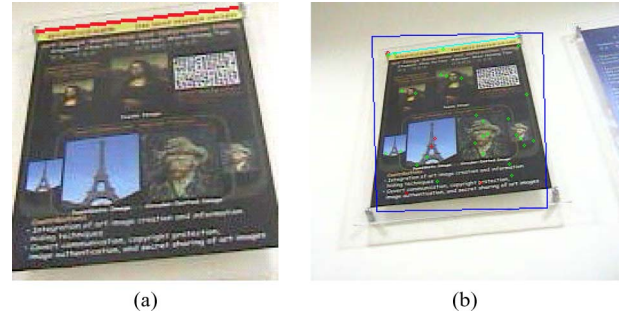


Fig. 3. Object images I_L and I_N taken in learning and navigation stages. (a) Red calibration line l_L detected or selected manually. (b) Cyan calibration line l_N (on the top of the poster) generated by applying the transform T .

parameters of an affine transformation from I_L to I_N , where the feature vector similarity is as defined in [18].

Stage 2—2-D affine transformation for finding the relative mapping between images taken in learning and navigation.

- 3) Put all found parameter groups into a Hough space.
- 4) Detect the peak in the space.
- 5) Find the affine transform T corresponding to the peak, and take it to represent the *best* relative mapping from I_L to I_N .

Stage 3—analytic 3-D space transformation for computing the deviation of the vehicle from the planned trajectory and conducting path correction.

- 6) Use T to transform the calibration line l_L , which was automatically detected or manually selected in I_L in the learning stage, into the image space of I_N , resulting in a new image line in I_N , which is denoted as l_N (see Fig. 3 for an example, and note that if the navigation incurs no deviation from the planned path, then l_L and l_N will perfectly match in position).
- 7) *Unambiguously* transform l_L and l_N into the RCS according to an *analytic* 3-D transformation process (proposed in this paper and described later in the next section) from the image space to the RCS to obtain two sets S_L and S_N of vehicle poses with respect to the RCS, i.e., one for the learning stage and the other for the navigation stage, respectively.
- 8) Use S_L and S_N to derive the translation V_t and the orientation θ_t of the vehicle’s current location with respect to its location planned in the learning stage, and call the parameter set (V_t, θ_t) the *deviation of the vehicle trajectory*.
- 9) Use the deviation (V_t, θ_t) to derive a sequence of vehicle commands to guide the vehicle to the correct path learned in advance, and continue the navigation work for the next session.

D. Advantages of Using SIFT for Object Image Matching

The use of the SIFT as described above is advantageous in several aspects. First, in traditional vehicle-localization techniques, *obvious* features like lines and curves on landmarks are detected. However, this limits the variety of objects for use as

landmarks because it is not always true that there exist obvious line or curve features on an object's surface. Use of the SIFT followed by the Hough transform removes this inconvenience; any point pattern on the object surface may be utilized, leading to great improvement in the flexibility to select objects as landmarks. Nearly all point-patterned object surfaces may be used as landmarks, as found in this paper. This also leads to the previously mentioned convenience of selecting a virtual calibration line on the object surface to construct an RCS for vehicle localization and path correction.

E. Sketch of the Proposed Learning Process

In the proposed learning stage, when the user drives the vehicle to visit each object to be monitored, relevant data are recorded, including the navigation path, the SIFT feature vectors of the monitored object, and the calibration line parameters. All data are saved in a *learning database* such that the learning process need only be conducted once, and the database can be repeatedly used in every navigation session. The proposed learning process is sketched in the following.

Algorithm 2: navigation path-learning process.

- 1) Direct the PTZ camera on the vehicle toward the object when the vehicle arrives at a spot in front of an object B .
 - 2) Take an image of B , and draw, as I_L , a rectangle to enclose B .
 - 3) Apply the SIFT to I_L to obtain a set F_L of SIFT feature vectors.
 - 4) Automatically detect or manually select a calibration line l_L in I_L by either of the two ways of calibration line creation mentioned previously.
 - 5) Save the set F_L and the parameters of l_L in the path node for B .
 - 6) Repeat the above steps for the next object to be monitored until all concerned along-path objects are "learned."
-

III. ANALYTIC 3-D SPACE TRANSFORMATION FROM IMAGE SPACE TO REAL-WORLD SPACE FOR VEHICLE LOCALIZATION

Vehicle localization and computation of the deviation of the vehicle trajectory are made possible by the use of the calibration line and unambiguous mapping of the image space to the real-world space specified by the previously mentioned RCS, as described in the following.

Let (X, Y, Z) denote the RCS that is set up on the calibration line L and one of its two endpoints, i.e., R_0 , with L taken to be the X -axis of the RCS and R_0 to be the origin. An illustration of the system is shown in Fig. 4. Recall that the calibration line L , possibly virtual, is assumed to be parallel to the floor so that the X - Y plane of the RCS, as illustrated, is parallel to the floor as well. Also, the images of L in I_L and I_N , respectively, are l_L and l_N , as mentioned previously. The configuration of the three axes of the RCS may be regarded as being similar to a *virtual Y-shaped corner* attached on the ceiling of a house, and therefore,

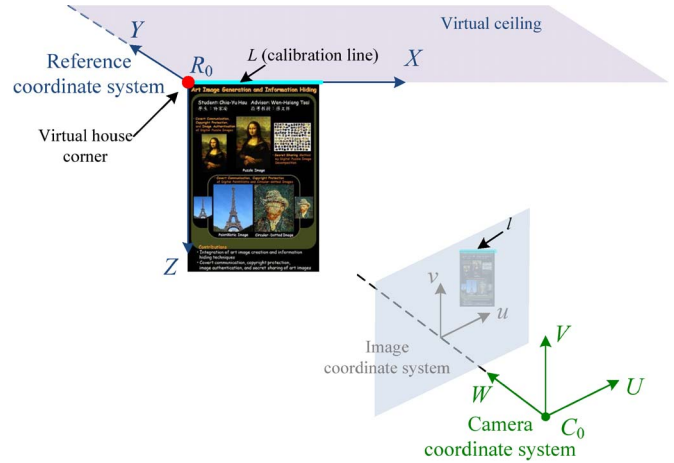


Fig. 4. Virtual house corner specified by a given calibration line (the cyan line on the top of the poster) and one of its end point (the red point on the left-top corner of the poster).

the vehicle location-estimation method using house corners, which was proposed by Chou and Tsai [9], can be employed and simplified to estimate the location (including the position and the direction) of the vehicle with respect to the RCS (i.e., with respect to the object to be monitored). This means that while the object is being monitored, localization of the vehicle with respect to the object is achieved simultaneously, as mentioned previously.

