3D INDOOR VIRTUAL ENVIRONMENT MODELING VIA VEHICLE NAVIGATION AND MULTI-CAMERA IMAGE DATA FUSION

Hui-Ping Fan and Wen-Hsiang Tsai

Department of Computer & Information Science

National Chiao Tung University, Hsinchu, Taiwan, R.O.C

Email:whtsai@cis.nctu.edu.tw

Abstract

In applications of virtual reality, it is often required to establish environment models, which is usually time-consuming. In this study, an offline automatic approach to 3-D indoor virtual environment modeling by the way of autonomous land vehicle (ALV) navigation is proposed. An ALV is first driven manually along a path by a driver to collect environment data using three cameras mounted on the ALV. The system then generates the environment model by a data fusion approach using the model learning result from the multiple-view data, performs intelligent post-processing works to create a more complete model, and uses the VRML language to construct a corresponding 3-D virtual reality (VR) model automatically. Good experimental results prove the feasibility of the proposed approach.

Keyword: Vehicle Navigation, Data Fusion, Multi-Camera, Indoor Environment Modeling, Computer Vision, Virtual Reality

1. Introduction

Autonomous land vehicles (ALV's) have attracted intensive research effort in recent years because of its versatile applications. With recent development of computer vision and image processing techniques, more applications of ALV's have become feasible. Many applications of ALV navigation in environments require a dependable model of the world. However, establishment of environment models is really a time-consuming work. Also, recent interests in virtual reality and multimedia have provided a great impetus to the development of automatic techniques for building graphical models and environments by sensing the real world. Learning of models in these ways is essential, particularly in terms of production times and attaining high fidelity. Many application researches have developed techniques for modeling environments using a variety of sensors and a spectrum of techniques. Illingworth and Hilton [1] introduced the principles and methodologies to build a 3-D model world using a variety of sensors and a spectrum of techniques. For 3-D indoor environment modeling, Lebègue and Aggarwal [2,3,4,5] developed an integrated system to generate architectural CAD models using a mobile robot. The system consists of a segment detector, a tracker, and a CAD modeler optimized for environments with prominent 3-D orientations. Takeshi Shakunaga [6] proposed a method by which specific corridor models can be recovered from a single image made by a well-calibrated camera. The recovery is based on a generic corridor model that covers a wide variety of corridors. Also, Egazzar, et al. [7] investigated modeling of indoor environments using a low-cost, compact, active-range camera, known as BIRIS, mounted onto a pan and tilt motor unit. Pan and Tsai [8] proposed an integrated approach to automatic model learning and path generation for vision-based ALV guidance in building corridors. In Chen and Tsai [9], an incremental-learning-by navigation approach to vision-based ALV guidance in indoor environments was proposed. The proposed learning approach is reliable because of the robustness of the use of the proposed MWGHT (Multi-Weighted Generalized Hough Transform) matching scheme. And the learned environment model can be updated after each navigation session. A new approach to vision-based unsupervised learning of unexplored indoor environments for ALV navigation was also proposed by Chen and Tsai [10]. The ALV can conduct automatic learning of navigation environments without human involvement.

2. Overview of Proposed System

2.1. System Configuration

An ALV with smart, compact, ridable characteristics, as shown in Figure 2.1 is used as a test bed for this study. It is a commercial motor-driven vehicle modified by adding sensors, electronic controls, an on-board PC, and several power conversion equipments. The ALV can be switched between the manual operation mode or the computer operation mode by the user.

Figure 2.1 The ALV used in the experiments.

