國立交通大學

多媒體工程研究所

碩士論文

利用搭載雙環場攝影機取像系統之自動車作自動建立房間立體空間圖之研究

A Study on Automatic 3-D House-layout Construction by a Vision-based Autonomous Vehicle Using a Two-camera Omni-directional Imaging System

研 究 生：尤柏智

指導教授：蔡文祥　教授

中華民國九十九年六月

利用搭載雙環場攝影機取像系統之自動車作自動建立房間立體空間圖之研究

A Study on Automatic 3-D House-layout Construction by a Vision-based Autonomous Vehicle Using a Two-camera Omni-directional Imaging System

研究生：尤柏智　　Student: Bo-Jhih You

指導教授：蔡文祥　　Advisor: Prof. Wen-Hsiang Tsai

國立交通大學
多媒體工程研究所
碩士論文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
For the Degree of
Master
In

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

利用搭載雙環場攝影機取像系統之自動車作自動建立房間立體空間圖之研究

研究生：尤柏智　　　　　　　　指導教授：蔡文祥 博士

國立交通大學多媒體工程研究所

摘要

本研究提出了一個利用有視覺自動車導航作自動建立房間立體空間圖的系統，整個過程無須人為指引。在假設空房間擁有彼此垂直的牆面以及像是門和窗戶的扁平物件的條件之下，此系統可用來獲取空房間的平面空間圖。為了取得環境影像，我們設計了一種新型的環場攝影機取像系統，該系統包含兩個同光軸且背對背連接的環場攝影機，可分別取得環境中上半及下半圓球之影像。我們採用空間對應表與取像系統所取得之環場影像去計算空間特徵點的三維深度資料。我們所提出房間立體空間圖的建立過程包含三個階段，分別為(1)跟隨踢腳進行自動車導航，(2)平面空間圖之建立，以及(3)房間立體空間圖之展示。在第一階段，我們透過跟隨房屋中的踢腳進行自動車導航，直接在環場影像進行影像處理以獲取影像中的踢腳點，再透過空間對應表的查詢，計算出踢腳點的空間位置。基於最小平方法，我們進行圖形分類，將這些踢腳點分成兩組互相垂直的集合，並使用這些踢腳邊線來表示其所在的牆壁。在第二階段，我們提出一個總體最佳化的方法，用以建立平面空間圖，此方法調整所有踢腳邊線，使整體的直線鑲合(line fitting)誤差最小。在最後階段，我們以離線作業方式對自動車航行時蒐集的環場影像進行門與窗戶的偵測。我們將這些在門與窗戶偵測所得到的資料進行合併，且結合踢腳邊線作為房屋架構的三維空間資料，最後以圖學形式展示整個房屋架構。實驗結果顯示我們所建立的平面空間圖是相當精確的，其近似平均誤差為 3%，而自動車可多次成功的進行航行並建立房屋立體空間圖。這些結果證明了本系統的可行性。

A Study on Automatic 3-D House-layout Construction by a Vision-based Autonomous Vehicle Using a Two-camera Omni-directional Imaging System

Student: Bo-Jhih You Advisor: Prof. Wen-Hsiang Tsai

Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

An automatic house-layout construction system via vision-based autonomous vehicle navigation without human involvement is proposed. The system can be used to acquire the floor layout of an empty room, which consists of mutually perpendicular wall lines as well as shapes of flat objects like doors and windows.

To achieve acquisition of environment images, a new type of omni-directional camera is designed, which consists of two omni-cameras aligned coaxially and back to back, with the upper and lower cameras taking respectively images of the upper and lower semi-spherical spaces of the environment. The pano-mapping approach is adopted to compute 3-D depth data of space feature points using the upper and lower omni-images taken by the proposed imaging system.

The proposed automatic house layout construction process consists of three stages, namely, vehicle navigation by mopboard following, floor layout construction, and 3-D graphic house layout display. In the first stage, vehicle navigation is conducted to follow the wall mopboards in a house. An image processing scheme is applied directly on the omni-image to extract mopboard points. Then, a pattern classification technique is proposed to classify the points in two perpendicular groups, which are then fitted by lines using an LSE criterion. Such mopboard edge lines are used to represent the walls including the mopboards.

In the second stage, a global optimization method is proposed to construct a floor layout from all the mopboard edge lines in a sense of minimizing the total line fitting error. And in the last stage, doors and windows are detected in an offline fashion from the omni-images taken in the navigation session. The data of the detected door and window then are merged into the mopboard edge data to get a complete 3-D data set as the house structure, which finally is transformed into a graphic form for 3-D display from any viewpoint.

The experimental results show that the constructed floor layout is precise enough with an approximate average error of 3%, and that automatic vehicle navigation may be repeated successfully to construct graphic house layouts. These facts show the feasibility of the proposed approach.

ACKNOWLEDGEMENTS

The author is in hearty appreciation of the continuous guidance, discussions, support, and encouragement received from my advisor, Dr. Wen-Hsiang Tsai, not only in the development of this thesis, but also in every aspect of his personal growth.

Thanks are due to Mr. Chih-Jen Wu, Mr. Che-Wei Lee, Mr. Guo-Feng Yang, Mr. Yi-Fu Chen, Mr. Jheng-Kuei Huang, Miss Pei-Hsuan Yuan, Mr. Chih-Hsien Yao, Miss Mei-Hua Ho, and Miss I-Jen Lai for their valuable discussions, suggestions, and encouragement. Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during his thesis study.

Finally, the author also extends his profound thanks to his family for their lasting love, care, and encouragement. The author dedicates this dissertation to his beloved mother.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Motivation

The study of autonomous vehicles or mobile robots has become one of the most popular research topics in the world in recent years. Related applications of the autonomous vehicle have been developed intensively by many researchers to help human beings in various areas of automation. Among these applications, vision-based vehicles are especially useful in human environments for carrying out routine but important works like security patrolling, unknown environment navigation, dangerous condition exploration, etc.

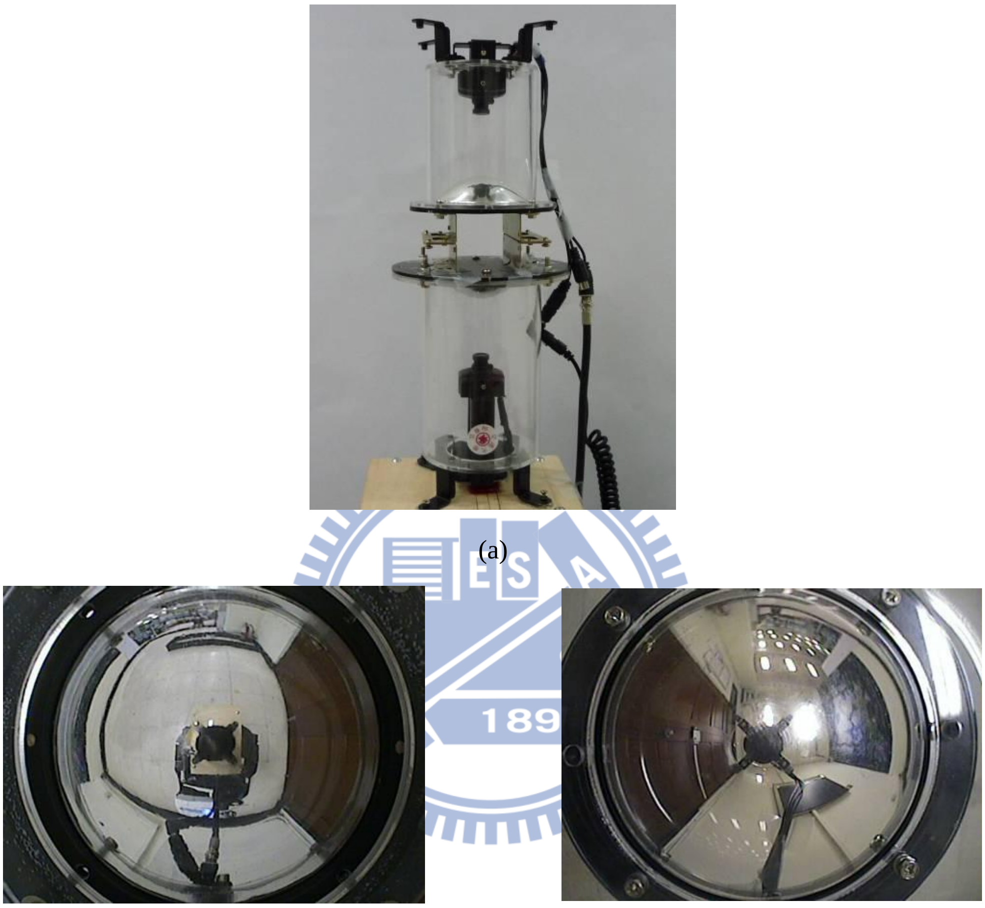
There are many empty pre-owned houses with uncertain interior space configurations, and a house agent may want to obtain the layout of each empty room in a house in advance before getting the house ready for sale. It is inconvenient and sometimes dangerous for the house agent to measure the house layout line by line and room by room *by hand*. Besides, this task usually includes measurement of the positions of doors and windows in rooms, which might be so numerous that the task becomes too time-consuming to be endurable. A possible way out is to conduct the task *automatically* without human involvement!

In order to achieve this goal, the use of a vision-based autonomous vehicle system is a good choice. In this study, it is desired to design an autonomous vehicle to possess an ability to navigate by *mopboard following* in an unknown empty room space and acquire automatically during navigation the information of the room layout and the structures (positions and heights) of the doors and windows in the room. This is feasible because most houses have mopboards at the roots of the walls. Use of the mopboard for vehicle guidance also simplifies the design of the vehicle navigation work for solving the above-mentioned *automatic house-layout construction* problem.

In addition, if we equip the vehicle with a traditional projective camera which has relatively narrow fields of view (FOV's), unseen areas by the camera often appear. This is not the case if an *omni-directional camera* (or simply, *omni-camera*) is used because the FOV of an omni-camera is much larger, covering a half sphere of the 3-D space around the camera. Therefore, it is desired in this study to equip the vehicle with omni-cameras to take wider views of house spaces for automatic house-layout construction as described above.

Moreover, to obtain 3-D information of the room space which is required for the above purpose, we use in this study two catadioptric omni-cameras attached together in a bottom-to-bottom fashion, forming a two-camera omni-directional imaging system, as illustrated in Figure 1.1 (a). The two omni-cameras cover respectively the upper and lower semi-spherical FOV's simultaneously around the autonomous vehicle, corresponding to the floor and the ceiling scenes, as shown in Figures 1.1(b) and 1.1(c). In this way, much more room space information can be acquired at a time without moving the cameras around the entire room space at many spots, which is necessary when a traditional projective camera is used, to get a full coverage of the 3-D space of the room.

When a vehicle navigates by mopboard following in a room space, the mopboards used for vehicle guidance may belong to different adjacent walls, and so the vehicle is likely to get confused in deciding which wall should include an extracted mopboard edge point. As a result, the vehicle is unable to know the distances between walls and itself in its moving direction. By classifying the mopboard edge data as done in this study, this problem can be solved. Consequently, the vehicle not only can make appropriate control of itself, but also can gather desired environment space data correctly.