More specifically, the X - and Y -axes may be regarded to specify the two perpendicular lines on the ceiling plane of the *virtual house corner*, and the Z -axis specifies the *virtual line* perpendicular to the X - and Y -axes, as shown at the left top of Fig. 4. Also shown in the figure are 1) a *camera coordinate system* (CCS), which is set up on the camera equipped on the vehicle with the lens center as the origin and the optical axis as one of its three axes (the W -axis) and 2) an *image coordinate system* (ICS), which is set up in the image taken by the camera. Before deriving the location of the vehicle, we derive that of the camera as an intermediate result in the following.

Let the projection of L and R_0 in an image be denoted as l and r_0 , respectively. Also, let the image coordinates of r_0 be (u_0, v_0) . Assume that the equation of l in the ICS in terms of image coordinates (u, v) is described by $u + bv + c = 0$. The pose of the camera may be represented by three position parameters X_c, Y_c , and Z_c , as well as three orientation parameters θ, ψ , and ω , as described in the following.

- 1) Z_c is the *distance* from the camera to the ceiling of the virtual house corner, which is *assumed to be known* (manually measured in advance).
- 2) θ is the angle between the optical axis of the camera and the Y -axis of the RCS.
- 3) ψ is the angle of the optical axis of the camera with respect to the RCS, which is also *assumed to be known* (provided by the PTZ camera system).
- 4) ω is the swing angle of the camera, which is *set to zero* in this study.

Accordingly, the camera pose may be described *just* by three parameters X_c, Y_c , and θ , which may be derived in terms of the two coefficients b and c of the equation of l . At first,

we transform the RCS coordinates into the CCS coordinates using four steps: 1) Translate the origin R_0 of the RCS with coordinates (X_c, Y_c, Z_c) to the origin of the CCS; 2) rotate the $X-Y$ plane about the Z -axis through the pan angle θ such that the $Y-Z$ plane is parallel to the $V-W$ plane of the CCS (see Fig. 4); 3) rotate the $Y-Z$ plane about the X -axis through the tilt angle ψ such that the $X-Y$ plane is parallel to the $U-V$ plane; and 4) reverse the Z -axis such that the positive direction of the Z -axis is identical to the negative direction of the W -axis.

As a result, the transformation of the RCS coordinates (x, y, z) of a point in the 3-D real-world space into the CCS coordinates (u, v, w) may be described by

$$(u, v, w, 1) = (x, y, z, 1) \cdot T_r \quad (1)$$

where

$$T_r = \begin{bmatrix} \cos \theta & -\sin \theta \cos \psi & -\sin \theta \sin \psi & 0 \\ \sin \theta & \cos \theta \cos \psi & \cos \theta \sin \psi & 0 \\ 0 & \sin \psi & -\cos \psi & 0 \\ x_0 & y_0 & z_0 & 1 \end{bmatrix} \quad (2)$$

with

$$\begin{aligned} x_0 &= -X_c \cos \theta - Y_c \sin \theta \\ y_0 &= (X_c \sin \theta - Y_c \cos \theta) \cos \psi - Z_c \sin \psi \\ z_0 &= (X_c \sin \theta - Y_c \cos \theta) \sin \psi + Z_c \cos \psi. \end{aligned} \quad (3)$$

Next, let P be any point on the X -axis (the calibration line L) with RCS coordinates $(x, 0, 0)$. Then, its CCS coordinates (u_x, v_x, w_x) can be derived, using the matrix T_r described above, to be

$$\begin{aligned} (u_x, v_x, w_x, 1) &= (x, 0, 0, 1) \cdot T_r \\ &= (x \cos \theta + x_0, -x \sin \theta \cos \psi + y_0 \\ &\quad - x \sin \theta \sin \psi + z_0, 1). \end{aligned} \quad (4)$$

Therefore, we have

$$\begin{aligned} u_x &= x \cos \theta + x_0 \\ v_x &= -x \sin \theta \cos \psi + y_0 \\ w_x &= -x \sin \theta \sin \psi + z_0. \end{aligned} \quad (5)$$

Also, let (u_p, v_p) be the image coordinates of the projection of P in the ICS. Then, according to the camera's optical geometry, we have the following:

$$\begin{aligned} u_x &= \frac{w_x \cdot u_p}{f} \\ v_x &= \frac{w_x \cdot v_p}{f} \end{aligned} \quad (6)$$

where f is the camera focus length, which is assumed to be known. Substituting the values of u_x, v_x , and w_x of the three equations of (5) into the above two equations and eliminating

the variable x , we get the equation for the projection of the calibration line l (the X -axis) in the image plane as follows:

$$u_p + \frac{z_0 \cos \theta + x_0 \sin \theta \sin \psi}{-y_0 \sin \theta \sin \psi + z_0 \sin \theta \cos \psi} v_p - \frac{f \cdot (y_0 \cos \theta + x_0 \sin \theta \cos \psi)}{-y_0 \sin \theta \sin \psi + z_0 \sin \theta \cos \psi} = 0. \quad (7)$$

Furthermore, assume that the equation of the calibration line (l_N or l_L mentioned previously) in the ICS is $u_p + bv_p + c = 0$, which can be obtained from the image, i.e., b and c are known. Then, by comparing the coefficients of this equation with those of (7) and substituting the values of x_0, y_0 , and z_0 previously described in (3) into the result, we obtain the following equalities:

$$b = \frac{Y_c \sin \psi - Z_c \cos \theta \cos \psi}{-Z_c \sin \theta} \quad (8)$$

$$c = \frac{f \cdot (-Y_c \cos \psi - Z_c \cos \theta \sin \psi)}{-Z_c \sin \theta}. \quad (9)$$

Since b, c , and Z_c in the above equations are known values, we can now derive the values of Y_c and θ . This is done first by eliminating Z_c and Y_c from the above equalities to get

$$\tan \theta = \frac{f}{fb \cos \psi + c \sin \psi} \quad (10)$$

from which the value of θ can be obtained. Also, from (8), we can get the value of Y_c as

$$Y_c = \frac{Z_c \cos \theta \cos \psi - b Z_c \sin \theta}{\sin \psi}. \quad (11)$$

By substituting the values of θ and Y_c above as well as the coordinates (u_0, v_0) of r_0 (the projection of the origin R_0 of the RCS) into (7) with the values of x_0, y_0 , and z_0 described by (3), we get, finally, after some equation simplifications, the following for computing X_c :

$$X_c = \frac{u_0(Y_c \cos \theta \cos \psi + Z_c \sin \psi \sin \theta) - Y_c u_0 \sin \theta}{u_0 \sin \theta \cos \psi + v_0 \cos \theta}. \quad (12)$$

In summary, we have completed the derivations of (10)–(12) for computing the parameters X_c, Y_c , and θ of the camera pose using the following known data:

- 1) the coefficients b and c of the equation $u + bv + c = 0$ of the calibration line in the image;
- 2) the height Z_c of the virtual ceiling above the camera (i.e., the height of the calibration line above the camera in the real-world space);
- 3) the tilt angle ψ of the camera provided by the PTZ camera system;
- 4) the image coordinates (u_0, v_0) of the origin R_0 of the RCS and the focal length f of the camera.