In the experiments of this study, we use three cameras to take images with a resolution of 640×480 elements. These cameras are mounted at fixed positions on the cross-shaped racks of the ALV. More specifically, there are two cameras on a crossbar. One camera is attached to the left position and the other is attached to the right position, of the crossbar. The third camera is on another crossbar and is attached to the middle position of that crossbar. The lower two cameras are used to grab images

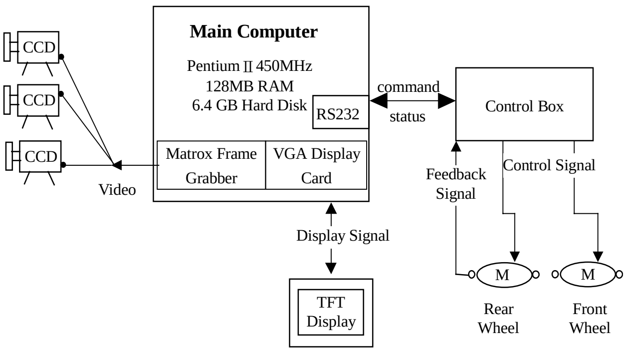


Figure 2.2 System structure

of the left and the right baselines of the wall in the building corridor, and the upper one is used to grab images of the ceiling of the corridor.

The ALV is computer-controlled with a modular architecture, as shown in Figure 2.2, including four major components, namely, a vision system, a central processing unit (the Intel Pentium II 450MHz PC mentioned above), a motor control system, and a DC power system. The vision system consists of the three cameras, a color monitor, and an image frame grabber. The motor control system consists of a main control box with the controller and the motor driver mentioned previously, and two motors.

2.2. Brief Description of Proposed System

The goal of this study is to construct an appropriate 3-D model of the indoor environment using the images taken by three cameras mounted on an AVL. The 3-D model data may be used to build a virtual reality environment for many applications (e.g., navigation, visualization, and so on). The proposed approach consists roughly of three stages. The first stage is initial learning, in which the ALV is driven manually along a path decided by a driver. Three images are captured simultaneously from the three cameras and the control status data are recorded in each navigation cycle. Then, a certain off-line procedure is performed to construct the initial model. This is accomplished by calculating the relations between the ALV and the environment features observed in each learning cycle, and by matching the features with the partially learned model, followed by the step of fusing the processed data from the three cameras. The second stage is to refine the observed 3-D raw data produced by imperfect matching and image processing techniques using data extracted from limited camera views. The third stage is to connect different corridors to complete the model by performing a line-pattern matching algorithm and to build the 3-D VR environment model, using the refined 3-D data.

2.3. Procedure of System Operations

The system operations are based on a hierarchical approach, which include the following steps:

(1) Perform camera calibration for each camera.

(2) Grab an image from each of the three cameras and save them as an initial model at the initial location.

(3) Drive the ALV manually from the initial location and grab three images of the current environment scene.

(4) Record the environment images, the counter values of

the odometer, and the turn angles of the front wheels.

(5) Manually drive the ALV a certain distance and make appropriate turns to keep it in the middle of the corridor.

(6) If the ALV reaches the destination, go to (8) to perform off-line processing; else, go to (3) for the next cycle.

(7) Perform image processing.

(8) Perform model learning.

(9) Generate the overall 2-D environment model.

(10) Construct a corresponding 3-D VR model.

3. Detection of Environment Features for Model Learning

3.1. Introduction

Selecting stable environment features and developing effective methods to extract these features are the most important keys to successful model learning. In this study, the selected environment features come from the baselines and ceiling information in building environments. Some advantages of selecting features from these sources are listed in the following:

Baselines: baselines are abundant and easily visible in buildings**;**

Ceiling information: ceiling information are seldom disturbed and usually are with uniform patterns.

In the proposed system, computer vision techniques are employed to locate environment features. At first, visual features are found by image processing techniques. Next, the locations of the features are calculated by computer vision techniques. At last, a model-matching algorithm for line segments and corners is proposed to find the correspondence between the sensed local model and the learned global model. The matching results then are used to locate the ALV and construct the environment model.

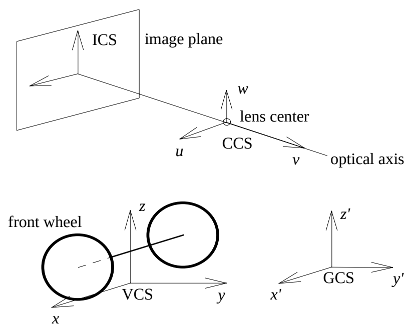


Figure 3.1 Coordinate Systems.