(a)

(b)

(c)

Figure 1.1 Two catadioptric omni-cameras connected in a bottom-to-bottom fashion into a single two-camera omni-directional imaging system used in this study. (a) The two-camera imaging system. (b) An acquired image of the floor using the lower camera. (c) An acquired image of the ceiling using the upper camera.

After a navigation session is completed, it is desired to construct an indoor environment map using the collected image information of the walls. Furthermore, it is also hoped that objects such as doors and windows on the walls can be found out with their positions estimated. We have also tried to solve these problems in this study.

As a summary, the objective of this study is to design a vision-based automatic house-layout construction system equipped on an autonomous vehicle system with the following capabilities:

1. navigating automatically in an unknown indoor room space by mopboard following;
2. analyzing environment layout information consisting of wall positions extracted from acquired omni-images;
3. estimating the positions of the windows and doors on the walls;
4. constructing a 3-D space layout of the indoor room space; and
5. drawing a graphic representation of the layout which can be viewed from any perspective directions and used for simulation of interior design for each room.

1.2 Survey of Related Studies

To achieve the goal of automatic house-layout construction by vision-based autonomous vehicle navigation via mopboard following as mentioned previously, object localization for estimating mopboard positions is required. Various visual sensing equipments such as cameras and ultrasonic sensors were used to acquire the 3-D information in the vehicle surroundings in the past studies [1-4]. For depth information estimation, the most common stereo matching method may be used, which is based on the triangulation principle to obtain the relation between a 3-D point and multiple cameras, but matching of corresponding points between images is often a difficult problem in this approach. Use of the laser range finder together with conventional imaging sensors were also used frequently [5-6]. However, the laser range finder has a drawback, that is, it can only scan a 2D plane at a specific height.

Jeng and Tsai [7] proposed a space-mapping method for the omni-camera image, which can be used to estimate the location of an object.

In this study, we propose a method which is based on two concepts. The first is the use of the triangulation principle to measure depths of objects (i.e, their ranges). The other is the use of the space-mapping technique proposed by Jeng and Tsai [7] to get the location for a concerned object. In this way, we can estimate the positions of objects on the floor and walls in a room.

Autonomous vehicles or mobile robots usually suffer from accumulations of mechanical errors during navigation, which cause inaccurate measures of the moving distances and orientations yielded by the odometer in the vehicle. Many methods have been proposed to correct such errors. Chiang and Tsai [8] proposed a method of vehicle localization by detecting a house corner. Wang and Tsai [9] proposed a method to correct the position and direction of a vehicle using omni-cameras on a house ceiling. Chen and Tsai [10] proposed a mechanical error correction method by curve-fitting the erroneous deviations from correct paths. In this study, we utilize the curve-fitting method proposed by Chen and Tsai [10] to correct the positions of the vehicle and use the edge information estimated from images to correct the direction of the vehicle.

Analyzing the omni-images taken by different types of camera is an important topic. Kim and Oh [6] proposed a method for extracting vertical lines from an omni-image with the help of the horizontal lines generated by a laser range finder. Wu and Tsai [11] proposed a method of approximating distorted circular shapes in omni-images by ellipses. It is always desired to construct a map for an unknown indoor environment so that more applications can be implemented. For example, Biber et al. [12] proposed a method for guiding a robot security guard. They calculated the 3-D geometry by a laser range finder and a graph-cut stereo inference technique to build the 3-D model of an environment. Kang and Szeliski [13] proposed a method to extract directly 3-D data from images with very wide FOV's. The goals of this method are achieved by producing panoramas at each camera location and by generating scene models from multiple 360° panoramic views. Meguro et al. [14] presented a motion-stereo device which combines the GPS/DR with omni-directional cameras to rebuild a 3-D space model. This method provides high precision in calculating the distance to far-away objects. However, the running time of the stereo matching process is too long to wait.

1.3 Overview of Proposed System

The goal of this study is to design a system for automatic house-layout construction in an empty indoor room space using a vision-based autonomous vehicle. It is assumed that the adjacency walls in the room are perpendicular to each other. Most houses are of such a type of layout. In order to achieve this goal, the system needs not only a capability of automatic navigation but also a capability to acquire environment information automatically. For the vehicle to have such capabilities, an imaging device with two vertically-aligned omni-cameras connected in a bottom-to-bottom fashion is designed in this study, which will be described later. With the environment information collected from the taken images, a method for constructing a 3-D house layout in graphic form by estimating the locations of mopboard edges, doors, and windows is also proposed. A rough flowchart of the proposed system is illustrated in Figure 1.2.

The system consists of three major phases: setup of the imaging system, vehicle navigation by mopboard following, and 3-D construction of room space, as described respectively in the following.

(1) *System setup* ---

The setup of the system includes space-mapping calibration and mechanic error correction. For camera calibration, we establish a space-mapping table for each omni-camera by finding the relations between specific points (on a specially-designed calibration object) in 2-D omni-images and the corresponding points in 3-D space. As a result, the vehicle which carries the imaging system has the abilities to estimate the distances between the vehicle and an object (the mopboard here) utilizing the omni-images. Mechanic error correction is an important task because the mechanic errors of an odometer usually will led to wrong control instructions and inaccurate vehicle localization information. We also propose a mechanic error correction technique to deal with this problem.

(2) *Vehicle navigation by mopboard following* ---

Mopboard positions are essential for vehicle navigation in this study. Therefore, we propose a method to detect mopboard edges from omni-images, and estimate relative mopboard positions with respect to the vehicle by using the space-mapping table mentioned above. When the vehicle navigates, the mopboard edges are also used to adjust the pose of the vehicle to keep its navigation path parallel to each wall. The vehicle system is designed to record the environment data during navigation. After the navigation session is completed, a *floor-layout map* is created according to the collected mopboard edge data by using a global optimization method for mopboard edge line fitting.

(3) *3-D construction of room space* ---

Only creating a floor-layout map is insufficient for use as a 3-D model of the indoor room space. The objects on walls such as windows and doors must also be detected and be drawn to appear in the desired 3-D room model. In this study, we also propose methods for detecting and recognizing the doors as well as windows on walls in the *upper* and *lower* omni-images (taken respectively by the upper omni-camera and the lower one in the proposed two-camera omni-directional imaging system), locating them in the 3-D space, and drawing them in the final 3-D room model in a graphic form.

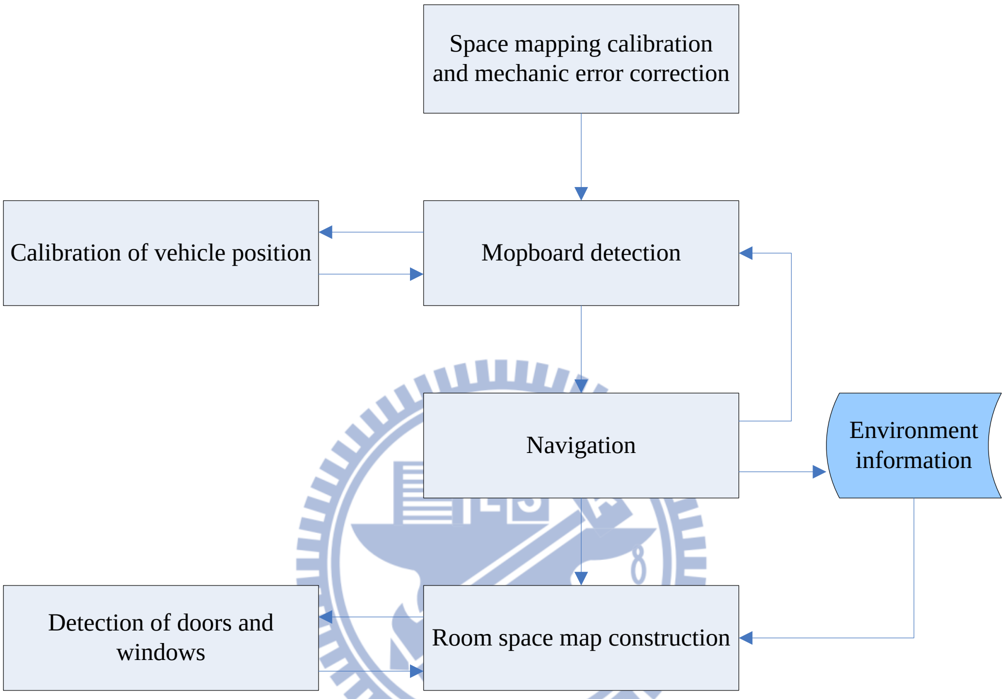

Space mapping calibration and mechanic error correction
Calibration of vehicle position
Mopboard detection
Navigation
Environment information
Detection of doors and windows
Room space map construction

Figure 1.2 Flowchart of proposed system.

1.4 Contributions

The major contributions of this study are listed as follows.

1. A technique for finding the focus point of a hyperbolic mirror used in a process of creating a space-mapping table is proposed.
2. A method for detection of mopboards in omni-images is proposed.
3. A method of using a look-up space-mapping table to estimate the positions of mopboard edges in omni-images is proposed.
4. A pattern classification technique that can assign each extracted mopboard

edge point correctly to the wall adjacent to the mopboard is proposed.

5. A strategy for vehicle navigation by mopboard following in an unknown indoor space with a pattern of mutually-perpendicular mopboard edges is proposed.
6. A global optimization method for constructing an indoor floor-layout map using extracted mopboard edges is proposed.
7. A method for retrieving accurate 3-D data from different omni-images taken by the two-camera omni-directional imaging system subject to the floor-layout map is proposed.
8. Methods for detecting and recognizing the doors as well as windows on walls from different omni-images are proposed.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, the configuration of the proposed two-camera omni-directional imaging system and the principle of automatic 3-D house-layout construction are described. In Chapter 3, the proposed techniques for space-mapping calibration of the omni-cameras and for mechanic-error correction are described. In Chapter 4, the proposed methods for mopboard detection and the proposed strategy of vehicle navigation by mopboard following are presented. In Chapter 5, the proposed methods for room space construction and analysis of environment data in different omni-images are described. Experimental results showing the feasibility of the proposed methods are described in Chapter 6. Conclusions and suggestions for future works are included finally in Chapter 7.

Chapter 2
Principle of Proposed Automatic 3-D House-layout Construction and System Configuration

2.1 Introduction

For indoor 3-D house-layout construction, the use of a vision-based intelligent autonomous vehicle is a good idea because it can save manpower and the vision sensors on the vehicle can assist navigation and localization. Besides, the vehicle can also gather indoor environment information with its mobility.

In the proposed system, we equip the vehicle with two catadioptric omni-cameras which have larger FOV’s, each covering a half sphere of the 3D space around the camera. The two omni-cameras are connected and vertically-aligned in a bottom-to-bottom fashion, form a two-camera imaging device, and are installed on the vehicle. In such a connected fashion, the imaging system covers the upper and lower semi-spherical FOV’s simultaneously so that the images of the floor and the ceiling can be captured simultaneously. With this imaging system, the vehicle can navigate and collect desired information in indoor rooms. Related control instructions of the vehicle and communication tools are also developed in this study. The entire system configuration, including hardware equipments and software, will be described in Section 2.2.

