A STUDY ON AUTONOMOUS VEHICLE NAVIGATION BY 2D OBJECT IMAGE MATCHING AND 3D COMPUTER VISION ANALYSIS FOR INDOOR SECURITY PATROLLING APPLICATIONS*

Kuan-Chieh Chen (陳冠傑)[1] and Wen-Hsiang Tsai (蔡文祥)[1, 2]
[1] *Institute of Multimedia Engineering, College of Computer Science*
National Chiao Tung University, Hsinchu, Taiwan, R. O. C.
[2] *Department of Computer Science and Information Engineering*
Asia University, Taiwan, Taiwan, R. O. C.
E-mail: whtsai@cis.nctu.edu.tw

ABSTRACT

A vision-based vehicle system for security patrolling in indoor environments using an autonomous vehicle is proposed. A small vehicle with wireless control and a web camera which has the capabilities of panning, tilting, and zooming is used as a test bed. The vehicle navigates according to the node data in a path map created in the learning phase, and monitors concerned objects by a simplified scale-invariant feature transform algorithm proposed in this study. Accordingly, the features of each monitored object are extracted from acquired images, and are matched with the corresponding learned data by the Hough transform. Furthermore, a vehicle location estimation technique for path correction utilizing the monitored object matching result is proposed. Good experimental results show the flexibility and feasibility of the proposed methods for the application of security patrolling in indoor environments.

Keywords: Vehicle, patrolling, security surveillance, location, SIFT.

1. INTRODUCTION

In recent years, studies on vision-based autonomous vehicle navigation are in high prominence because of its great potential in various applications and the developments of computer vision techniques [1-13]. Autonomous vehicles are becoming more capable to perform a great variety of dangerous or dreary works in replacement of human beings, for example, interoffice document delivering, unmanned transportation, house cleaning, security patrolling, etc.

To develop autonomous vehicle systems for indoor security patrolling applications, the most critical issue is to guide the vehicle smartly to navigates in indoor environments. Facing this challenge, learning artificial landmarks or specific scene features in the environment and locating the vehicle by landmark or feature matching are feasible solutions. Although many works based on this idea have been developed in the past decade, most of them can only learn landmarks with special shapes or in ideal backgrounds like pure-colored ones, resulting in unreasonable restrictions on environments in which the vehicle can navigate. Therefore, it is desired in this study to design a method utilizing the technique of monitored-object image matching for vehicle location estimation. The idea, simply speaking, is to analyze the 3D geometric transformation of different monitored object views to estimate the vehicle location.

More specifically, in a traditional vision-based autonomous vehicle navigation system, the vehicle is usually equipped with a fixed pinhole camera, and the view of the vehicle is restricted to be at a lower area. Instead of using a fixed pinhole camera, we equip the vehicle with a pen-tilt-zoom camera (PTZ camera) in this study. With the PTZ camera and its movement, the view of the vehicle may be extended to a wider range. Then we can monitor both objects located higher than the camera and obstacles placed lower than the camera, by the images taken with the PTZ camera.

For object recognition and matching, Lowe [14] proposed the scale-invariant feature transform (SIFT) to extract features from given images as descriptors and used a best-bin first algorithm for SIFT descriptor matching, as mentioned previously. Since in the navigation phase, the position of the same monitored object will be just close to, instead of exactly at, the one found in the learning phase, resulting in a slight variation on the scale of the taken images, we propose in this study a simplified SIFT which reduces the difference of Gaussian scale layers. It is faster than the original SIFT to meet real-time security monitoring needs.

In the remainder of this study, we first describe the vehicle learning and guidance principles in Section 2. Then we describe the proposed method for detecting monitored objects by object image matching in Section 3. The proposed vision-based vehicle location estimation by object image matching results to correct the odometer records in the vehicle is described in Section 4. Some experimental results are described in Section 5, followed by conclusions in Section 6.