3.2. Coordinate System Transformation

Four coordinate systems and coordinate transformations are defined here for use in the following sections. They include the camera coordinate system (CCS), the image coordinate system (ICS), the vehicle coordinate system (VCS), and the global coordinate system (GCS). These coordinate systems are shown in Figure 3.1. Since the origins of the ICS, CCS, and VCS are attached to some points on the ALV, the ICS, CCS, and VCS are moving with the vehicle during navigation. On the contrary, the GCS is fixed all the time, and is defined to be coincident with the VCS when the ALV is at the starting position in the initial model learning stage.

The location of the vehicle can be assured once the relation between the VCS and the GCS is found. Since the vehicle is on the ground all the time, the z-axis and the z'-axis can be ignored.

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

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{bmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x'_p & y'_p & 1 \end{bmatrix}$$

where $\left(x'_p, y'_p\right)$ is the translation vector from the origin of x'-y' to the origin of x-y and ω is the relative rotation angle of x-y with respect to x'-y', as shown in Figure 3.2. The translation vector $\left(x'_p, y'_p\right)$ and the rotation angle ω of the ALV in the x'-y' coordinate system determine the position and the direction of the vehicle in the GCS, respectively.

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

$$\begin{pmatrix} u & v & w & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_d & -y_d & -z_d & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ where

$$\begin{aligned}
r_{11} &= \cos\theta\cos\varphi + \sin\theta\sin\phi\sin\varphi, \\
r_{12} &= -\sin\theta\cos\phi, \\
r_{13} &= \sin\theta\sin\phi\cos\varphi - \cos\theta\sin\varphi, \\
r_{21} &= \sin\theta\cos\varphi - \cos\theta\sin\phi\sin\varphi, \\
r_{22} &= \cos\theta\cos\phi, \\
r_{23} &= -\cos\theta\sin\phi\cos\varphi - \sin\theta\sin\varphi,
\end{aligned}$$

$$r_{31} = \cos\phi \sin\varphi,$$
$$r_{32} = \sin\phi,$$
$$r_{33} = \cos\phi \cos\varphi,$$

and θ is the pan angle, ϕ the tilt angle, and φ the swing angle, of the camera with respect to the VCS; and (x_d, y_d, z_d) is the translation vector from the origin of the CCS to the origin of the VCS. The values of θ, ϕ, φ, x_d, y_d, and z_d are measured by performing camera calibration.

3.3. Locating Environment Features

One selected environment feature is the baseline on the building wall. On the abstract level, these visual features can be categorized into two classes, straight lines and corners of line segments. In this section, the geometric properties of the baseline, whose height is fixed and known in advance, are used to calculate the VCS coordinates of the baseline segments detected in an input image. The method for calculating the VCS coordinates of a corner point, which is located on a baseline, is described in the following.

As shown in Figure3.3, the height of the baseline is known in advance, after back-projecting a point P in the image into the VCS, we can get P'.

Another type of selected environment feature comes from the boundary information of ceiling lamps. Note that just one set of parallel lines on the ceiling in the acquired image is utilized. The reason is that we view a ceiling lamp as a rectangle, so one set of parallel lines are enough to describe the characteristics of it. In this study, we select the left and the right boundaries of the ceiling lamp as our features. The way we employ for calculating the VCS coordinates of the ceiling lamp information, including the line segments and the corners, is similar to the method described above. In addition, the height between the floor and the ceiling is known in advance.

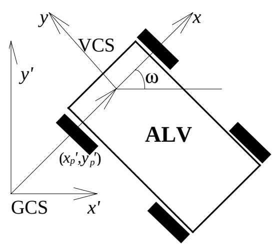


Figure 3.2 The relation between 2D coordinate systems x-y and x'-y' represented by a translation vector (x'_p, y'_p) and a rotation angle ω.

4. Matching Features with Learned Model

There are several existing algorithms for line-pattern and point-pattern matching. For example, the Generalized Hough Transform (GHT) is a popular approach to arbitrary pattern matching. It also works for this study. However, the GHT is time consuming. Thus it is desired to use a faster and simpler matching method for applications. Another problem arises when computer vision inaccuracy and image processing errors are involved. This makes perfect matching impossible, so a fault-tolerant matching algorithm is required. Furthermore, sometimes the newly detected features in the local model may not exist in the learned global model, so the algorithm should also be capable of partial matching.