In order to use the vehicle to carry out the indoor house-layout construction task, it is necessary to have an appropriate strategy to tell the vehicle where to go, when to turn around, and what kind of information should be collected. When a house agent wants to use the vehicle to obtain the layout of each empty room in a house in advance before getting it ready for sale, it is inconvenient to conduct a learning work to teach the vehicle a navigation path that includes some specific positions as turning spots. Instead, it is desired that the vehicle can navigate and gather environment information in an unlearned indoor room space *automatically*. We will introduce the main idea of such an automatic 3D house-layout construction process, which we propose in this study, in Section 2.3. Also, the system must have a process to transform the collected data into house-layout structure information. In Section 2.4, we will describe the outline of the proposed process of house-layout construction.

2.2 System Configuration

In the proposed vehicle system, we make use of a Pioneer 3-DX vehicle made by MobileRobots Inc. as a test bed. The vehicle is equipped with an imaging system composed of two omni-directional catadioptric cameras. The imaging system is not only part of the vehicle system but also plays an important role of gathering environment information and locating the vehicle. A diagram illustrating the configuration of this system is shown in Figure 2.1. Because we control the vehicle system remotely, some wireless communication equipments are necessary. The detail of the hardware architecture and the used equipments are described in Section 2.2.1.

In order to develop the vehicle system, the software that provides some commands and control interface is essential for users to control the vehicle and cameras. The software we use or developed in this study is described in Section 2.2.2.

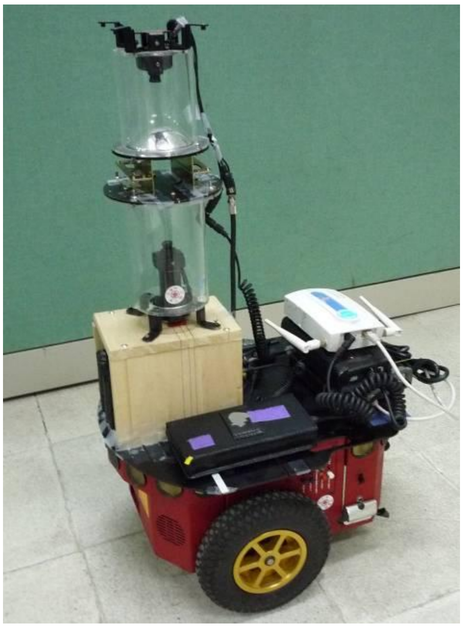
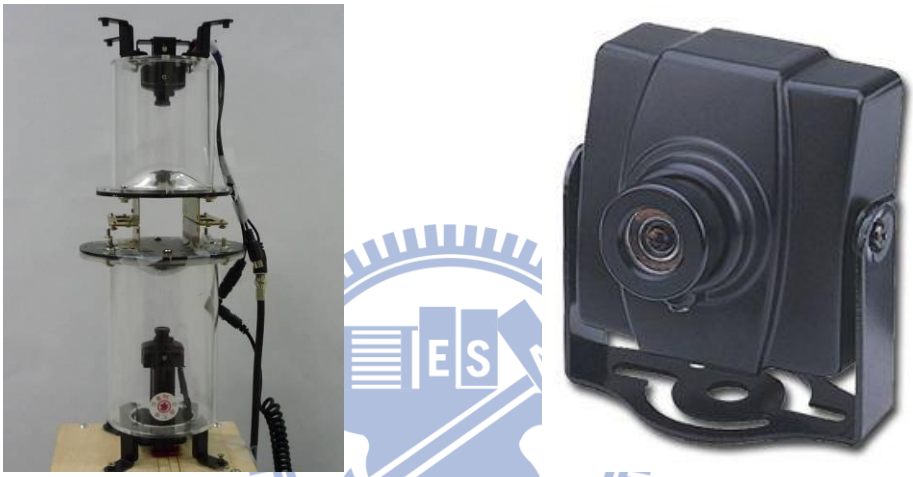


Figure 2.1 Equipment on proposed vehicle system used in this study.

2.2.1 Hardware configuration

The hardware equipments we use in this study include three principal systems --- the vehicle system, the control system, and the imaging system. In the vehicle system, the Pioneer 3-DX vehicle itself has a 44cm×38cm×22cm aluminum body with two 19cm wheels and a caster. It can reach a speed of 1.6 meters per second on flat floors, and climb grades of 25° and sills of 2.5cm. It can carry payloads up to 23 kg. The payloads include additional batteries and all accessories. By three 12V rechargeable lead-acid batteries, the vehicle can run 18-24 hours if the batteries are fully charged initially. An embedded control system in the vehicle allows a user to issue commands to control the vehicle to move forward or backward and turn around. The system can also return some status information to the user. Besides, the vehicle is equipped with an odometer which is used to record the pose and the position of the vehicle. To control the vehicle remotely, a wireless connection between the user and the vehicle is necessary. A *WiBox* is used to communicate with the vehicle by RS-232, so the user can control the vehicle remotely over a network from anywhere.

The second part is the two-camera imaging device which includes two catadioptric omni-cameras vertically-aligned and connected in a bottom-to-bottom fashion, as shown by Figure 2.2(a). The catadioptric omni-camera used in this study is a combination of a *reflective hyperbolic mirror* and a *CCD camera*. Each CCD camera used in the imaging system is shown in Figure 2.2(b), and a detailed specification about it is included in Table 2.1.



(a) (b)

Figure 2.2 Two catadioptric omni-cameras connected in a bottom-to-bottom fashion into a single two-camera imaging device used in this study. (a) The two-camera device. (b) The CCD camera used in the imaging device.

Table 2.1 Specification of the CCD cameras used in the imaging device.

Model Name	RYK-2849
Image Device	Color camera with SONY 1/3" super had CCD sensor in mini metal case
Picture Elements	NTSC:510x492, PAL:500x582
Resolution	420 TVL
Power Supply	DC12V±10%
Operating Temp.	-10℃ ~ 50℃
Video Output	1 Vp-p / 75 Ohms
Audio Output	2 Vp-p / 50 Ohms (option)

Because the output signals of the CCD cameras used in this study are analog, an *AXIS 241QA Video Server* installed in it, as shown in Figure 2.3, has a maximum frame rate of 30 fps. It can convert analog video signals into digital video streams and send them over an IP network to the main control unit (described next). The imaging device and the AXIS Video Server are combined into an imaging system.

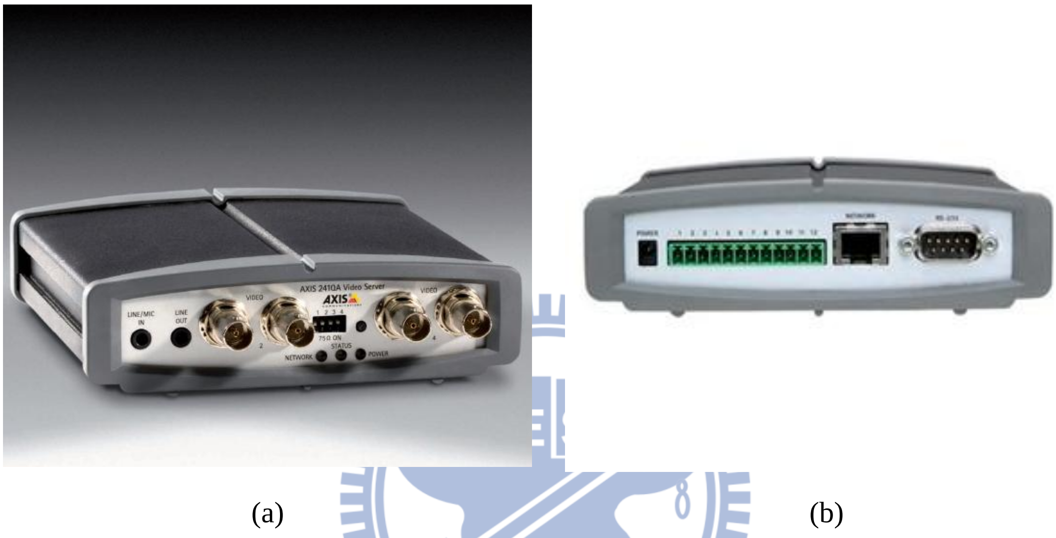


Figure 2.3 The Axis 241QA Video Server used in this study. (a) A front view of the Video Server. (b) A back view.

In the control system, a laptop computer is used to run the program developed in this study. A kernel program can be executed on the laptop to control the vehicle by issuing commands to it and to conduct processing tasks on captured images. With an access point, all information between the user and the vehicle can be delivered through wireless networks (IEEE 802.11b and 802.11g), and captured images can also be transmitted to users at speeds up to 54 Mbit/s. The entire structure of the vehicle system used in the study is shown in Figure 2.4.

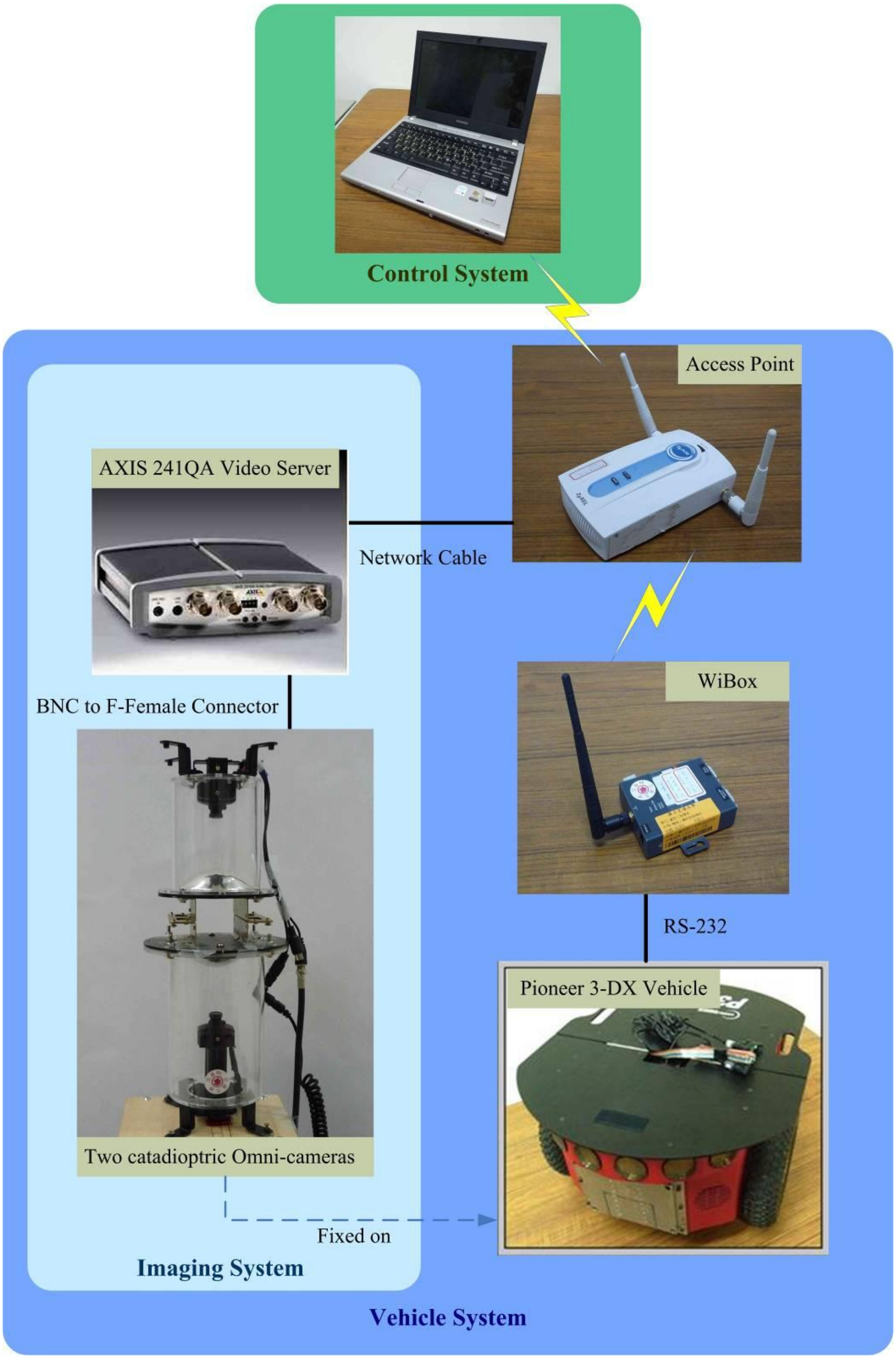


Figure 2.4 Structure of proposed system.