* This work was supported by the Ministry of Economic Affairs under Project No. MOEA 94-EC-17-A-02-S1-032 in Technology Development Program for Academia.

2. LEARNING AND GUIDANCE PRINCIPLES

The appearance of the vehicle used as a test bed in this study is shown in Fig 1. We use the odometer to provide the position of the vehicle and analyze the image captured by the PTZ camera equipped on the vehicle to monitor higher-located objects as well as the surrounding environment. We divide the work conducted by the system into two phases: the learning phase and the navigation phase.

(a) (b)

Fig. 1 The test bed used in this study. (a) The vehicle. (b) The PTZ camera.

In the learning phase, the user drives the vehicle to navigate in indoor environments and move to the front of concerned objects. The main recorded data include two categories, namely, path-related data and object-related ones. As soon as the learning process ends, all data are saved in a learning database, such that the learning process is only executed once and the data can be used repeatedly.

More specifically, while the vehicle navigates in an open space by the control of a user, it records the path data provided by the odometer, and denotes them as navigation nodes. When the vehicle arrives at the front of a concerned object, the user can control the PTZ camera to move toward the object and select the object in the image captured by the camera. Then, the features of the object are computed automatically from the images by performing the simplified SIFT. And the relative position between the vehicle and the monitored object is also computed automatically from the image subsequently. In such manners, the user can specify concerned objects continuously along the path until finishing a learning process. A navigation map, which consists of the path and the monitored objects data, is then created and saved into a text file for use in the navigation phase.

In the navigation stage, the vehicle moves sequentially from one node to another according to a selected path in the navigation map. When the vehicle reaches the next node, it checks first whether the node includes the monitored object data. If the node includes the monitored objects data, the vehicle uses the learned data to detect whether the object still exists or not. If the detection or matching process of the object fails, the system will issue an alarm message to the user. Otherwise, the vehicle uses the learned vehicle locations' data to adjust the vehicle's current location.

With such a navigation process, the vehicle can navigate alone the learned path to accomplish specified security patrolling works.

3. DETECTION OF MONITORED OBJECTS BY 2D OBJECT IMAGE MATCHING

While the vehicle patrols in the navigation phase, it stops in front of the monitored object by the use of learned path nodes. But the stop position at a monitored object may not be precise every time, mostly just close to the one recorded in the learning phase. This results in a slight change in the viewing angle of the monitored object from the camera. And the image of the same monitored object will be different in the scale, orientation, or position with respect to the one taken in the learning phase. Thus, a method with the ability to match corresponding objects in images taken with different camera poses and illuminations is needed.

In the past years, the scale invariant feature transform (SIFT) has been proven to be one of the most robust methods which use local invariant feature descriptors with respect to different geometrical changes [15]. In order to allow efficient matching between images, all images are represented as a set of vectors, called SIFT features. Each SIFT feature consists of local image measurements invariant to image translation, scaling, and rotation, and partially invariant to illumination and 3D viewpoint changes. In this study, we take advantage of the SIFT to match monitored object images and propose a simplified-SIFT which is faster than the original one, by reducing the difference of Gaussian scale layers to meet real-time security monitoring needs.

The time consumption of the process of the original SIFT algorithm can be divided into two parts: the processing time for feature localization and the processing time for feature descriptor generation. The first part is bounded by the size of the input image and the process layers specified by the number of intervals and octaves, and the second part is bounded by the number of features and the dimensions of each feature descriptor. In this study, the image captured by the camera of the proposed system is of a fixed resolution of 320×240 pixels. Hence, in the first part, we can only control the number of intervals and octaves to reduce the processing time. In the second part, because the number of features is uncontrollable, and the low dimension may result in unstable matching results, so we do not simplify the process of the feature descriptor generation.