There is a basic assumption for the experimental environment: the objects, namely, the walls and doors, and the ceiling lamps, in the environment are all in two orthogonal directions. With this assumption, the environment features can be treated as a set of orthogonal line segments. In this study, we use two matching algorithms, one for line-segment matching [8] and the other for corner-matching [10] to meet the above three requirements.

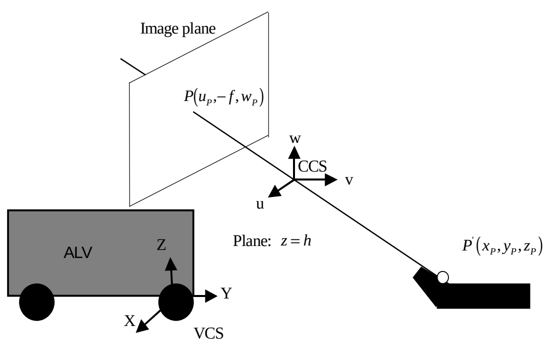


Figure 3.3 Configuration of the system for finding the back-projection point for an image pixel.

5. Side-View Model Learning

The goal of the proposed model learning method in this study is to construct environment models from collected data automatically. Because of the inaccuracy of control actions, the estimated location of an ALV is not very accurate. Correcting the position of ALV is a necessary work for building an accurate environment model. This can avoid error accumulation in a long learning process. As shown in Figure 3.2, the ALV location is described by the ALV slant angle ω and the ALV position (x, y). We first correct the slant angle of the ALV. Then a model-matching approach is used to correct the position of the ALV. The learning procedure is described as follow.

Step 1. Set the initial global model as empty.

Step 2. Extract environment features from the captured image.

Step 3.Calculate the estimated position and orientation of the ALV by the control data [12]. Calculate VCS coordinates of the extracted environment features.

Step 4. Adjust the turn angle of the ALV by input features and re-compute ALV location.

Step 5. Calculate the GCS of the detected environment features by the re-computed ALV location.

Step 6. Set up a local model by collecting the data of the local features computed in Step 6.

Step 7. If the global model is non-empty, match the local model with the global model by using the line-segment or corner-matching scheme; else, go to Step 10.

Step 8. Recalculate the accurate ALV location. Then recalculate the accurate position of the local features by the matching result and the recalculated ALV location.

Step 9. Attach the local model to the global model.

Step 10. Repeat Steps 2 through 10 for the next learning cycle.

6. Data Fusion for Generating Overall Environment Model

6.1. Generating Environment Model Using Multiple Image

At each model learning cycle, there are three sets of independent image data grabbed from the three cameras at the same instance. From each image data, there yields a position vector (xt, yt, ω) for computing an accurate ALV location after performing the side-view model learning procedure described in section 5. Ideally, all the three displacements must be identical. Unfortunately, because of image processing and vision computation errors, they might be different. This is undesirable. Therefore, we have to merge the model learning results from the three cameras for building the top view of the environment model. The data-fusion idea for generating

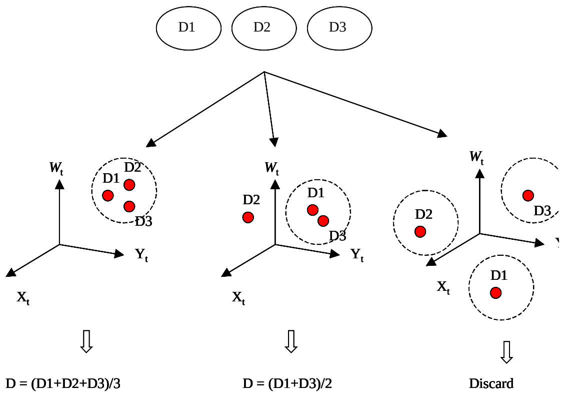


Figure 6.1 Decide a universal position vector.

the environment model using the learning result from the multiple data is shown in Figure 6.1.