2.2.2 Software configuration

The MobileRobots, Inc. provides an *ARIA (Advanced Robotics Interface*

Application) for use in this study, which is an API (application programming interface) that assists developers in communicating with the embedded system of the vehicle, through an RS-232 serial port or a TCP/IP network connection. The ARIA is an object-oriented toolkit usable under Linux or Win32 OS by the C++ language. Therefore, we use the Borland C++ builder as the development tool in our experiments to control the vehicle by the ARIA. The status information of the vehicle can be obtained by means of the ARIA.

About the AXIS 241QA Video Server, the AXIS Company provides a development toolkit called *AXIS Media Control SDK*. Using the Media ActiveX component from the SDK, we can easily have a preview of the omni-image and capture the current image data. It is helpful for developers to conduct any task with the grabbed image.

2.3 Principle of Proposed Automatic Floor-layout Construction by Autonomous Vehicle Navigation and Data Collection

For vehicle navigation in an unknown empty room space, we propose a navigation strategy based on mopboard following. After a navigation session is completed, we use the estimated mopboard edge points to construct a floor-layout map. The main process is shown in Figure 2.5.

In this study, we use a space-mapping technique proposed by Jeng and Tsai [7] to compute the location of a concerned object using a so-called pano-mapping table. When the vehicle starts a navigation session, it uses the detected mopboard edge points from omni-images to decide whether or not it has to conduct an adjustment of its direction to keep its navigation path parallel to each wall. Then, the mopboard edge points are detected again from the omni-images taken by the imaging system mentioned previously. By a pattern classification technique, the distance between the vehicle and each wall can be estimated more accurately, and each extracted mopboard edge point can be assigned to the wall which contains it. In this way, the vehicle can estimate the distances between the walls and itself, and know whether to move forward or turn around. After the navigation session is completed, a floor-layout map is created according to the collected mopboard edge data by using a global optimization method for mopboard edge line fitting proposed in this study.

2.4 Outline of Proposed Automatic 3-D House-layout Construction Process

Only creating a floor-layout map, as described in Section 2.3, is insufficient for use as a 3-D model of the indoor room space. The objects on walls such as windows and doors must also be detected and be drawn to appear in the desired 3-D room model. We have proposed methods for detecting and recognizing the doors as well as windows on walls in the *upper* and *lower* omni-images. The principal steps of the methods are shown in Figure 2.6.

First, we determine a *scanning range* with two direction angles for each pair of omni-images based on the line equation of the floor layout. Because the lower omni-camera is installed to look downward, it can cover the mopboard on the wall. We use a pano-mapping table lookup technique to get the scanning radius in the omni-image by transforming the 3-D space points into corresponding image coordinates. With the scanning region for each omni-image, we can retrieve appropriate 3-D information from different omni-images. Each object which is detected by the scanning region of each omni-image is regard as an individual one. Some objects on the wall, such as windows and doors, may appear in the pair of omni-images (the upper and the lower ones). Therefore, we have to merge the objects which are detected separately from the upper omni-image and the lower omni-image according to their positions in order to recognize the doors as well as the windows on walls. Then, we can locate them in the 3-D space and draw them in the final 3-D room model in a graphic form.

In summary, the proposed automatic 3-D house-layout construction process includes the following major steps:

1. automatic floor-layout construction by autonomous vehicle navigation and data collection as described in Section 2.4;
2. determine a scanning region for each omni-image according to the floor–layout edges;
3. retrieve information from the scanning region of each omni-image;
4. combine those objects which are detected separately from the upper and lower omni-cameras according to their positions;
5. recognize doors and windows from these combined objects;
6. construct the house-layout model with doors and windows on it in a graphic form.

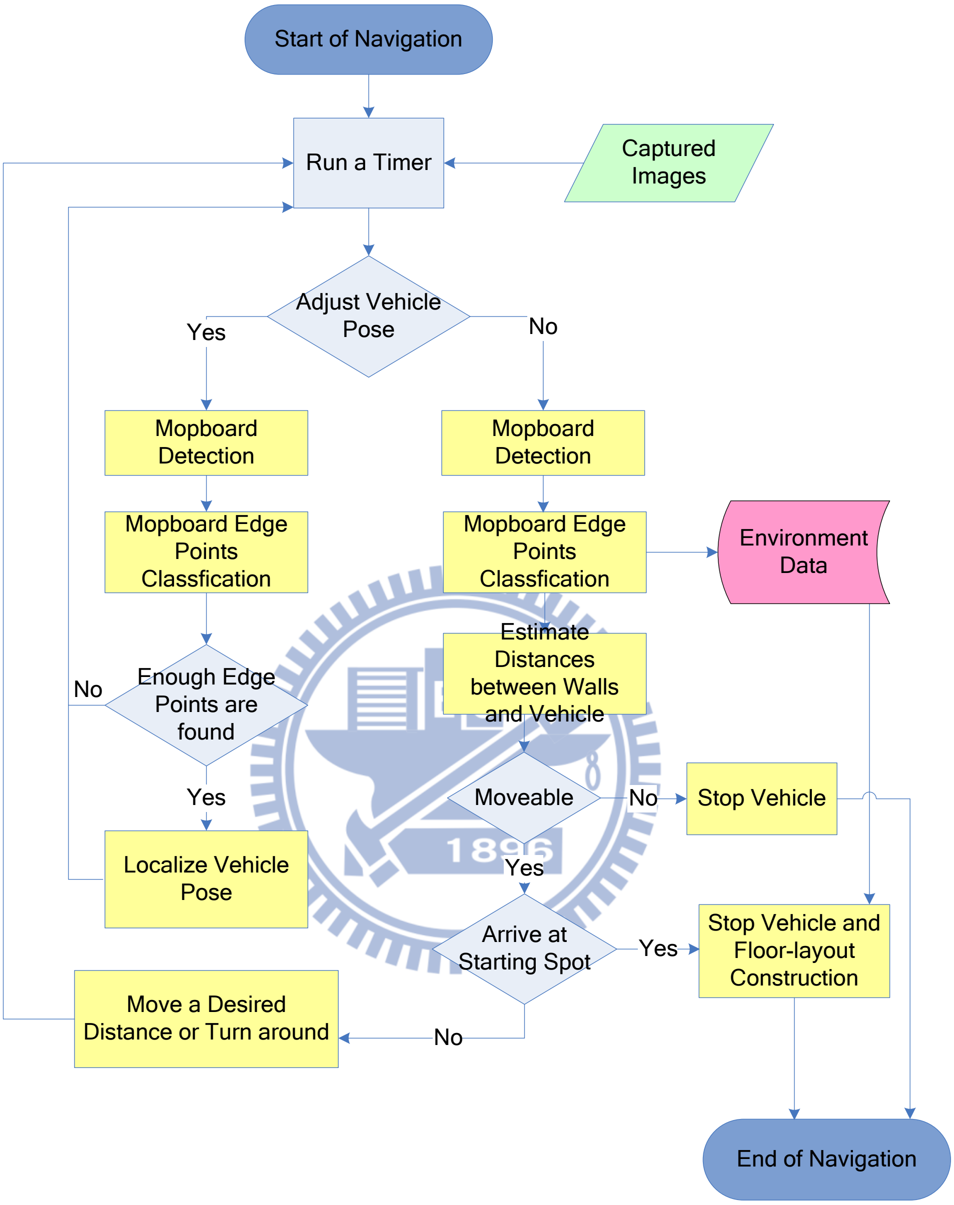

Start of Navigation
Run a Timer
Captured Images
Adjust Vehicle Pose
Yes
No
Mopboard Detection
Mopboard Detection
Mopboard Edge Points Classfication
Mopboard Edge Points Classfication
Environment Data
Estimate Distances between Walls and Vehicle
No
Enough Edge Points are found
Yes
Moveable
No
Stop Vehicle
Yes
Localize Vehicle Pose
Arrive at Starting Spot
Yes
Stop Vehicle and Floor-layout Construction
Move a Desired Distance or Turn around
No
End of Navigation

Figure 2.5 Flowchart of proposed process of automatic floor-layout construction.