For security monitoring, while the vehicle navigates to the monitoring node which is learned in the learning phase, the position of the monitored object will be close to the one found in the learning phase. Hence, the scale of the monitored-object image will not change too much. Therefore, while adapting the SIFT algorithm, we

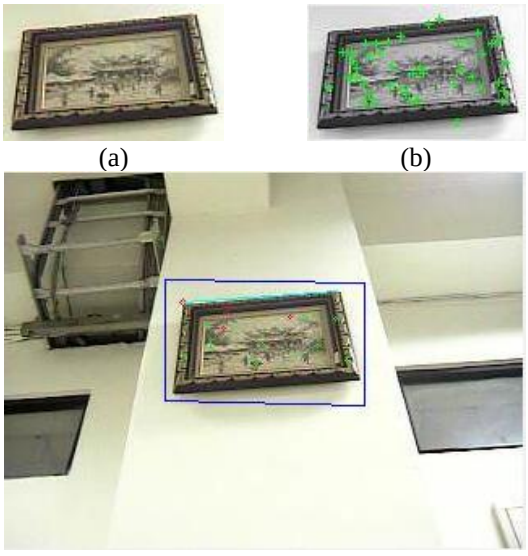
propose a simplified version by reducing the number of octaves to omit unnecessary process layers.

After the captured image of the monitored object is transformed into a set of simplified-SIFT features, we adopted a matching algorithm based on the Hough transform according to Lowe [14, 16]. For the given feature set, the best candidate match for each feature is firstly found by identifying its nearest neighbor in the other feature set. The nearest neighbor is defined as the feature with the closest Euclidean distance for the feature. After discarding the outliers, the Hough transform is used to identify the best subsets of matches. Let the given feature set which is found in the navigation phase be denoted as $\boldsymbol{F}_{navi}$ and the one which is learned in the learning phase be denoted as $\boldsymbol{F}_{learn}$. Each simplified-SIFT feature specifies 4 parameters: two coordinates of the feature location in the image, scale, and orientation. By applying the affine transform model, as shown in the following equation:

$$\begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \\ & \cdots & & \end{bmatrix} \times \begin{bmatrix} m \\ n \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \cdots \end{bmatrix},$$

where $m = s\cos\theta$, $n = s\sin\theta$, and (x, y) and (u,v) specify the locations of $\boldsymbol{F}_{navi}$ and $\boldsymbol{F}_{learn}$, respectively, the unknown similarity transform parameters between each match pair are collected as t_x, t_y, s, and θ by the following equations:

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



(a) (b)

(c)

Fig. 2 Experimental results of monitored object matching process. (a) Monitored object learned. (b) Locations of features marked as green crosses. (c) Successful matched result specified by the blue region.

A Hough transform entry is then created to predict the model location, orientation, and scale from the match hypothesis, and each feature votes for all poses that are consistent with the feature. Then, a peak cluster found in the Hough space is regarded to specify the best subsets of matches. An experiment result is shown in Fig. 2.

4. VEHICLE GUIDANCE BY LOCATION ESTIMATION BASED ON 2D OBJECT IMAGE MATCHING RESULTS

Let (X, Y, Z) denote the reference coordinate system (RCS). A horizontal line is given in the learning phase to specify X-axis of the RCS. A start point of the given horizontal line also specifies the origin R_0 of the RCS. Because the X-Y plane is parallel to the floor, we can treat the RCS as a *virtual* house corner. The X- and Y-axes specify the two perpendicular lines on the ceiling of the virtual house corner, as shown in the left-top of Fig. 3, and the Z-axis specifies the virtual line of the virtual house corner.

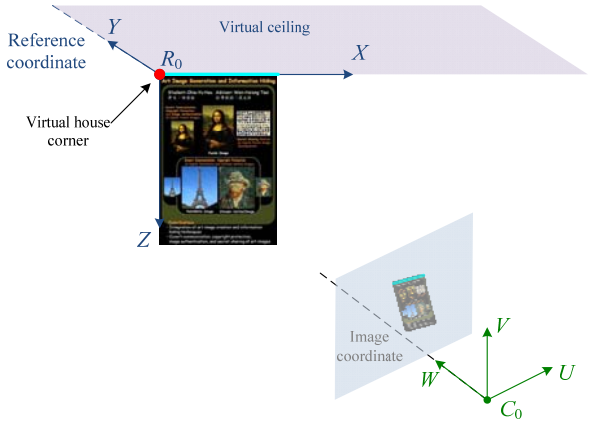


Fig. 3 A diagram of a virtual house corner specified by a given horizontal line (the cyan line on the top of the poster), and a start point (the red point on the left-top of the poster).