In each learning cycle, three sets of image data are collected, and each yields a position vector (xt, yt, ω) for computing an accurate ALV location after performing the matching step in the side-view model learning procedure. The three displacements, D1, D2, and D3, presumable should be identical. But due to image processing and vision computation errors, they might be different, and it is undesirable. We propose here a method to decide the single displacement, D, from the multiple-view data for use in all the three side views (the two corridor sides and the corridor ceiling). The distance "*Dis*" between two position vectors, ($x1$, $y1$, ω_1) and ($x2$, $y2$, ω_1), is defined as following.

$$Dis = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (\omega_1 - \omega_2)^2}$$

As shown in Figure 6.1, if any two displacements of the three are separated by a distance which is smaller than a pre-selected value T, then we decide that the three position vectors are similar enough. Then, we use their average as the desired single position vector. If two of these displacements whose distance is closed enough (<T) and the third one is far away from the two, then we use

the average of the two similar position vectors as the desired vector. Otherwise, if the three position vectors are all mutually dissimilar, i.e., if the distance between any two of them is larger than a pre-selected value T, then we discard all of them, ignore the result of this learning cycle, and go to the next learning cycle.

6.2. Restoration of 2-D Environment Model

As shown in Figure 6.2, some features cannot be modeled accurately by the ALV because of the limit viewed of the cameras. In the example, each corner is marked by a small black spot, and each unseen feature is marked by a dash line. The ALV cannot observe the corner C, and the two neighboring lines of C are also hidden partially. Therefore, restoring these unseen features is necessary to make the acquired 2-D environment model more completely.

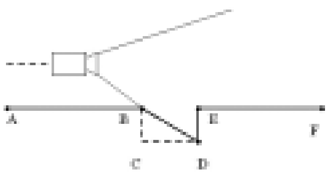


Figure 6.2: An example of senses out of sight.

6.3. Merge of Data of Distinct Corridor Sections

If more than one corridor exists in the indoor environment, or more specifically if there exist crossings in the environment, the proposed method, which performs model learning for each corridor separately, is insufficient. Instead, merging of data of distinct corridor sections is necessary for constructing a complete environment model. An assumption made in this study is that for any two neighboring corridor sections there must exist a certain overlapping area. This assumption is reasonable because data collection is involved by human beings. After traveling through the entire environment, the relation between two corridors is known. To merge the two sections, we can find the corresponding points of the two corridors. Using the relation between two corresponding points, we can merge two distinct corridor sections.

7. Construction of 3-D Indoor VR Environment Model

The final step of this study is to process 3-D data building a VR environment model. The result is composed of a set of planar surfaces, represented as polygons, in 3-D space. These polygons are defined using the virtual reality markup language (VRML). To view a VRML document, a plug-in application should be installed to the browser. This plug-in application allow users to access VRML worlds with their current browser technologies. Our program will generate VRML document automatically depending on the 3-D environment data acquired by the method described in section 6. The result can be used for various applications such as ALV navigation, exhibit houses on world wild web, in which users can have a better feeling of involvement in the VR environment.

8. Experimental Results

The experiments of ALV model learning and 3-D VR environment reconstruction were performed in the corridor of a building in National Chiao Tung University.

Figure 8.1(a) shows a top view of real corridor data acquired by the model learning scheme. Figure 8.2 (b) shows restoration of a corridor data of Figure 8.1(b). Figure 8.2 is an illustrative example of corridor section merging. The corresponding points of the two corridors are drawn with points. Figure 8.3 is a complete model of Figure 8.2(f). Figure 8.4 and Figure 8.5 shows a result of 3-D indoor VR model reconstruction of Figure 8.3.