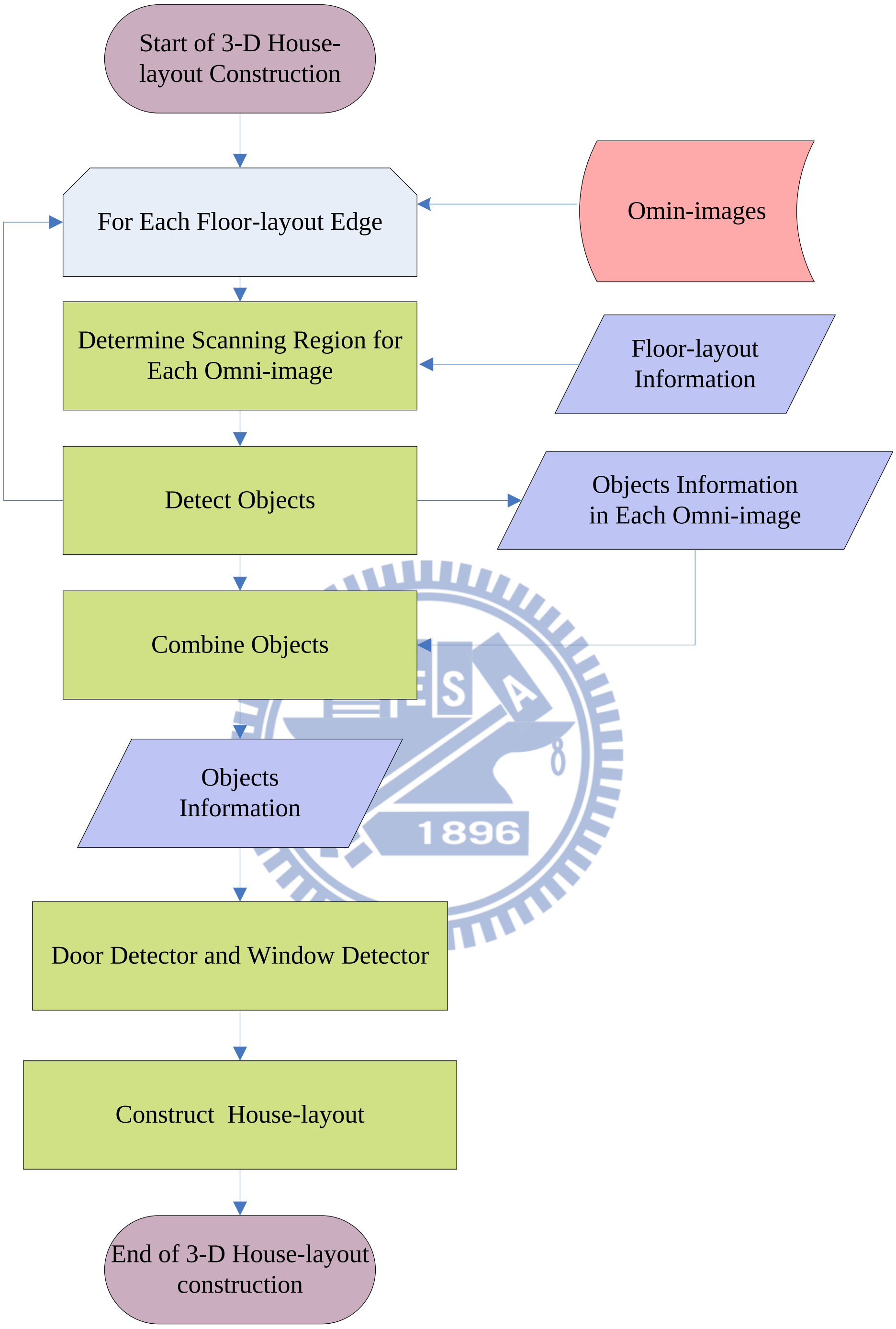

Start of 3-D House-layout Construction
For Each Floor-layout Edge
Omin-images
Determine Scanning Region for Each Omni-image
Floor-layout Information
Detect Objects
Objects Information in Each Omni-image
Combine Objects
Objects Information
Door Detector and Window Detector
Construct House-layout
End of 3-D House-layout construction

Figure 2.6 Flowchart of proposed outline of automatic 3-D house-layout construction.

Chapter 3
Calibration of a Two-camera Omni-directional Imaging System and Vehicle Odometer

3.1 Introduction

The vehicle used in this study is equipped with two important devices which are a two-camera imaging system and a vehicle odometer. We describe the proposed methods of calibration for these two equipments in this chapter. Before describing the proposed methods, we introduce the definition of the coordinate system used in this study in Section 3.1.1 and the relevant coordinate transformation in Section 3.1.2.

The catadioptric omni-camera used in this study is a combination of a reflective hyperboloidal-shaped mirror and a perspective CCD camera. Both the perspective CCD camera and the mirror are assumed to be properly set up so that the omni-camera becomes to be of a *single-viewpoin*t (SVP) configuration. It is also assumed that the optical axis of the CCD camera coincides with the *transverse axi*s of the hyperboloidal mirror and that the transverse axis is perpendicular to the mirror base plane.

For vehicle navigation by mopboard following, the mopboard positions are essential for vehicle guidance. Besides, these mopboard edge points are very important for a 3-D house-layout construction. In the proposed system, the vehicle estimates the distance information by analyzing images captured from the imaging system. Before the use of the imaging system, a camera calibration procedure is needed. For this purpose, we use a space-mapping technique proposed by Jeng and Tsai [7] to create a space-mapping table for each omni-camera by finding the relations between specific points in 2-D omni-images and the corresponding points in 3-D space. In this way, the conventional task of calculating the projection matrix for transforming points between 2-D omni-image and 3-D space can be omitted. The detail about camera calibration is described in Section 3.2.

For vehicle navigation in indoor environments, the vehicle position is the most important information which is not only used for guiding the vehicle but also as a local center to transform the estimated positions in the camera coordinate system (CCS) into the global positions in the global coordinate system (GCS). Though, the position of the vehicle provided by the odometer of the vehicle may be imprecise because of the incremental mechanic errors of the odometer. It also results in deviations from a planned navigation path. Therefore, it is desired to conduct a calibration task to eliminate the errors. In Section 3.3, we will review the method for vehicle position calibration which was proposed by Chen and Tsai [10], and a vision-based calibration method for adjusting the vehicle direction during navigation will be described in the following chapter.

3.1.1 Coordinate Systems

Four coordinate systems are utilized in this study to describe the relative locations between the vehicle and the navigation environment. The coordinate systems are illustrated in Figure 3.1. The definitions of all the coordinate systems are described in the following.

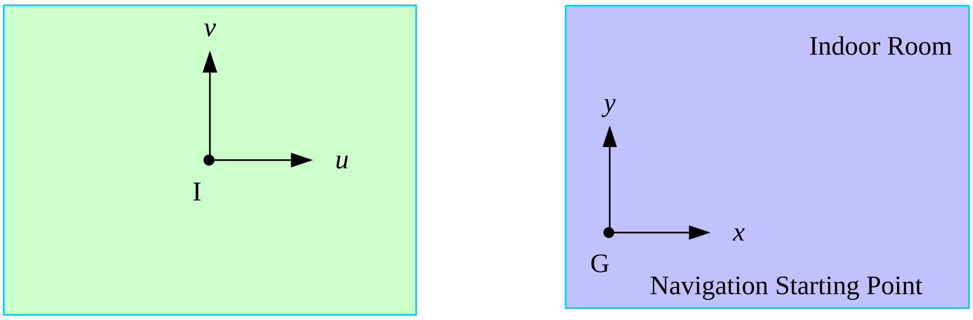
(1) Image coordinate system (ICS): denoted as (*u*, *v*). The *u*-*v* plane coincides with the image plane and the origin I of the ICS is placed at the center of the image

plane.

(2) Global coordinate system (GCS): denoted as (x, y). The origin G of the GCS is a pre-defined point on the ground. In this study, we define G as the starting position of the vehicle navigation by the mopboard following process.

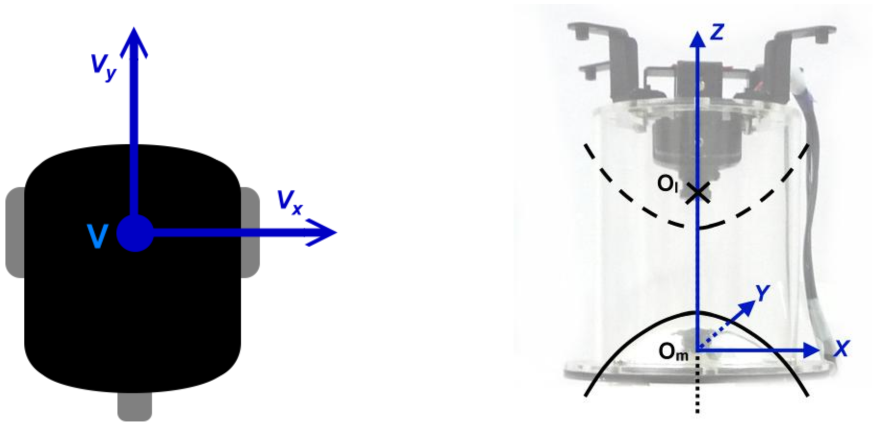
(3) Vehicle coordinate system (VCS): denoted as (V_x, V_y). The V_x-V_y plane is coincident with the ground. And the origin V is placed at the middle of the line segment that connects the two contact points of the two driving wheels with the ground. The V_x-axis of the VCS is parallel to the line segment joining the two driving wheels and through the origin V. The V_y-axis is perpendicular to the V_x-axis and passes through V.

(4) Camera coordinate system (CCS): denoted as (X, Y, Z). The origin O_m of the CCS is a focal point of the hyperbolic mirror. And the X-Y plane coincides with the image plane and the Z-axis coincides with the optical center inside the lens of the CCD camera.



(a) (b)

Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The vehicle coordinate system. (c) The global coordinate system. (d) The camera coordinate system.



(c) (d)

Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The vehicle coordinate system. (c) The global coordinate system. (d) The camera coordinate system. (continued)

3.1.2 Coordinate Transformation

In this study, the GCS is determined when starting a navigation session. The CCS and VCS follow the vehicle during navigation. The relation between the GCS and the VCS is illustrated in Figure 3.2(a). We assume that (x_p, y_p) represents the coordinates of the vehicle in the GCS, and that the relative rotation angle denoted as θ is the directional angle between the positive direction of the x-axis in the GCS and the positive direction of the V_x-axis in VCS. The coordinate transformation between the VCS and the GCS can be described by the following equations:

$$x = V_x \times \cos\theta - V_y \times \sin\theta + x_p; \tag{3.1}$$

$$y = V_x \times \sin\theta + V_y \times \cos\theta + y_p. \tag{3.2}$$

The main concept about the relation between the CCS and the ICS is illustrated in Figure 3.2(b), though the CCS in Figure 3.2(b) is a little different from the CCS in this study. The relation plays an important role for transforming the camera coordinates (X, Y, Z) of a space point P into the image coordinates (u, v) of its corresponding image point p. The relation may be described by the following equations [15-17]:

$$\tan\alpha = \frac{(b^2+c^2)\sin\beta - 2bc}{(b^2-c^2)\cos\beta}; \tag{3.3}$$

$$r = \sqrt{u^2+v^2}; \tag{3.4}$$

$$\sin\beta = \frac{f}{\sqrt{r^2+f^2}}; \tag{3.5}$$

$$\cos\beta = \frac{r}{\sqrt{r^2+f^2}}; \tag{3.6}$$

$$\tan\alpha = \frac{Z-c}{\sqrt{X^2+Y^2}}, \tag{3.7}$$

where a and b are two parameters satisfying the equation of the hyperboloidal mirror as follows:

$$\frac{R^2}{a^2} - \frac{Z^2}{b^2} = -1, \quad R = \sqrt{X^2+Y^2}, \quad c = \sqrt{a^2+b^2};$$

O$_\text{m}$ and O$_\text{l}$ are located at (0, 0, 0) and (0, 0, $-c$) separately in the CCS; and f is the focal length of the camera. According to the *rotational invariance* property of the omni-camera [17], and by combining with the above equations, we have