The equations of the edge line through the corner point in terms of image coordinates (u, v) are described by $u_p + bv_p + c = 0$. The desired vehicle location can be described by three position parameters X_c, Y_c, and Z_c and two direction parameters θ and ψ, where Z_c is the distance from the camera to the ceiling and is assumed to be known; θ is the pan angle between the optical direction of the camera and the Y-axis of the RCS; and ψ is the tilt angle of the optical direction of the camera with respect to the RCS and is also assumed to be known by solving the equation $\psi = 90^\circ - \phi$, where ϕ is the tilt angle provided by the PTZ camera. The five vehicle location parameters can be derived in terms of the two coefficients b and c of the edge line equation and the

start point (u_1, v_1) in the image taken by the camera. Finally the vehicle location can be estimated by computation of these parameters, as described in the following.

At first, we transform the reference coordinates into the camera coordinates. The transformation consists of four steps.

Step 1. Translate the origin ($-X_c$, $-Y_c$, $-Z_c$) of the RCS to the origin of the camera system in the following way:

$$T(X_c, Y_c, Z_c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_c & -Y_c & -Z_c & 1 \end{bmatrix}.$$

Step 2. Rotate the *X*-*Y* plane about the *Z*-axis through the pan angle θ using the following equation such that the *X*-*Y* plane is parallel to the *U*-*V* plane:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Step 3. Rotate the *Y*-*Z* plane about the *X*-axis through the tilt angle ψ using the following equation such that the *X*-*Y* plane is parallel to the *U*-*V* plane:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Step 4. Reverse the *Z*-axis using the following equation such that the positive direction of the *Z*-axis is identical to the negative direction of the *W*-axis:

$$F_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Let *P* be any point in the 3-D space with reference coordinates (*x*, *y*, *z*) and camera coordinates (*u*, *v*, *w*). Then the above coordinate transformation can be described as follows:

$$(u, v, w, 1) = (x, y, z, 1) \cdot T(X_c, Y_c, Z_c) \cdot R_z(\psi) \cdot R_x(\theta) \cdot F_z$$
$$= (x, y, z, 1) T_r$$

where

$$T_r = T(X_c, Y_c, Z_c) \cdot R_z(\theta) \cdot R_x(\psi) \cdot F_z$$
$$= \begin{bmatrix} \cos\theta & -\sin\theta\cos\psi & -\sin\theta\sin\psi & 0 \\ \sin\theta & \cos\theta\cos\psi & \cos\theta\sin\psi & 0 \\ 0 & \sin\psi & -\cos\psi & 0 \\ x_0 & x_0 & x_0 & 1 \end{bmatrix}$$

with

$$x_0 = -X_c \cos\theta - Y_c \sin\theta,$$
$$y_0 = (X_c \sin\theta - Y_c \cos\theta)\cos\psi - Z_c \cos\psi,$$
$$y_0 = (X_c \sin\theta - Y_c \cos\theta)\cos\psi + Z_c \cos\psi.$$

Let *P* be any point on the *X*-axis with reference coordinates (*x*, 0, 0). Then its camera coordinates (u_x, v_x, w_x) can be derived to be:

$$(u_x, v_x, w_x, 1) = (x, 0, 0, 1) \cdot T_r$$
$$= (x\cos\theta + x_0, -x\sin\theta\cos\psi + y_0, -x\sin\theta\sin\psi + z_0, 1).$$

Let (u_p, v_p) be the image coordinates of the projection of *P*. Then, according to the triangulation principle, we have the following two equations:

$$u_p = \frac{f \cdot u_x}{w_x}, v_p = \frac{f \cdot v_x}{w_x},$$

where *f* is the camera focus length.