IV. PATH CORRECTION

We now describe the proposed path-correction process for use in the navigation stage. With the camera pose described by (X_c, Y_c) and θ with respect to the RCS on the planar surface

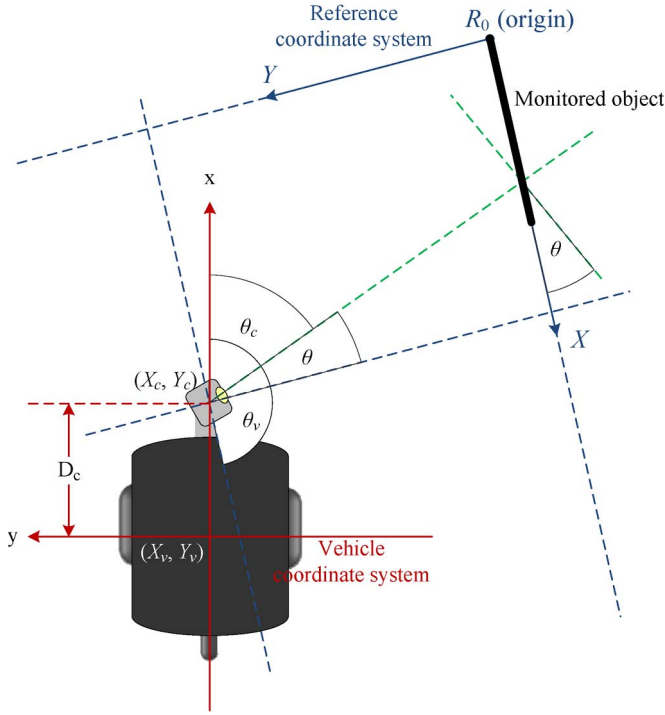


Fig. 5. Top view of the relations among the vehicle, camera, and reference coordinate systems.

of an object as derived in the previous section, we want to accordingly derive the location of the vehicle with respect to the RCS. For this, we define an additional coordinate system, which is called the *vehicle coordinate system* (VCS), as illustrated in Fig. 5. The VCS is 2-D with its origin located at the middle point of the line segment joining the centers of the two vehicle wheels, which is called the *center of the vehicle* in the sequel. The y -axis of the VCS is defined to align with this line segment, and the x -axis is defined to be perpendicular to the y -axis and parallel to the ground. The location of the vehicle with respect to the RCS is described by the orientation θ_v and the translation (X_v, Y_v) of the vehicle in the RCS, which are easy to derive by the geometry shown in Fig. 5 to be as follows:

$$\begin{aligned} \theta_v &= \theta_c + \theta + 90^\circ \\ X_v &= X_c - D_c \cos \theta_v \\ Y_v &= Y_c + D_c \sin \theta_v \end{aligned} \quad (13)$$

where θ_c is the pan angle of the camera provided by the PTZ camera system, and D_c is the distance between the camera and the center of the vehicle (assumed to be known in advance by manual measurement). For convenience, the vehicle location will be described by the 3-tuple (X_v, Y_v, θ_v) in the sequel.

We now describe how to correct the path of the vehicle at a certain node N in front of an object using the information of the *current vehicle location* $S_N = (X_N, Y_N, \theta_N)$, as well as that of the *learned vehicle location* $S_L = (X_L, Y_L, \theta_L)$, which is obtained through a similar process for obtaining the current vehicle location, as described previously. On the other hand, let the current and learned vehicle locations in the global coordinate system (GCS) be described by $S'_N = (X'_N, Y'_N, \theta'_N)$ and

$S'_L = (X'_L, Y'_L, \theta'_L)$, respectively, where the latter was provided by the odometer during the learning stage, whereas the former is to be derived now using the relation among the RCS, VCS, and GCS illustrated in Fig. 5. The basic idea is to find the relative angle and translation between the current vehicle location S_N and the learned one S_L in the RCS, and then, add the angle and translation to the learned vehicle location S'_L in the GCS. The result is finally used to guide the vehicle to navigate to the location of the next learned node N' in the navigation path, whose location is denoted as $S''_L = (X''_L, Y''_L, \theta''_L)$ in the GCS. The details are as follows, with Fig. 6 as an illustration for notation references.

Algorithm 3: path-correction process.

- 1) Compute the relative angle θ_r between θ_N and θ_L by

$$\theta_r = \theta_N - \theta_L. \quad (14)$$

- 2) Compute the relative translation in the following way.

- a) Compute the angle

$$\phi = \tan^{-1} \left(\frac{Y_N - Y_L}{X_N - X_L} \right) - \theta_N. \quad (15)$$

- b) Compute the angle ϕ_G of the line segment from S'_N to S'_L in the GCS by

$$\phi_G = \theta'_L - \phi - \theta_r. \quad (16)$$

- c) Compute the translation (X'_r, Y'_r) in the GCS with respect to the learned vehicle location S'_L by

$$\begin{aligned} X'_r &= D_r \cos \phi_G \\ Y'_r &= D_r \sin \phi_G \end{aligned} \quad (17)$$

where

$$D_r = \sqrt{(X_N - X_L)^2 + (Y_N - Y_L)^2}. \quad (18)$$

- 3) Compute the desired current vehicle location $S'_N = (X'_N, Y'_N, \theta'_N)$ using the following:

$$\begin{aligned} X'_N &= X'_L - X'_r \\ Y'_N &= Y'_L - Y'_r \\ \theta'_N &= \theta'_L - \theta_r. \end{aligned} \quad (19)$$

- 4) Compute the vector V_t from S'_N to S''_L of the next node N' and the direction θ_t of V_t by

$$\begin{aligned} V_t &= (X''_L - X'_N, Y''_L - Y'_N) \\ \theta_t &= \tan^{-1} \left(\frac{Y''_L - Y'_N}{X''_L - X'_N} \right). \end{aligned} \quad (20)$$

- 5) Use V_t and θ_t to derive the following vehicle commands for path correction.