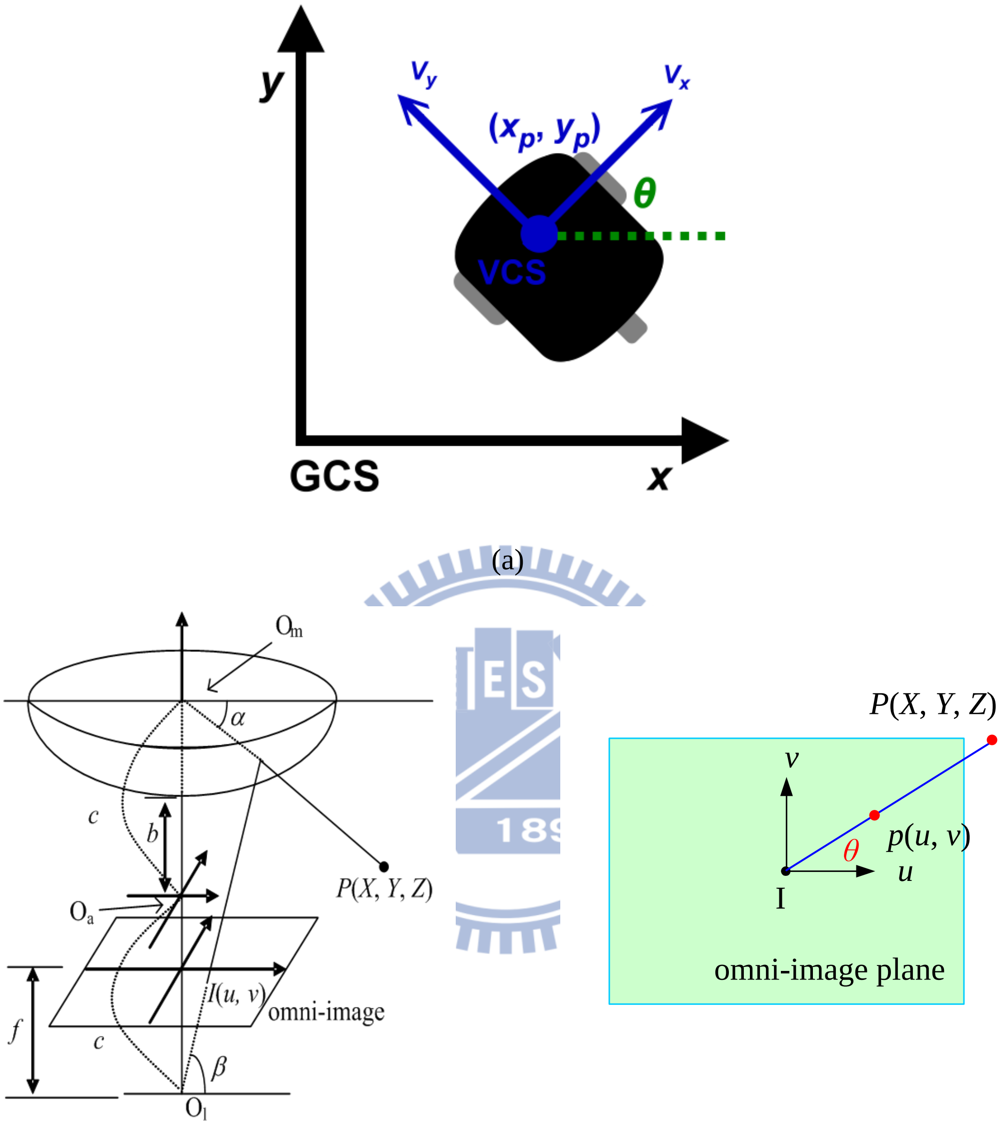
$$\cos\theta = \frac{X}{\sqrt{X^2+Y^2}} = \frac{u}{\sqrt{u^2+v^2}}; \tag{3.8}$$

$$\sin\theta = \frac{Y}{\sqrt{X^2+Y^2}} = \frac{v}{\sqrt{u^2+v^2}}; \tag{3.9}$$

$$u = \frac{Xf(b^2-c^2)}{(b^2+c^2)(Z-c) - 2bc\sqrt{(Z-c)^2+X^2+Y^2}}; \tag{3.10}$$

$$v = \frac{Yf(b^2-c^2)}{(b^2+c^2)(Z-c) - 2bc\sqrt{(Z-c)^2+X^2+Y^2}}, \tag{3.11}$$

where θ is the angle of the space point P with respect to the X-axis as well as that of the corresponding image point *p* with respect to the *u*-axis, as shown in Figure 3.2(c).



(a)

(b)

(c)

Figure 3.2 The relations between different coordinate systems in this study. (a) The relation between the GCS and VCS (b) Omni-camera and image coordinate systems [11]. (c) Top view of (b).

3.2 Calibration of Omni-directional Cameras

Before using the imaging system, a calibration procedure for the omni-cameras is indispensible. However, the conventional calibration method is complicated for calculating *intrinsic* and *extrinsic* parameters. An alternative way is to use the *space-mapping* technique to estimate the relation between points in the 2-D image plane and 3-D space and to establish a space-mapping table for it [7]. The detailed process is reviewed in Section 3.2.2. In the process of establishing the space-mapping table, the information about the focal location of the hyperboloidal mirror is important because the focal point is taken to be the origin in the CCS. The process to find the focal point of the hyperboloidal mirror is described in Section 3.2.1.

3.2.1 Proposed Technique for Finding Focal Point of Hyperboloidal Mirror

In order to creating the space-mapping table, it is indispensible to select some pairs of world space points with known positions and the corresponding points in the omni-images. Note that an image point *p* is formed by any of the world space points which all lie on the incoming ray R, as shown in Figure 3.3, where we suppose that O_m is the focal point of the hyperboloidal mirror, O_w is on the transverse axis of the hyperboloidal mirror, and P_1 and P_2 are two space points on the ray *R*. Besides, we also assume that the corresponding image point is *p* .Subsequently, we have the corresponding point pairs (P_1, *p*) and (P_2, *p*) which then are used to create the table. However, if we take O_w as the focal point, as a result P_1 and P_2 will lie on different light rays, though the corresponding image points are still *p*. In this way, the incorrect pairs will result in an incorrect space-mapping table. To provide accurate pairs, we must find out the position of the focal point of the hyperboloidal mirror.

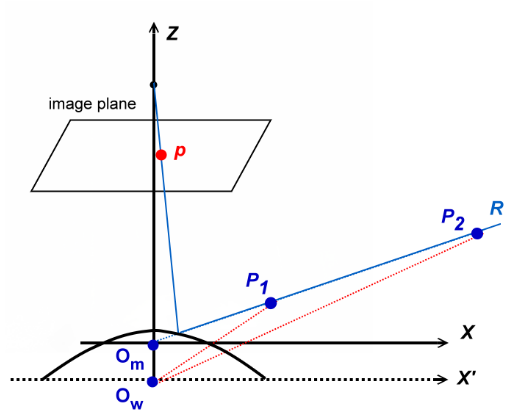


Figure 3.3 The space points and their corresponding image points.

To find out the focal point of the hyperboloidal mirror, as shown in Figure 3.4, we use two different landmarks L_1 and L_2 which have the same corresponding image point p with known heights and horizontal distances from the transverse axis of the hyperboloidal mirror. We assume that O_w is at (0, 0, 0). Then, according to the involved geometry shown in Figure 3.4, the position of the focal point can be computed by the following equations:

$$\tan\theta = \frac{H_1 - \overline{O_m O_w}}{D_1} = \frac{H_2 - H_1}{D_2 - D_1}\,; \tag{3.12}$$

$$\overline{O_m O_w} = H_1 - D_1 \times \tan\theta = H_1 - D_1 \times \frac{H_2 - H_1}{D_2 - D_1}\,. \tag{3.13}$$

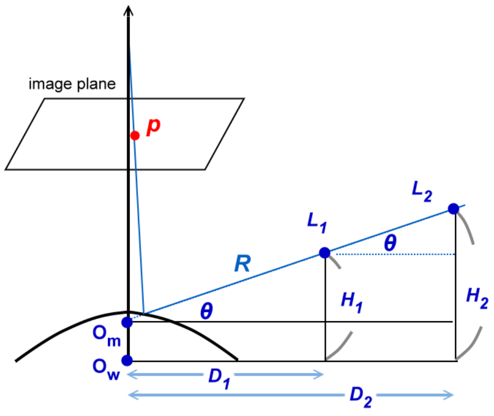


Figure 3.4 Finding out the focal point O_m.

3.2.2 Review of Adopted Camera Calibration Method

Jeng and Tsai [7] proposed a space-mapping method to estimate the relation between points in the 2-D image plane and the 3-D space and to establish a space-mapping table. By observing Figure 3.2(b) and Eq. (3.3) through (3.7), it is noted that there exists a one-to-one relation between the elevation angle α and the radial distance r. By using the space-mapping table as well as according to the rotational invariance property, we can know the relative elevations and directions of the concerned targets in images.

The adopted method [7] includes three major procedures: landmark learning, estimation of coefficients of a radial stretching function, and space-mapping table creation, as described respectively in the following.

(1) *Landmark learning* ---

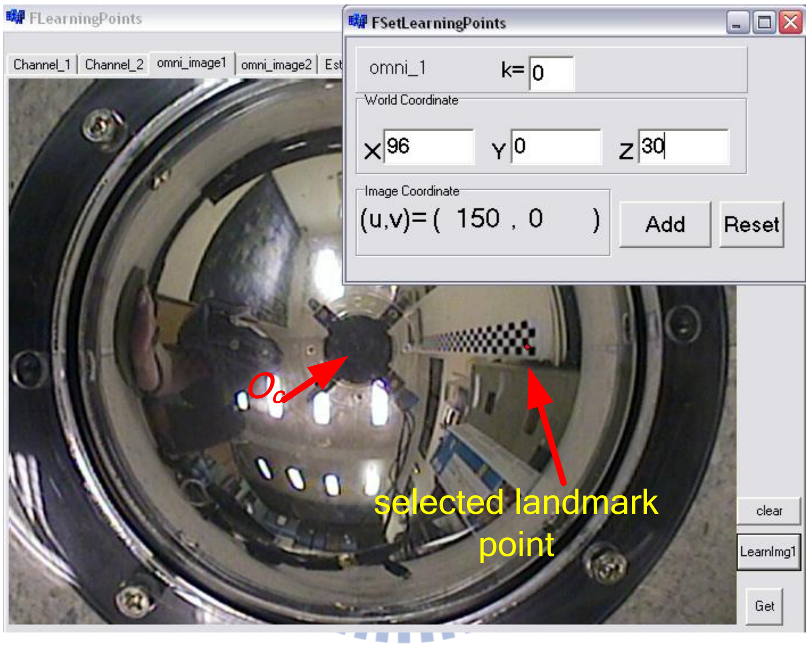
We select some landmark point pairs of world space points with known positions and their corresponding pixels in a taken omni-image. More specifically, the coordinates of the landmark points are measured manually with respect to a selected origin in the CCS. In this study, the origin in the CCS is a focal point of the hyperboloidal mirror which can be found as described in Section 3.2.1. As shown in Figure 3.5, we selected n landmark points on the omni-image, and recorded the pairs of the space coordinates (X_k, Y_k, Z_k) and the image coordinates (u_k, v_k), where $k = 0$, 1,…, $n - 1$.