Substituting the values of u_x and v_x above into the previous equation and eliminating the variable *x*, we can get the equation for the projection of the *X*-axis in the image plane in the following:

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

After substituting u_p into $u_p + bv_p + c = 0$, we obtain the following two equalities:

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

Then, we can use the two equations above to derive the variable θ. The two equations above can be transformed into the following two equations, respectively:

$$Z_c \cdot (-b\sin\theta + \cos\theta\cos\psi) = Y_c \sin\psi,$$
$$Z_c \cdot (-c\sin\theta + f\cos\theta\cos\psi) = -fY_c \cos\psi.$$

By eliminating Z_c and Y_c from above equations, we can get:

$$\tan\theta = \frac{f}{fb\cos\psi + c\sin\psi}.$$

Because the value ψ can obtain from the use of the tilt angle ϕ provided by the PTZ camera and the following equation:

$$\psi = 90^{\circ} - \phi ,$$

we can apply the above equations with known values ψ, θ, Z_c and the start point (u_1,v_1) to obtain the values θ, Y_c, and X_c.

After we compute the estimated vehicle location between the vehicle and the origin R_0 of the RCS, the next step is path correction by the use of the estimated results. The relation among the vehicle, the camera, and the RCS is illustrated in Fig. 4.

The direction angle of the vehicle can be derived by substituting the θ into the following equation:

$$\theta_v = 90° - (-\theta) + \theta_c$$

where the angle θ is negative because the angle of the clockwise rotation is positive and the X-Y plane is rotated through a pan angle $-\theta$ to be parallel to the image plane, and θ_c is the pan angle of the PTZ camera. As soon as the direction angle θ_v of the vehicle is obtained, we can compute the vehicle location in the RCS by substituting the angle θ_v and the distance between the camera and the center of the vehicle into the following equations:

$$X_v = X_c - D_c \cdot \cos\theta_v ,$$

$$Y_v = Y_c + D_c \cdot \sin\theta_v .$$

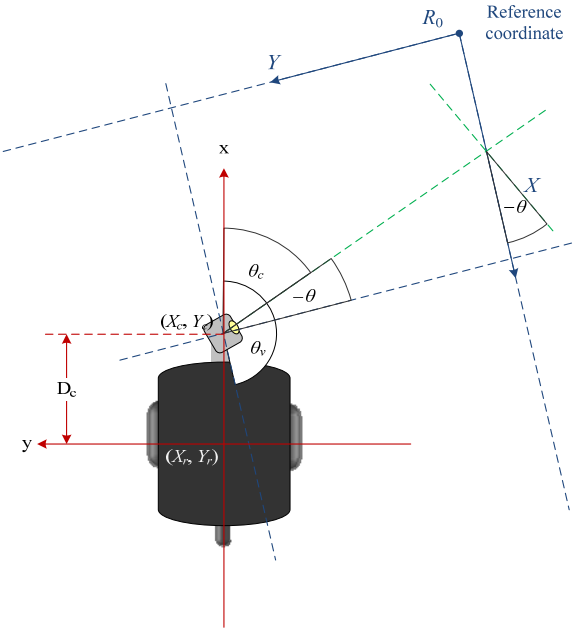


Fig. 4 Relation among the vehicle, the camera, and the RCS.

Finally, the location (X_v, Y_v) and the direction angle θ_v of the vehicle are acquired. If the vehicle is in the learning phase, these parameters are saved as the calibration information data. While the vehicle navigates in the navigation phase, we can utilize the parameters obtained above and the learned ones to correct the navigation path.