- a) Turn the angle $\theta'_N - \theta_t$ to guide the vehicle to head to the direction of V_t to approach the next node N' .

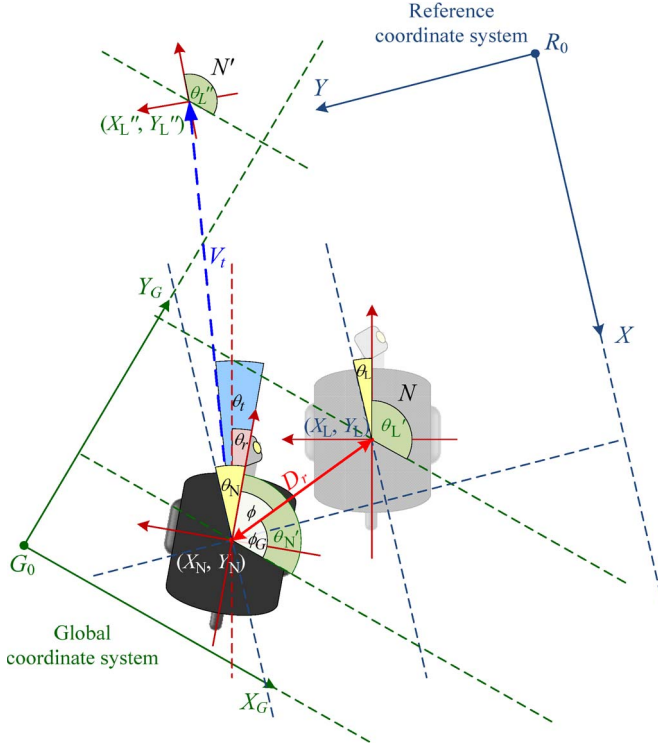


Fig. 6. Top view of the relations among the RCS, the VCS, and the GCS, as well as corresponding angles of the vehicle, where the learned vehicle location is at position (X_L, Y_L) in the RCS, the current vehicle location is at position (X_N, Y_N) in the RCS, and the location of the next node N' is at position (X_L'', Y_L'') in the GCS.

- b) Move ahead for the following distance to reach N' :

$$|V_t| = \sqrt{(X_L'' - X_N)^2 + (Y_L'' - Y_N)^2}.$$

- c) Turn the angle $\theta_t' = \theta_L'' - \theta_t$ after arriving at N' to resume the original vehicle moving direction at N' .

V. OBJECT IMAGE MATCHING

In the navigation stage, the vehicle stops in front of each object to be monitored by the use of the learned path data. However, the stop position in front of an object may not be accurate in every navigation session but is just *close to* the one recorded in the learning stage. This results in a slight change in the viewing angle of the object from the camera or, equivalently, in the camera pose with respect to the object. An image acquired of the same object will so be different in translation, scaling, and orientation from the one taken in the learning stage. Thus, a method with the ability to match corresponding objects in images that are taken with different camera poses is needed.

In the past, SIFT has been proven to be one of the robust image-matching techniques that use local invariant feature descriptors with respect to different geometrical changes [19]. To allow efficient matching between images, each image is processed to extract feature points, each of which is then represented as a SIFT feature vector, as mentioned previously. Each SIFT feature vector consists of local image measures that are

invariant to image translation, scaling, and rotation and partially invariant to 3-D viewpoint changes. In this study, we utilize such invariance properties of SIFT feature vectors to match object images. Specifically, the SIFT process [18] includes four major stages to generate a set of SIFT feature vectors: 1) selection of a set of feature points in a scale space of the input image; 2) localization of them to determine their locations (x, y) and scales s ; 3) assignment of orientations θ to them; and 4) generation of a descriptor Φ for each of the feature points as a vector of all the gradient-orientation histogram entries in a region around the feature point.

By using the aforementioned SIFT process for our case here, the object images I_N and I_L taken in the navigation and learning stages are first transformed to obtain two sets of SIFT feature vectors: one set F_N from I_N and the other F_L from I_L , as mentioned previously. According to the matching algorithm proposed by Lowe [18], [20], the best candidate match for each feature vector f_N in F_N can be found by identifying its *nearest neighbor* in F_L . Each nearest neighbor is defined as the feature vector with the closest Euclidean distance to the feature vector f_N . Then, an affine transform between the match pairs is estimated. For this purpose, many well-known fitting methods, such as the random sample consensus algorithm [22], can be used. However, according to [18], a better performance can be obtained using the Hough transform [21], which is adopted in this paper to identify the *best* affine transform between the match pairs.

In more detail, using the aforementioned notations, for each feature vector denoted as $f_N(x_N, y_N, s_N, \theta_N, \Phi_N)$, a point (x'_N, y'_N) in the neighborhood of (x_N, y_N) can be generated using s_N and θ_N to be

$$\begin{aligned} x'_N &= x_N + k \cdot s_N \cdot \cos \theta_N \\ y'_N &= y_N + k \cdot s_N \cdot \sin \theta_N \end{aligned} \quad (21)$$

where k is a constant value. In addition, the corresponding point (x'_L, y'_L) in F_L can be computed in the same manner. Then, an affine transform defined in the following way can be used to obtain mapping from the pair (x_L, y_L) and (x'_L, y'_L) in F_L to the pair (x_N, y_N) and (x'_N, y'_N) in F_N in terms of a scaling factor s , an orientation θ , and a translation (t_x, t_y) described in the following:

$$\begin{bmatrix} x_L & -y_L & 1 & 0 \\ y_L & x_L & 0 & 1 \\ x'_L & -y'_L & 1 & 0 \\ y'_L & x'_L & 0 & 1 \end{bmatrix} \times \begin{bmatrix} m \\ n \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x_N \\ y_N \\ x'_N \\ y'_N \end{bmatrix} \quad (22)$$

where $m = s \cdot \cos \theta$, and $n = s \cdot \sin \theta$. Equation (22) can be solved to obtain the parameters m , n , and (t_x, t_y) . In addition, accordingly, the values s and θ can be computed by the following:

$$\theta = \tan^{-1} \left(\frac{n}{m} \right) \quad \text{and} \quad s = \frac{m}{\cos \theta}. \quad (23)$$