(2) *Estimation of coefficients of radial stretching function* ---

As shown in Figure 3.6, the same elevation angles correspond to the same radial distances. Therefore, the radial distance r from each image pixel p with (u, v) in the ICS to the image center O_c at (u_0, v_0) may be computed by $r=f_r(\rho)$. In this study, we attempt to describe the function $f_r(\rho)$, called a *radial stretching function*, by the following 5th-degree polynomial function:

$$r = f_r(\rho) = a_0 + a_1 \times \rho^1 + a_2 \times \rho^2 + a_3 \times \rho^3 + a_4 \times \rho^4 + a_5 \times \rho^5, \tag{3.14}$$

where a_0 to a_5 can be estimated using the landmark point pairs, as described in the following major steps [7].



omni_1 k= 0

X 96 Y 0 Z 30

(u,v)=(150 , 0)

Figure 3.5 The interface for selecting landmark points.

Step 1. Elevation angle and radial distance calculation ---

Use the selected landmark point pair (P_k, p_k), including the coordinates (X_k, Y_k, Z_k) in the CCS and the coordinates (u_k, v_k) in the ICS, to calculate the elevation angle ρ_k of P_k, in the CCS and the radial distance r_k of p_k in the ICS by the following equations:

$$\rho_k = \tan^{-1}(\frac{Z_k}{D_k}), \tag{3.15}$$

$$r_k = \sqrt{u_k^2 + v_k^2}, \tag{3.16}$$

where D_k is the horizontal distance between the landmark point P_k and the focal point

O_m of the mirror.

Step 2. Calculation of coefficients of the radial stretching function ---

Substitute all pairs (ρ_k, r_k), where k=0, 1, …, n – 1, into Eq. (3.14) to get n homogeneous equations as follows:

$$\begin{array}{l} r_0 = f_r(\rho_0) = a_0 + a_1\times\rho_0^1 + a_2\times\rho_0^2 + a_3\times\rho_0^3 + a_4\times\rho_0^4 + a_5\times\rho_0^5, \\ r_1 = f_r(\rho_1) = a_0 + a_1\times\rho_1^1 + a_2\times\rho_1^2 + a_3\times\rho_1^3 + a_4\times\rho_1^4 + a_5\times\rho_1^5, \\ \qquad\vdots \\ r_{n-1} = f_r(\rho_{n-1}) = a_0 + a_1\times\rho_{n-1}^1 + a_2\times\rho_{n-1}^2 + a_3\times\rho_{n-1}^3 + a_4\times\rho_{n-1}^4 + a_5\times\rho_{n-1}^5. \end{array} , \qquad (3.17)$$

and solve them to get the coefficients (a_0, a_1, a_2, a_3, a_4, a_5) of the function $f_r(\rho)$.

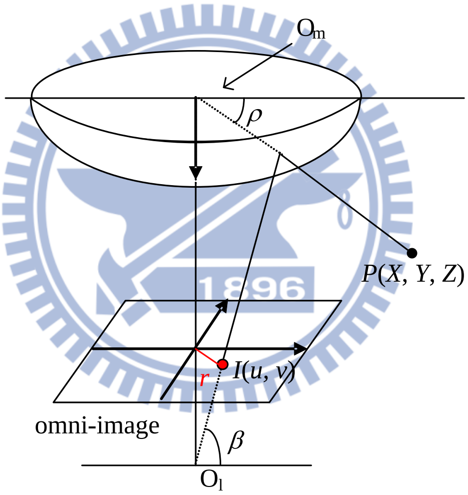


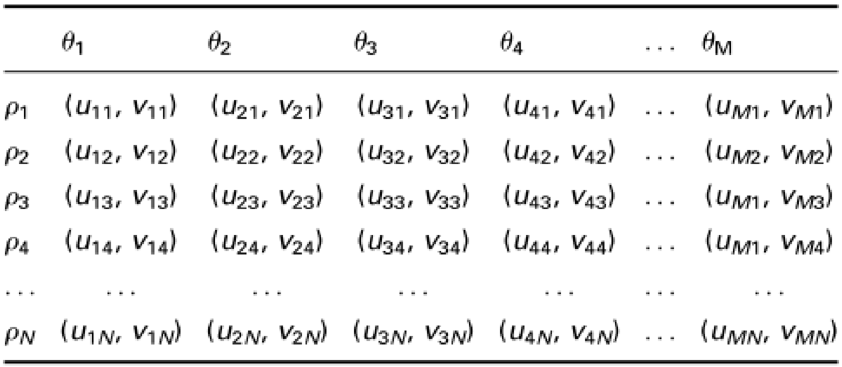
Figure 3.6 Illustration of the relation between radial r in ICS and elevation ρ in CCS.

Step 3. Space-mapping table creation ---

The space-mapping table to be constructed is a 2-dimensional table with horizontal and vertical axes specifying respectively, the range of the azimuth angle θ as well as that of the elevation angel ρ of all possible incident light rays going through the focal point of the mirror.

Table 3.1 shows an example of the pano-mapping table of size M×N. Each entry E_{ij} with indices (i, j) in the table specifies an azimuth-elevation angle pair (θ_i, ρ_j), which represents an infinite set S_{ij} of points in the CCS passing through by the light ray with azimuth angel θ_i and elevation angle ρ_j. And for the reason that these world space points in S_{ij} are all projected onto the identical pixel p_{ij} in any omni-image taken by the camera, a mapping relation between S_{ij} and p_{ij} as shown in Figure 3.7 can be derived. The mapping table shown in Table 3.1 is constructed by filling entry E_{ij} with coordinates (u_{ij}, v_{ij}) of pixel p_{ij} in the omni-image.

Table 3.1 Example of pano-mapping table of size M×N [7].



	θ_1	θ_2	θ_3	θ_4	...	θ_M
ρ_1	(u_{11}, v_{11})	(u_{21}, v_{21})	(u_{31}, v_{31})	(u_{41}, v_{41})	...	(u_{M1}, v_{M1})
ρ_2	(u_{12}, v_{12})	(u_{22}, v_{22})	(u_{32}, v_{32})	(u_{42}, v_{42})	...	(u_{M2}, v_{M2})
ρ_3	(u_{13}, v_{13})	(u_{23}, v_{23})	(u_{33}, v_{33})	(u_{43}, v_{43})	...	(u_{M1}, v_{M3})
ρ_4	(u_{14}, v_{14})	(u_{24}, v_{24})	(u_{34}, v_{34})	(u_{44}, v_{44})	...	(u_{M1}, v_{M4})
...	...	...	...	...	...	...
ρ_N	(u_{1N}, v_{1N})	(u_{2N}, v_{2N})	(u_{3N}, v_{3N})	(u_{4N}, v_{4N})	...	(u_{MN}, v_{MN})

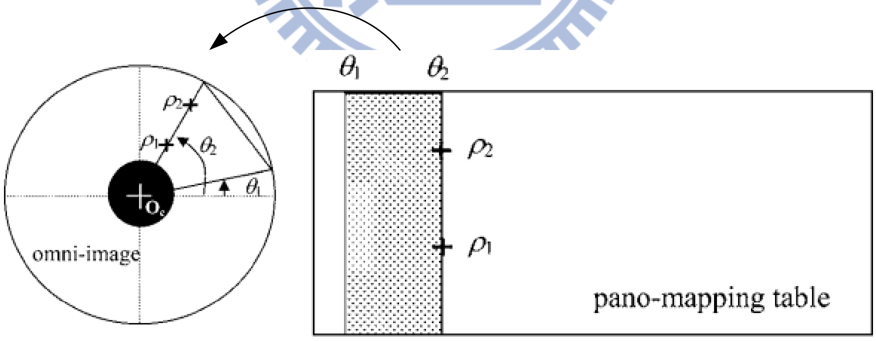


Figure 3.7 Mapping between pano-mapping table and omni-image [7] (or space-mapping table in this study).

More specifically, the process of filling entries of the pano-mapping table can be summarized by the following major steps [7].

Step 1. Divide the range 2π of azimuth angles into M intervals and compute θ_i by

$$\theta_i = i \times (2\pi / M),\ \text{for } i = 0,1,\ldots,M-1. \tag{3.18}$$

Step 2. Divide the range [ρ_e, ρ_s] of the elevation angles into N intervals and compute ρ_j by

$$\rho_j = j \times [(\rho_e - \rho_s)/N] + \rho_s,\ \text{for } j = 0, 1, ..., N-1. \tag{3.19}$$

Step 3. Fill the entry E_{ij} with corresponding image coordinates (u_{ij}, v_{ij}) by

$$\begin{aligned} u_{ij} &= r_j \times \cos\theta_i; \\ v_{ij} &= r_j \times \sin\theta_i, \end{aligned} \tag{3.20}$$

where r_j is computed by

$$r_j = f_r(\rho_j) = a_0 + a_1 \times \rho_j^1 + a_2 \times \rho_j^2 + a_3 \times \rho_j^3 + a_4 \times \rho_j^4 + a_5 \times \rho_j^5 \tag{3.21}$$

with the coefficients (a_0, a_1, a_2, a_3, a_4, a_5) computed above.

After establishing the space-mapping table, we can determine the elevation and azimuth angle of a space point by looking up the table.

3.3 Calibration of Vehicle Odometer

Autonomous vehicles or mobile robots usually suffer from accumulations of mechanical errors during navigation. The vehicle we used in this study has been found to have a defect of this kind that when it moves forward, the vehicle deviates leftward from the original path gradually, and the position and the direction angle of the vehicle provided by the odometer are different from the real position and direction angle of the vehicle. Because the mechanic errors of an odometer usually will lead to wrong control instructions and inaccurate vehicle localization information, it is desired to conduct a calibration task to eliminate such errors.

Chen and Tsai [10] proposed a mechanical error correction method by curve-fitting the erroneous deviations from correct paths. In this study, we use this method for correcting the vehicle odometer readings. In Section 3.3.1, we describe how we find out the deviation values in different distances and build an odometer calibration model for them. In Section 3.3.2, we will derive an odometer reading correction equation by a curve fitting technique. Finally, the use of such error correction results is described in Section 3.3.3.

3.3.1 Odometer Calibration Model

In this section, first we describe an experiment we conducted to record the vehicle's deviations from a planned navigation path at different distances from a navigation starting point. We attach a stick on the frontal of vehicle and let it point to the ground in such a way that the stick is perpendicular to not only the V_y-axis in the VCS but also the ground. The stick and the two contact points of the two driving wheels with the ground are used to find out the origin of the VCS. The equipments used in this experiment were a measuring tape, two laser levelers, and the autonomous vehicle. We chose a point from the crossing points formed by the boundaries of the rectangular-shaped tiles on the floor, as the initial position O of vehicle navigation; and mark the positions by pasting a sticky tape on the ground. Also, we designated a straight line L along the tile boundaries from O.

First, we used the two laser levelers to adjust the position of the vehicle by hand such that the two contact points of the two driving wheels with the ground lie on the boundaries of the tiles and that the stick on the vehicle is perpendicular to L. Second, we drove the vehicle forward for a desired distance on a straight line L. Third, we marked the terminal position T reached by the vehicle by finding out the origin in the VCS. Fourth, we measured the perpendicular distance between T and L. Finally, we repeated the steps at least five times for each desired distance. An illustration of the experiment is shown in Figure 3.8, a detailed algorithm for the process is described in the following, and the experimental results are shown in Table 3.2.