9. Conclusions

In this paper, we have proposed a system for learning environment models and reconstructing corresponding 3-D VR models by ALV navigation and computer vision techniques. Certain assumptions about the scene structure are utilized to reduce the complexity of the system. The system not only collects the information of the environment features to build up a top-view environment model but also reconstructs a corresponding 3-D VR environment model by performing some post-processing works to the observed 3-D data. For environment learning, an algorithm for orthogonal-line-segment and corner-pattern matching and a systematical algorithm for constructing the learned environment model by multi-camera image data fusion have been utilized to improve the system performance. Because of the limit of the camera view and defective matching results, a scheme for restoring the observed 3-D data has also been proposed. Furthermore, a scheme to reconstruct 3-D VR environment models by using the virtual reality markup language (VRML) has been proposed. The proposed learning and reconstruction system has been implemented on a prototype ALV. Successful reconstruction of 3-D VR environment models in indoor corridor environments confirms the feasibility of the proposed approach.

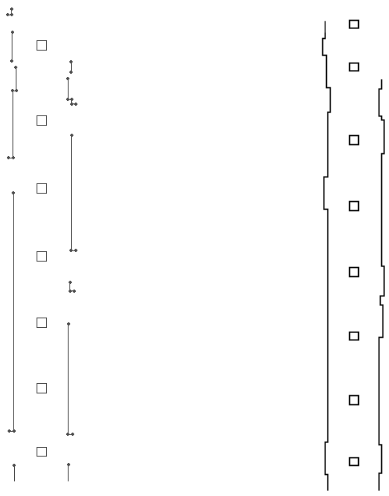
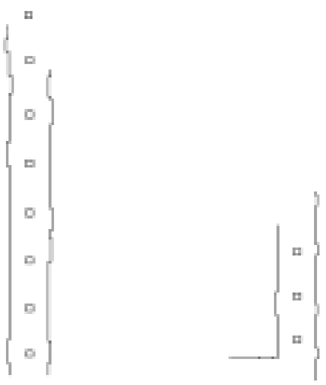
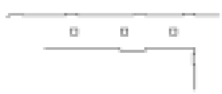


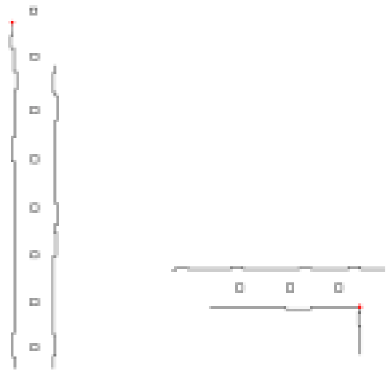
Figure 8.1: An example of restoration of a corridor in real environment. (a) Before restoration. (b) After restoration. The corners are marked by a small black spot.



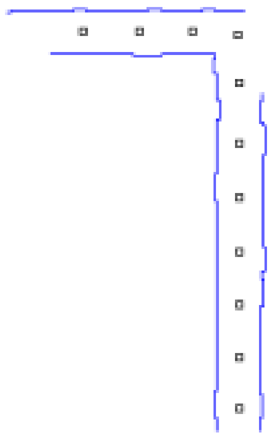
(a) (b)



(c)



(d) (e)



(f)

Figure 8.2 An illustrative example of corridor section merging. (a) One corridor model. (b) A neighboring corridor model of (a). (c) Translate view point of (b). (d) and (e) Corresponding point in (a) and (c). (f) Result of merge.

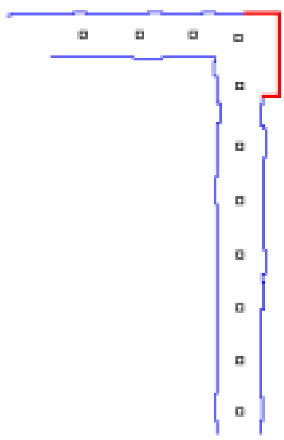


Figure 8.3 A complete model of Figure 8.2(f) which is the result of restoration. The border line are added to made the model more complete

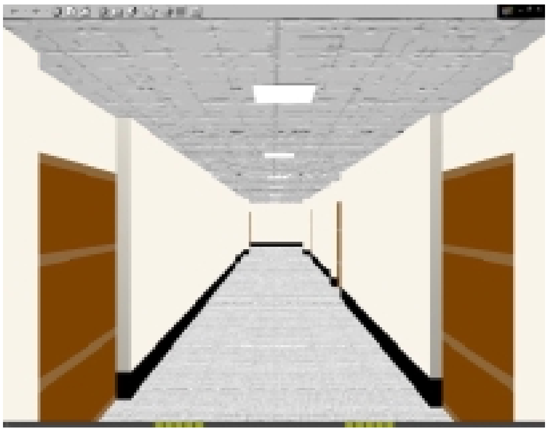


Fig. 8-4 The reconstructed 3D-VR Model.