The above process creates a set of pose parameters (t_x, t_y, θ, s) for each match pair of the SIFT feature vectors. Next, a Hough space is created to find the *best* affine transform for the computed pose parameters (t_x, t_y, θ, s) of *all* the match pairs. Each match pair will yield a *vote* in the Hough space,

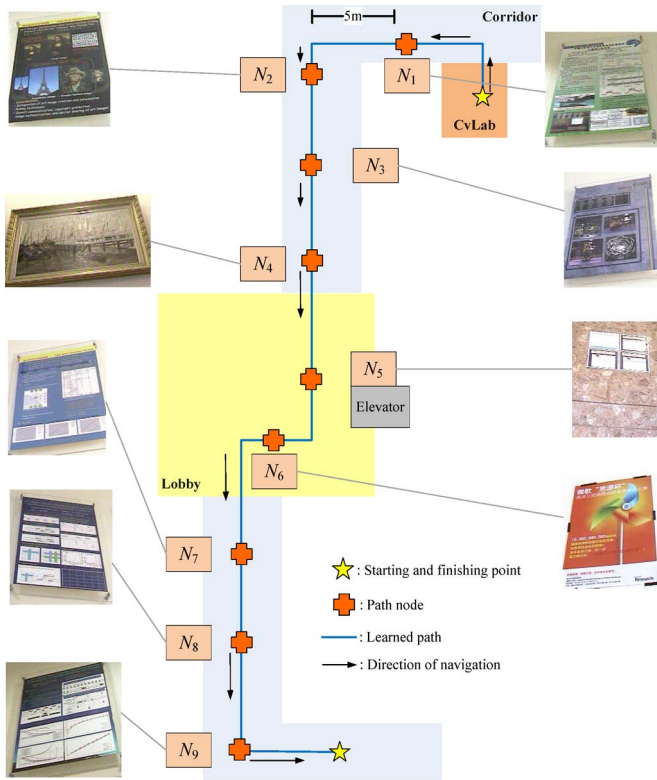


Fig. 7. Learned navigation map.

and a peak that is found in the space with the largest number of votes is finally taken to specify the best affine transform from I_L to I_N .

VI. EXPERIMENTAL RESULTS

To test the feasibility of the proposed method, we conducted a series of experiments with a commercially available vehicle called Pioneer 3-DX manufactured by MobileRobots, Inc., and a PTZ camera produced by Axis Communications, Ltd. The testing environment, as shown in Fig. 7, is a large building with an M-shaped corridor whose length is 75.83 m. In addition, nine objects labeled as N_1 through N_9 were monitored, including one painting, seven posters, and one granite-surfaced wall plane, which are roughly evenly distributed in the entire range of the corridor with an average between-node distance of 7.58 m.

At first, in the learning stage, the vehicle was controlled to learn a path in the corridor and the nine objects along the path. The objects are all on walls at the two sides of the path. Whenever the vehicle arrived at a spot in front of one object, the object features were extracted from the image that is acquired at the spot, which, together with the vehicle location, were recorded into the vehicle system. As a result, a navigation map that contains nine path nodes was created, as illustrated by Fig. 7.

A. Vehicle-Localization Accuracy

To evaluate the accuracy of vehicle localization performed by the proposed method, we compared the estimated vehicle positions and orientations at the nine nodes N_1 through N_9 with the corresponding real values measured manually. In more detail, when the vehicle was guided to navigate to each node

TABLE I
RESULTS OF ACCURACY ANALYSIS OF CONDUCTED
VEHICLE LOCALIZATION

Image No.	Position (cm)						Orientation (degree)			
	X_e	Y_e	X_r	Y_r	X_{err} ratio	Y_{err} ratio	θ_r	θ_e	θ_{err}	
N_1	1	49.38	121.94	47.8	120	3.31%	1.62%	180	178.11	1.89
	2	49.53	120.84	47.8	120	3.61%	0.70%	180	178.21	1.79
	3	49.34	124.55	47.8	120	3.23%	3.79%	180	178.59	1.41
N_2	4	66.54	144.94	66.4	139.5	0.22%	3.90%	180	178.97	1.03
	5	66.67	141.73	66.4	139.5	0.41%	1.60%	180	179.24	0.76
	6	66.73	144.86	66.4	139.5	0.50%	3.84%	180	178.93	1.07
N_3	7	91.73	137.42	90	138	1.92%	0.42%	180	179.62	0.38
	8	91.66	138.31	90	138	1.85%	0.22%	180	179.63	0.37
	9	91.28	139.43	90	138	1.42%	1.03%	180	179.59	0.41
N_4	10	59.24	203.05	59.5	203	0.44%	0.03%	190	190.97	0.97
	11	58.90	203.01	59.5	203	1.01%	0.00%	190	190.84	0.84
	12	58.97	202.66	59.5	203	0.89%	0.17%	190	190.75	0.75
N_5	13	51.86	148.03	53.5	146	3.06%	1.39%	153	151.50	1.50
	14	51.64	147.80	53.5	146	3.47%	1.23%	153	151.05	1.95
	15	51.40	147.23	53.5	146	3.93%	0.84%	153	151.55	1.45
N_6	16	21.56	127.16	22	124.5	1.99%	2.14%	173	174.46	1.46
	17	22.61	128.15	22	124.5	2.79%	2.93%	173	173.83	0.83
	18	22.64	126.15	22	124.5	2.90%	1.33%	173	174.17	1.17
N_7	19	30.14	123.16	29.6	119	1.82%	3.50%	180	178.69	1.31
	20	30.72	122.16	29.6	119	3.78%	2.66%	180	178.49	1.51
	21	28.63	123.56	29.6	119	3.28%	3.83%	180	177.99	2.01
N_8	22	-80.44	168.95	-77.8	168	3.39%	0.57%	26	26.18	0.18
	23	-81.61	170.10	-77.8	168	4.90%	1.25%	26	26.37	0.37
	24	-80.98	169.58	-77.8	168	4.09%	0.94%	26	26.01	0.01
N_9	25	16.48	77.84	16.7	74.2	1.33%	4.90%	175	175.73	0.73
	26	16.50	77.36	16.7	74.2	1.20%	4.26%	175	175.88	0.88
	27	17.05	76.58	16.7	74.2	2.07%	3.21%	175	175.43	0.43
Average					2.33%	1.94%			1.02	
Standard deviation					1.32%	1.5%			0.565	

in the learning stage, the real vehicle location described by (X_r, Y_r, θ_r) at the node was measured as the reference data, where 1) X_r and Y_r denote, respectively, the distances in the X - and Y -directions from the vehicle center to the origin of the RCS, which is set up on the planar-surfaced object on the wall; and 2) θ_r denotes the angle of the vehicle's moving direction with respect to the $X-Z$ plane, which coincides with the wall plane. After the proposed vehicle localization process was performed, the estimated vehicle location parameters (X_e, Y_e, θ_e) were computed and compared with the corresponding reference data. Such comparisons were conducted three times at each node. Table I includes a summation of the estimated and reference data, as well as the average and the standard deviation of the *error ratios* yielded by the comparison. Here, the *errors* $(X_{err}, Y_{err}, \theta_{err})$ of a vehicle-localization result are defined as the absolute differences between the estimated and real vehicle location data, respectively, and the *error ratio* of an estimated distance in the X - or Y -direction is defined as the ratio of the error X_{err} or Y_{err} over the real one, respectively.