Let the learned location parameters including the location and the direction angle of the vehicle be denoted as $L(X_l, Y_l, \theta_l)$ in the RCS, and the estimated ones as $V(X_v, Y_v, \theta_v)$. Utilizing these parameters above and the corresponding learned path node (L_x, L_y) and the direction angle Θ_l of the vehicle at this path node in the GCS, we can compute the corrected location (N_x, N_y) and the adjustment angle θ_{adj} of the vehicle by transforming the relative location between (X_v, Y_v) and (X_l, Y_l) in the RCS into the GCS and computing the adjusting angle between θ_v and θ_l. The relation among the RCS, the VCS, and the GCS, and the corresponding angle is illustrated in Fig. 5. And an experiment result is shown in Fig. 6.

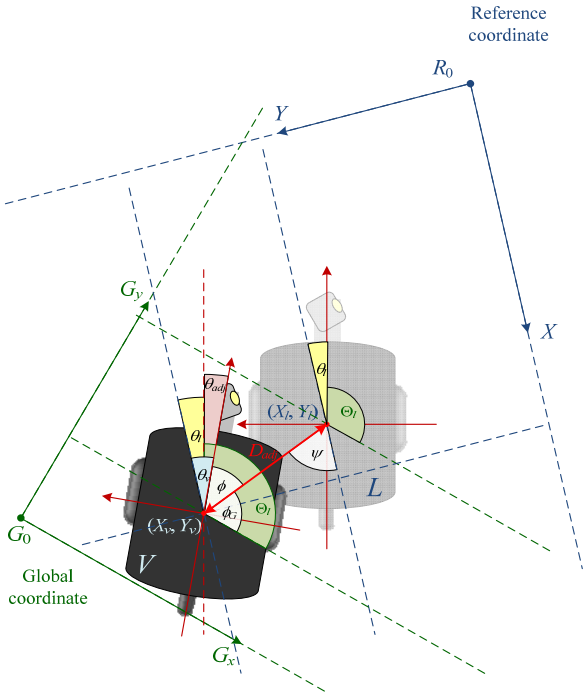


Fig. 5 Relation among reference coordinate system (RCS), vehicle coordinate system, and global coordinate system, and corresponding angle of vehicle. Learned vehicle location is denoted as a pastel vehicle with location (X_l, Y_l) in global coordinate system, and current vehicle location is denote as the colored vehicle with location (X_v, Y_v) in RCS.

In order to conduct experiments about the ability of path correction, we set up a navigation path including a monitoring node and a path node, as shown in Fig. 6. With the successively learned navigation path, we firstly put the vehicle at an identical start position to start the navigation. The experimental result shows that the vehicle navigated correctly on the navigation path, as shown in Figs. 7(a) and 7(b). And we also tested another case with artificial path deviations. We put the vehicle at a different position to simulate the condition that the vehicle navigates outside the learned path. The experimental result shows that the vehicle can self-correct the navigation path successfully by estimating the location with respect to the monitored object, as shown in Figs. 7(c) and 7(d).

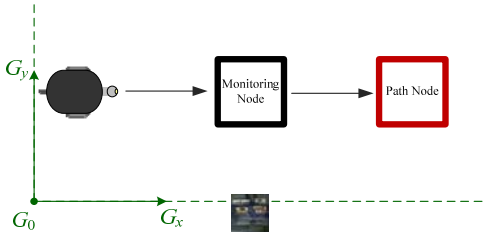


Fig. 6 Diagram of an experimental navigation path including a monitoring node and a path node.