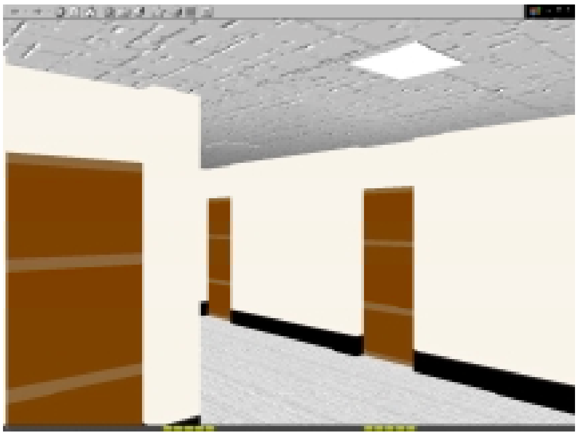


Figure 8.5: 3-D VR model of Figure 8.4 from different viewpoint.

References

[1] J. Illingworth and A. Hiton, “Looking to build a model world: automatic construction of static object models using computer vision,” *Journal of Electronics& Communication Engineering*, June 1998.

[2] X. Lebègue and J. K. Aggarwal, “Extraction and Interpretation of Semantically Significant Line Segments for a Mobile Robot,” *Proc. of 1992 IEEE Intl. Conf. on Robotics and Automation*, Nice, France, pp. 1778-1785, May 1992.

[3] X. Lebègue and J. K. Aggarwal, “Significant Line Segments for an Indoor Mobile Robot,” *Proc. of 1993 IEEE Transactions on Robotics and Automation. Vol. 9, No. 6,* pp 801-815, December 1993.

[4] X. Lebègue and J. K. Aggarwal, “Generation of Architectural CAD Models Using a Mobile Robot,” P*roc. of 1994 IEEE Intl. Conf. on Robotics and Automation, San Diego, California, U.S.A., Vol. 1*, pp. 711-717, May 1994.

[5] X. Lebègue and J. K. Aggarwal, “Automatic Creation of Architectural CAD models,” *CAD-Based Vision Workshop*, 1994., Proceedings of the 1994 Second, page(s) 82-89.

[6] Takeshi Shakunaga “3-D Corridor Scene Modeling from a Single View under Natural Lighting Conditions” *IEEE Transactions on Pattern Analysis and machine intelligence*, Vol. 14, No. 2, pp. 293-298, February 1992.

[7] Shadia Elgazzar, Ramiro Lisscano, François Blais, and Andrew Miles, “ Active Range Sensing for Indoor Environment Modeling” , *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 1, pp. 260-264, February 1998.

[8] F. M. Pan and W. H. Tsai, “Automatic environment learning and path generation for indoor autonomous land vehicle guidance using computer vision techniques” , *Proc. of 1993 National Computer Symposium*, Chia-Yi, Taiwan, Republic of China, pp. 311-321, 1993.

[9] Guan-Yu Chen and W. H. Tsai, “An Incremental-Learning-by-Navigation Approach to Vision-Based Autonomous Land Vehicle Guidance in Indoor Environments Using Vertical Line Information and Multi-Weighted Generalized Hough Transform Technique” , *IEEE Transaction on System, Man and Cybernetics-Part B: Cybernetics*, Vol. 28, No. 5,pp. 740-748, October 1998.

[10] Guan-Yu and W. H. Tsai, “A New approach to vision-based unsupervised learning of unexplored indoor environment for autonomous land vehicle navigation” , *Robotics and Computer Integrated Manufacturing 15 (1999)* pp. 353-364.

[11] Zhi-Fang Yang and W. H. Tsai, “Viewing Corridors as Right Parallelepipeds for Vision-Based Vehicle Location” , *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 3, June 1999.

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