From Table I, we can see that all the estimated distances have error ratios that are smaller than 5% and that all the estimated orientations have errors that are smaller than 2° . The averages of the X - and Y -distance error ratios, as well as those of the orientation errors, are 2.33%, 1.93%, and 1.02° , respectively. The standard deviation of these error parameters are 1.32%, 1.5%, and 0.565° , respectively, which are also small. Such vehicle-localization results may be considered to be accurate enough for *smooth* vehicle navigation applications, as shown by our experiments.

TABLE II
RESULTS OF PATH-CORRECTION ACCURACY ANALYSIS

	Position (cm)							Orientation (degree)				
	X_r	Y_r	X_n	Y_n	X'_n	Y'_n	X'_{err} ratio	Y'_{err} ratio	θ_r	θ_n	θ'_n	θ'_{err}
P_{11}			65.8	111.4	74.2	120.9	5.54%	0.72%	180	180.8	0.8	1.2
					74.6	120.4	6.06%	0.37%	180	181.2	1.2	1.3
					74.6	120.4	6.16%	0.34%	180	181.3	1.3	0.2
					68.7	119.9	2.24%	0.12%	180	180.2	0.2	0.2
N_1 P_{12}	70.3	120	73.1	134.6	69.7	120.3	0.80%	0.29%	180	184	179.8	0.2
					68.7	119.9	2.24%	0.12%	180	180.2	0.2	0.2
					72.8	120.0	3.63%	0.03%	180	178.9	1.1	0.7
P_{13}			60.7	134.6	74.5	120.2	6.04%	0.13%	181	179.3	0.7	1.0
					72.8	120.6	3.55%	0.53%	180	179.0	1.0	2.1
					89.9	140.3	2.03%	0.55%	182.1	182.1	2.1	2.2
P_{21}		94.6	142.9	90.1	140.1	2.27%	0.45%	193	182.2	2.2	2.4	
					89.8	139.1	2.59%	0.31%	182.4	182.4	2.4	1.3
					86.8	140.7	2.39%	0.86%	178.7	178.7	1.3	1.2
N_2 P_{22}	88.9	139.5	90.8	115.6	87.1	141.2	2.06%	1.22%	180	179	178.8	1.0
					87.2	140.8	1.93%	0.94%	179.0	179.0	1.0	0.4
					89.7	141.2	0.85%	1.20%	179.6	179.6	0.4	0.2
P_{23}		53.3	148.0	88.6	141.4	0.38%	1.36%	186	179.8	0.2	0.1	
					89.3	141.1	0.46%	1.18%	180.1	180.1	0.1	0.4
					113.5	140.7	0.90%	1.95%	179.6	179.6	0.4	0.0
P_{31}		85.7	124.7	114.4	140.1	1.67%	1.49%	170	180.0	0.0	0.1	
					113.9	139.6	1.29%	1.19%	179.9	179.9	0.1	0.6
					111.6	139.2	0.82%	0.84%	179.4	179.4	0.6	0.9
N_3 P_{32}	112.5	138	93.5	154.4	111.4	139.3	1.01%	0.97%	180	183	179.1	0.8
					110.9	139.6	1.38%	1.14%	179.2	179.2	0.8	0.5
					113.8	141.7	1.15%	2.71%	179.5	179.5	0.5	0.4
P_{33}		132.5	133.7	114.6	141.2	1.85%	2.31%	181	179.6	0.4	0.1	
					115.6	140.2	2.74%	1.56%	180.1	180.1	0.1	0.8
Average							2.37%	0.92%				0.7
Standard deviation							1.74%	0.68%				0.7

B. Path-Correction Accuracy

To show the effectiveness of the proposed path correction process, we compared the corrected vehicle location yielded by the process with the one learned in the learning stage at each of the three selected path nodes N_1 through N_3 . More specifically, at each node, we put the vehicle at three different testing locations described by $S_n = (X_n, Y_n, \theta_n)$ in the RCS. Next, we performed the vehicle-localization and path-correction processes to guide the vehicle to move toward the learned location $S_r = (X_r, Y_r, \theta_r)$, yielding a *corrected location* $S'_n = (X'_n, Y'_n, \theta'_n)$. Then, we computed the path-correction accuracy by finding the *location errors* $(X'_{err}, Y'_{err}, \theta'_{err})$, which are the absolute differences between the parameters of S'_n and S_r , respectively. Here, the values of the location parameters in S_r , S_n , and S'_n were all measured manually. The results are summarized in Table II, in which the testing locations are denoted as P_{11} through P_{33} . From the table, we can see that the largest distance error ratio in the X - and Y -directions occurred at P_{11} , which is about 6%, and the largest orientation error occurred at P_{21} , which is about 2.4° . The averages of the distance error ratios in the X - and Y -directions as well as that of the orientation error are 2.37%, 0.92%, and 0.8° , respectively. The standard deviations of these error parameters are 1.74%, 0.68%, and 0.7° , respectively. Again, these data are all small in value. Accordingly, the maximum allowable between-node vehicle traveling distance may be computed by $40 \text{ cm} / \sin(0.8^\circ) = 28.6 \text{ m}$, under the condition that the largest deviation from the trajectory is set to be the reasonable value of 40 cm in the corridor of our experimental environment, whose width is 2.5 m. Even in the worst case with an orientation error of 2.4° , the distance is $40 \text{ cm} / \sin(2.4^\circ) = 9.55 \text{ m}$, which is



Fig. 8. Experimental results of object monitoring and path correction at four selected path nodes. (a) Labels of monitored objects. (b) Vehicle monitoring the objects at the nodes. (c) Matching results and the calibration lines used for path correction. (d) Images of the learned objects.

still larger than the average between-node distance of 7.5 m in our experimental environment mentioned previously. As a summary, as these experiment results illustrate, the proposed method is feasible for guiding the vehicle to accurately navigate to the next path node in a normal corridor.