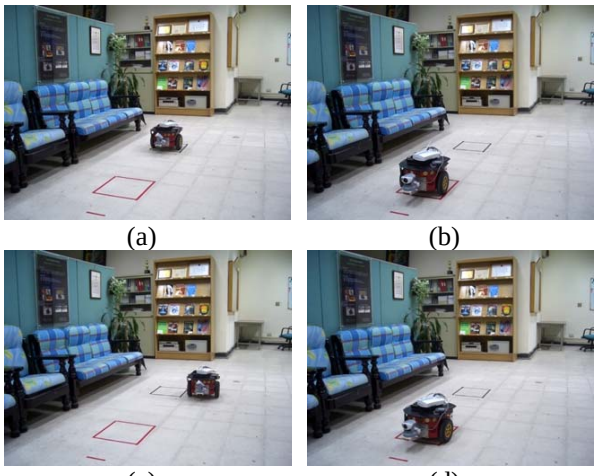


Fig. 7 Experimental result of path correction. (a) The vehicle arrived at a monitoring node and performed the matching and path correction. (b) The vehicle navigated to the next path node after correcting the navigation path. (c) The vehicle arrived at a wrong place which is a monitoring node, and performed the matching and path correction. (d) The vehicle navigated to the next path node after successfully matching the monitored object and correcting the navigation path.

5. EXPERIMENTAL RESULTS

Our test bed is an agile, versatile intelligent vehicle named Pioneer 3-DX made by MobileRobots Inc. At first, a user controls the vehicle to learn a path and some monitored objects on the walls. In this study, monitored objects are paintings and posters. Whenever the vehicle arrives at a spot, the user controls the system to record the monitored-object features and the calibration information. After the learning process, a navigation map is created. An illustration of the learned data, the navigation map, and the actual navigation path of one of our experiments is shown in Fig. 8.

The vehicle starts security patrolling according to the created map. The navigation process is shown in Fig. 8. Whenever the vehicle arrives at a learned monitoring node, it performs the security check of the existence of the monitored object. If the check is successful, the vehicle adjusts its location to continue its navigation on the right way according to the matching result; otherwise, a message is issued. For each monitored object shown in Fig. 8, the experimental results are shown in Fig. 9. The vehicle performed security monitoring to monitor 7 monitored objects. The vehicle arrived at the learned monitored node, as shown in Fig. 9(b). Then, it extracted the features of the image and matched them with the corresponding learned data. The matching results are shown in Fig. 9(c) and the learned monitored objects are shown in Fig. 9(d).

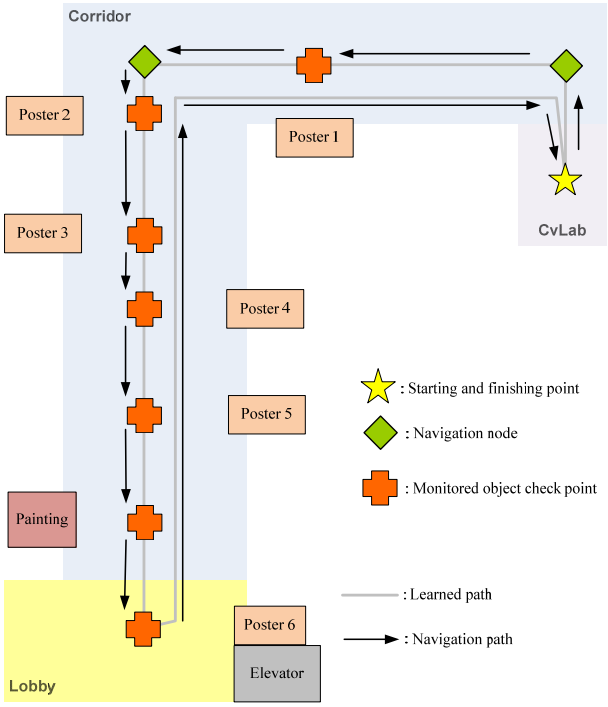


Fig. 8 An illustration of learned data and navigation map.

Fig. 9 Experimental results of object monitoring and navigation path correction. (a) Monitored object labels. (b) The vehicle monitors the monitored objects. (c) The matching result and the horizontal line used for path correction. (d) The image of learned monitored objects.

6. CONCLUSIONS

Several techniques and strategies have been proposed and integrated into an autonomous vehicle system for security patrolling in indoor environments with capabilities of specific-object monitoring and self-adjustment of navigation paths. Satisfactory navigation results have been obtained by this system.