C. Indoor Security-Patrolling Navigation

We have also conducted many experiments of complete security patrolling navigations in our M-shaped corridor environment, all according to the learned navigation path, as shown in Fig. 7. Whenever the vehicle was guided to arrive at a learned node in front of an object, object image matching was performed to check the existence of the object. If successful, the deviation of the trajectory of the vehicle was computed according to the matching result, and the proposed path correction was performed to continue its navigation on the path; otherwise, a warning message was issued. Some results, together with the acquired monitored object images, are shown in Fig. 8, where Fig. 8(b) shows a view of the vehicle arriving at a node with a monitored object, Fig. 8(c) shows the matching result and a computed calibration line in the image, and Fig. 8(d) shows the object image taken in the learning stage. The average navigation speed of the vehicle is 0.3 m/s or, equivalently, 1.08 km/h. The average navigation cycle, which includes the five processes of image taking, object image matching, vehicle localization, object monitoring, and path correction, is about

TABLE III
RESULTS OF ACCURACY ANALYSIS OF THE NAVIGATION TRAJECTORIES
TRAVERSED BY THE VEHICLE WITH RESPECT TO THE REFERENCE
NAVIGATION PATH

	Position (cm)							Orientation (degree)		
	X_r	Y_r	X_c	Y_c	D_{err}	D_r	D_{err} ratio	θ_r	θ_c	θ_{err}
N_1	385.0	447.0	380.7	446.6	4.35	828.0	0.52%	89.5	88.7	0.7
N_2	284.9	944.3	285.8	943.1	1.48	600.5	0.25%	180.0	180.6	0.6
N_3	-426.4	924.2	-428.9	924.2	2.55	711.6	0.36%	181.6	181.1	0.5
N_4	-940.4	948.5	-944.6	948.3	4.25	514.6	0.83%	177.3	180.7	3.4
N_5	-1842.6	928.8	-1843.6	926.1	2.87	902.4	0.32%	181.3	180.4	0.8
N_6	-2274.1	1188.9	-2272.5	1187.3	2.21	606.4	0.36%	90.0	90.8	0.8
N_7	-3122.3	1324.2	-3120.5	1322.5	2.50	996.6	0.25%	180.7	181.2	0.6
N_8	-3758.6	1331.8	-3759.4	1330.1	1.83	636.3	0.29%	179.3	180.4	1.1
N_9	-4636.0	1329.6	-4638.3	1330.0	2.34	877.4	0.27%	180.1	177.6	2.6
Finish point	-4632.0	419.6	-4630.0	419.1	2.06	910.0	0.23%	-89.7	-90.8	1.1
Average					2.6	758.4	0.37%			1.2
Standard deviation					1.0		0.18%			1.0

4 s, in which the image-matching process takes about 2.3 s. Since there is no real-time requirement in the navigation process from the viewpoint of security patrolling, this amount of time required to monitor each node is considered to be tolerable. In addition, if necessary, the implementation of the proposed algorithms or the speed of the used personal computer in the vehicle system can be improved to raise the overall computation speed of the system.

To show the *overall accuracy of the navigation trajectory* traversed by the vehicle with respect to the reference navigation path, at each path node in a navigation session, which we conducted in our experiments, we measured further both the *reference location* described by $S_r = (X_r, Y_r, \theta_r)$ and the *corrected location* by $S_c = (X_c, Y_c, \theta_c)$ after the path-correction process was performed, followed by computations of the *location errors* (D_{err}, θ_{err}), which are defined as the differences between the parameters of the two locations. The computation results of these errors and their error ratios with respect to the between-node distances D_r are summarized in Table III. In addition, the measured reference and corrected positions of the vehicle at the nine path nodes are illustrated in Fig. 9. The average errors in position and orientation at each node are 2.6 cm and 1.2° with standard deviations of 1.0 cm and 1.0° , respectively, and the average error ratio is 0.37% with a standard deviation of 0.18%. These data are all small enough again. They, together with Fig. 9, show that the errors resulting from mechanic and vehicle-localization errors can be *reset* at each path node by the proposed vehicle-localization and path-correction processes. This fact shows that, with a sufficient number of nodes distributed along the navigation path, the vehicle can accurately and smoothly navigate according to the learned path from a viewpoint of the entire navigation path range.

We also compared our results *both* with those yielded by the ceiling-based method proposed by Xu *et al.* [13] for robot positioning, where the average position error ratio with respect to a traversed distance of 8.742 m was 0.34%, and with those yielded by the visual path-following method proposed by Achar

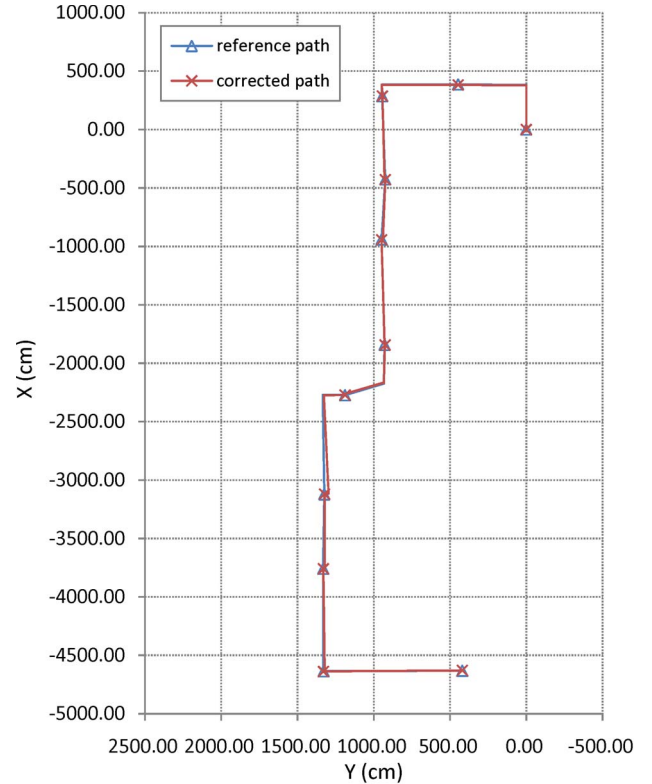


Fig. 9. Results of accuracy analysis of visited spots in front of the objects yielded by the proposed method. The blue line segments were the planned reference path, and the red ones were the corrected navigation path.

and Jawahar [26], where the average position error ratio with respect to a traversed distance of 80 m was 0.48%. Obviously, our results with the average position error ratio of only 0.37% for a traversed distance of 75.83 m have better accuracy.

VII. CONCLUSION

A novel method for vehicle localization utilizing images of objects to be monitored has been proposed. The method is based on the use of a SIFT-based object image-matching process, a 2-D affine transformation scheme, and an analytic 3-D space transformation technique for vehicle-location estimation. The method can simultaneously perform vehicle-location estimation and object monitoring. At first, a relative transformation between each image taken in the learning stage and a corresponding one taken in the navigation stage has been derived by matching the SIFT feature vector pairs extracted from the images and finding the best affine transform yielded by the Hough transform. Next, it has been proposed to set up a calibration line in the image of each object to be monitored to compute the deviation of the vehicle trajectory utilizing the found relative transformation and the aforementioned analytic 3-D space transformation newly proposed in this paper. Finally, a path-correction scheme has been proposed to compute necessary path adjustment according to the found vehicle trajectory deviation and to accordingly guide the vehicle to move toward the next path node.

An advantage of the proposed method over others is that nearly any planar-surfaced objects can be used as a landmark

for vehicle-location estimation, resulting in the improvement of flexibility in choosing landmarks. By adopting the SIFT and the Hough transform, as long as a sufficient number of feature pairs are available, the vehicle can precisely estimate its location with respect to every monitored object. The experimental results and the accuracy analysis of the yielded data revealed the feasibility and practicality of the proposed system for smooth and accurate navigation and object-security monitoring in indoor environments.

Since the SIFT process in the proposed system can be substituted separately, more accurate and efficient feature-matching techniques, like some recently proposed SIFT accelerations and feature-transformation techniques [23]–[25], may be fitted into the proposed method to yield better results. Other future works may be directed to extending the proposed method to deal with more types of objects in more complicated environments.

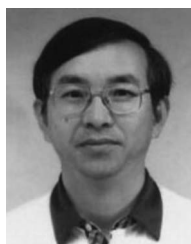
REFERENCES

- [1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.
- [2] G. J. Jang, S. H. Lee, and I. S. Kweon, "Color landmark based self-localization for indoor mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, May 2002, vol. 1, pp. 1037–1042.
- [3] S. Segvic and S. Ribaric, "Determining the absolute orientation in a corridor using projective geometry and active vision," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 696–710, Jun. 2001.
- [4] C. H. Ku and W. H. Tsai, "Obstacle avoidance in person following for vision-based autonomous land vehicle guidance using vehicle location estimation and quadratic pattern classifier," *IEEE Trans. Ind. Electron.*, vol. 48, no. 1, pp. 205–215, Feb. 2001.
- [5] N. X. Dao, B.-J. You, S.-R. Oh, and M. Hwangbo, "Visual self-localization for indoor mobile robots using natural lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, Oct. 2003, vol. 2, pp. 1252–1257.
- [6] Z. Jia, A. Balasuriya, and S. Challa, "Recent developments in vision based target tracking for autonomous vehicles navigation," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 765–770.
- [7] K. L. Chiang and W. H. Tsai, "Vision-based autonomous vehicle guidance in indoor environments using odometer and house corner location information," in *Proc. IEEE Int. Conf. Intell. Inf. Hiding Multimed. Signal Process.*, Pasadena, CA, Dec. 2006, pp. 415–418.
- [8] C. Micheloni, G. L. Foresti, C. Picciarelli, and L. Cinque, "An autonomous vehicle for video surveillance of indoor environments," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 487–498, Mar. 2007.
- [9] H. L. Chou and W. H. Tsai, "A new approach to robot location by house corners," *Pattern Recognit.*, vol. 19, no. 6, pp. 439–451, 1986.
- [10] T. Okuma, K. Sakaue, H. Takemura, and N. Yokoya, "Real-time camera parameter estimation from images for a mixed reality system," in *Proc. 15th Int. Conf. Pattern Recognit.*, Barcelona, Spain, Sep. 2000, vol. 4, pp. 482–486.
- [11] J. Huang, C. Zhao, Y. Ohtake, H. Li, and Q. Zhao, "Robot position identification using specially designed landmarks," in *Proc. IEEE Conf. Instrum. Meas. Technol.*, Sorrento, Italy, Apr. 2006, pp. 2091–2094.
- [12] C. J. Wu and W. H. Tsai, "Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 546–555, May 2009.
- [13] D. Xu, L. Han, M. Tan, and Y. F. Li, "Ceiling-based visual positioning for an indoor mobile robot with monocular vision," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1617–1628, May 2009.
- [14] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue, "View-based approach to robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Takamatsu, Japan, Nov. 2000, vol. 3, pp. 1702–1708.
- [15] G. Blanc, Y. Mezour, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 3365–3370.
- [16] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory," *Robot. Auton. Syst.*, vol. 55, no. 4, pp. 345–356, Apr. 2007.
- [17] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, Oct. 2007, pp. 3872–3877.
- [18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, Madison, WI, Jun. 2003, vol. 2, pp. 257–263.
- [20] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Kauai, HI, Dec. 2001, vol. 1, pp. 682–688.
- [21] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary patterns," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.
- [22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [23] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Washington, DC, Jun. 2004, vol. 2, pp. 506–513.
- [24] H. Lejsek, F. H. Asmundsson, B. T. Jonsson, and L. Amsaleg, "Scalability of local image descriptors: A comparative study," in *Proc. 14th Annu. ACM Int. Conf. Multimed.*, Santa Barbara, CA, Oct. 2006, pp. 589–598.
- [25] J. M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, Apr. 2009.
- [26] S. Achar and C. V. Jawahar, "Adaptation and learning for image based navigation," in *Proc. 6th Indian Conf. Comput. Vis. Graph. Image Process.*, Bhubaneswar, India, Dec. 2008, pp. 103–110.



Kuan-Chieh Chen received the B.S. and M.S. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2005 and 2008, respectively.

He is currently a Research Assistant with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan. His current research interests include computer vision, robotics, pattern recognition, and image processing, as well as all of their applications.



Wen-Hsiang Tsai (SM'91) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1973, the M.S. degree in electrical engineering from Brown University, Providence, RI, in 1977, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1979.

Since 1979, he has been with National Chiao Tung University (NCTU), Hsinchu, Taiwan, where he is currently a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Department of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan and, from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taichung, Taiwan. He has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has been a consultant to many major research organizations in Taiwan. He has published 141 journal papers and 223 conference papers. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications.

Dr. Tsai has received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society in Taiwan.