At first, a security patrolling method by vehicle navigation with security monitoring capability has been proposed. The vehicle navigates according to the node data of the path map which is created in the learning phase and monitors the concerned objects by a 2D object image matching technique proposed in this study, the simplified-SIFT algorithm. Accordingly, we can extract the features of the monitored object from acquired images and match them with the learned data. The matching technique is based on the Hough transform. We construct a Hough transform histogram to predict the model location, orientation, and scale from the match hypothesis, and find the best match by finding the peak in the Hough space.

Next, a vehicle location estimation technique by utilizing the monitored object matching result has been proposed. The coefficients of the equation of a horizontal line and the location of the start point in the image are used to estimate the vehicle location. Also proposed is a path correction method, which compares the estimated location and the learned one to compute necessary path adjustment and transform it into the global coordinate system to correct the navigation path. The experimental results have revealed the feasibility and practicality of the proposed system.

REFERENCES

[1] G.N. DeSouza and A.C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No 2, pp. 237-267, February 2002.

[2] C. C. Lai, "A study on automatic indoor navigation techniques for vision-based mini-vehicle with off-line environment learning capability," *M. S. Thesis*, Dept. of Computer and Inform. Sci., National Chiao Tung Univ., Taiwan, June, 2003.

[3] K. L. Chiang, "Security patrolling and danger condition monitoring in indoor environments by vision-based autonomous vehicle navigation," *M. S. Thesis*, Institute of Computer Sci. and Eng., National Chiao Tung Univ., Taiwan, June, 2006.

[4] Y. C. Chen and W. H. Tsai, "Vision-based autonomous vehicle navigation in complicated room environments with collision avoidance capability by simple learning and fuzzy guidance techniques," *Proc. of 2004 Conf. on Computer Vision, Graphics and Image Processing,* Hualien, Taiwan, Aug. 2004.

[5] M. C. Chen and W. H. Tsai "Vision-based security patrolling in indoor environments using

autonomous vehicles," *Proc. of 2005 Conf. on Computer Vision, Graphics and Image Processing,* Taipei, Taiwan, Aug, 2005.

[6] C. Shmid and R. Mohr. "Local greyvalue invariants for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5, pp. 530-534, 1997.

[7] K. Mikoljczyk and C. Schmid. "Indexing based on scale-invariant features," *Proc. of International Conf. on Computer Vision*, pp. 525-531, 2001.

[8] I. Fukui, "TV image processing to determine the position of a robot vehicle," *Pattern Recognition,* Vol. 14, pp. 101-109, 1981.

[9] M. J. Magee and J. K. Aggarwal, "Determining the position of a robot using a single calibration object," *Proc. of IEEE Conf. on Robotics*, pp. 57-62, Atlanta, Georgia, May 1983.

[10] J. Huang, C. Zhao, Y. Ohtake, H. Li, and Q. Zhao, "Robot position identification using specially designed landmarks," *Proc. of 2006 IEEE Conf. on Instrumentation and Measurement Technology*, April 2006.

[11] H. L. Chou and W. H. Tsai "A new approach to robot location by house corners," *Pattern Recognition*, Vol. 19, pp. 439-451, 1986.

[12] K. L. Chiang and W. H. Tsai, "Vision-based autonomous vehicle guidance in indoor environments using odometer and house corner location information," *Proc. of 2006 IEEE International Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 415-418, Dec. 18 - 20, 2006.

[13] 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. on Industrial Electronics*, Vol. 48, No. 1, pp. 205-215, 2001.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[15] K. Mikolajczyk, and C. Schmid, "A performance evaluation of local descriptors," *Proc. of International Conf. on Computer Vision & Pattern Recognition*, Vol. 2, pp. 257-263, June 2003.

[16] D. G. Lowe, "Local feature view clustering for 3D object recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, pp. 682-688, Dec. 2001.