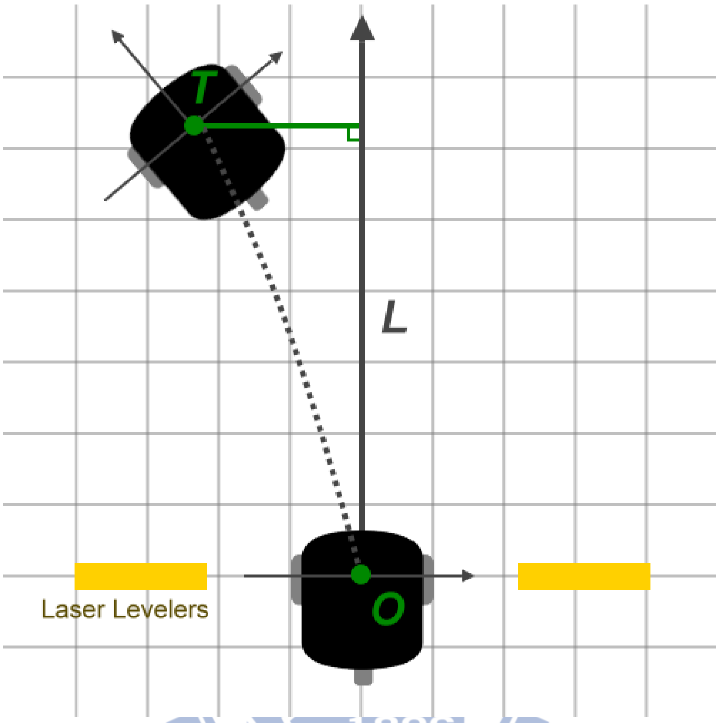


Figure 3.8 Illustration of the experiment.

Algorithm 3.1. Building a calibration model of the odometer.

Input: none

Output: A calibration model of the odometer.

Steps:

Step 1. Choose an initial point *O* from the crossing points of the boundaries of the rectangular-shaped tiles on the floor, and designate a straight line *L* which starts at *O* and coincides with some of the tile boundaries.

Step 2. Use the laser levelers to check if the two contact points of the two driving wheels with the ground lie on the boundaries; and if not, adjust vehicle pose such that it coincides with the straight line *L*.

Step 3. Drive the vehicle to move forward until it arrives at a desired distance on a straight line *L*.

Step 4. Mark the terminal position *T* reached by the vehicle.

Step 5. Measure the distance between *T* and *L*.

Step 6. Repeat Steps 2 through 5 at least five times for each desired moving distance and record the results.

Table 3.2 The results of the experiment of building an odometer calibration model.

Move Distance (cm)	Average Distance of Deviation (cm)
30	0.31
40	0.41
50	0.83
60	0.88
70	0.93
80	1.38
90	1.93
100	2.14
110	2.55
120	2.70
130	2.90
140	3.2
150	3.32

3.3.2 Curve Fitting

The distribution of the deviations of the experimental results in Section 3.3.1 is shown in Figure 3.9. We found out that the distribution of the data can be described as a curve as a mechanical-error calibration model for the vehicle. Therefore, we use the least-squares-error (LSE) curve fitting technique to obtain a third-order equation to fit these data. After that, we can use the derived curve equation to calibrate the vehicle odometer during navigation sessions.

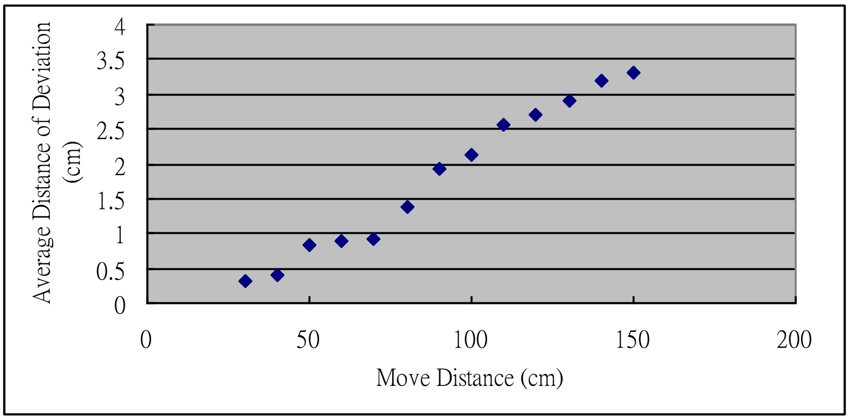


Figure 3.9 The distribution of the deviations.

The principle of the curve fitting technique we adopt can be explained as follows. We consider a polynomial function *L* of degree k:

$$L: y = a_0 + a_1 x + \ldots + a_k x^k . \tag{3.22}$$

And with the pairs of measured data, (x_1, y_1), (x_2, y_2), …, (x_n, y_n) mentioned above, we can use the LSE principle to fit the pairs by

$$E = \sum_{i=1}^{n} [y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)]^2 . \tag{3.23}$$

The partial derivatives of *E* with respect to a_0, $a_{1,}$ …, and a_k, respectively, are

$$\frac{\partial E}{\partial a_t} = -2\sum_{i=1}^{n} [y_i - (a_0 + a_1 x_i + \ldots + a_k x_i^k)] \frac{\partial y}{\partial a_t} = 0, \forall t = 0, 1, \ldots, n . \tag{3.24}$$

From Eq. (3.24), we can derive a relation matrix as follows:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \cdots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \cdots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_1 \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^k y_i \end{bmatrix} . \tag{3.25}$$

In order to solve the matrix equation, we rewrite Eq. (3.22) into a matrix form *Y*=*XA*:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}, \tag{3.26}$$

and we multiply Eq. (3.26) by the transpose of the matrix X as follows:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}, \tag{3.27}$$

which may be simplified to get the following matrix equation:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \dots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \dots & \sum_{i=1}^{n} x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \dots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_1 \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^k y_i \end{bmatrix}. \tag{3.28}$$

It is true that Eqs. (3.28) and (3.25) are identical, and as a result, solving Eq. (3.26) is equivalent to solving Eq. (3.25). The deviation processes from Eqs. (3.26) to (3.28) can be rewritten in the notation form respectively as follows:

$$Y = XA; \tag{3.29}$$

$$X^T Y = X^T X A; \tag{3.30}$$

$$A = (X^T X)^{-1} X^T Y. \tag{3.31}$$

The numerical result of curve fitting for the calibration model using that data shown in Table 3.2 is shown below and illustrated in Figure 3.10:

$$f(x) = -0.0000014478x^3 + 0.000379467x^2 - 0.002033815x - 0.000081164. \tag{3.32}$$

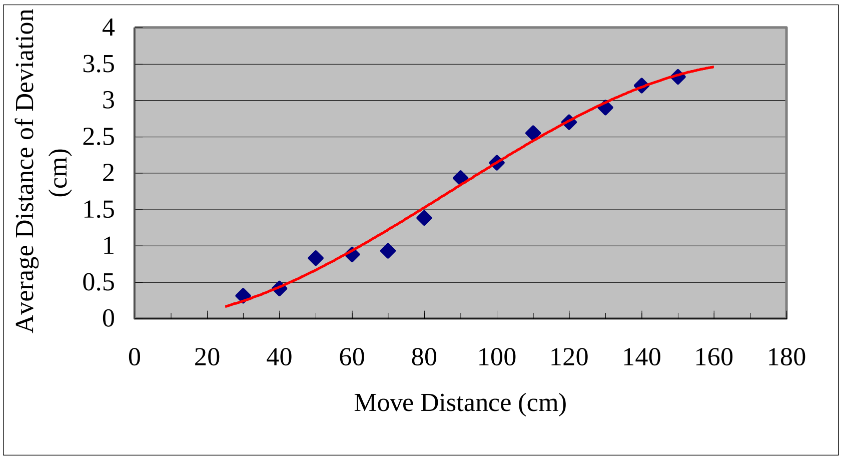


Figure 3.10 The result of curve fitting of the deviations with order $k=3$.

3.3.3 Correction of vehicle position coordinates

The goal to correct the mechanical error of the vehicle now can be carried out by the curve equation derived above. As shown in Figure 3.11, the vehicle starts moving forward from $P_1(x_1, y_1)$ toward the goal $P_2(x_2, y_2)$ in the GCS. Denote the direction of the vehicle at P_1 as θ. Assume that due to mechanical errors, the vehicle arrives at P_2' which is different from the desired goal P_1. To get the position of the vehicle, or equivalently, to get P_2', at first we compute the distance D used to estimate the deviation distance d_y as follows:

$$D=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}\,. \tag{3.33}$$

By substituting D into the curve equation $y=ax^3+\ bx^2+\ cx\ +d$, d_y is derived by

$$d_y=aD^3+bD^2+cD+d\,. \tag{3.34}$$

Subsequently, according to the involved simple geometry we can derive the following equations to compute the position of P_2':

$$x_2'=D\cos\theta-d_y\sin\theta+x_1'\,; \tag{3.35}$$

$$y_2'=D\sin\theta+d_y\cos\theta+y_1'\,. \tag{3.36}$$

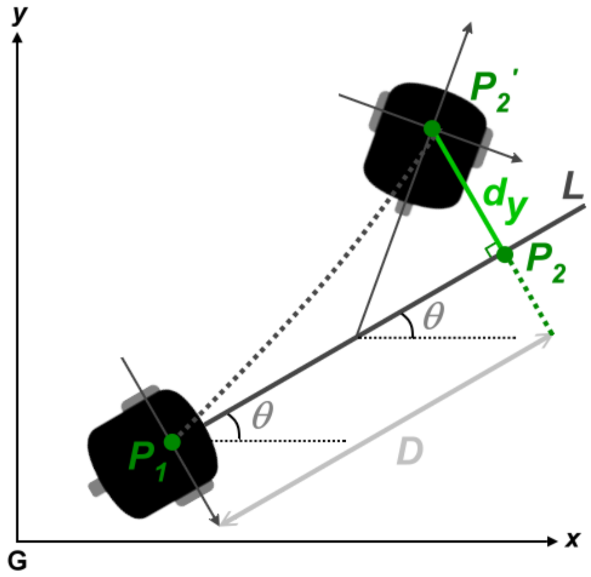


Figure 3.11 An illustration of coordinate correction.

The process to correct the position of the odometer during vehicle navigation now can be summarized as the following steps:

1. before driving the vehicle straight forward, compute the distance *D* by Eq. (3.33) according to the position of the desired goal at the current navigation cycle, and get also from the odometer of the vehicle the current direction θ with respect to the *x*-axis in the GCS;
2. compute the deviation d_y by substituting the value *D* into Eq. (3.32);
3. calculate the corrected position of the vehicle according to Eqs. (3.35) and (3.36); and
4. update the vehicle odometer with the corrected position.

Chapter 4
Vehicle Navigation by Mopboard Following Using a Look-up Pano-mapping Table

4.1 Introduction

As mentioned previously, it is desired to conduct the house-layout construction task by driving the vehicle around the room and collect environment data. More specifically, the major stages the house-layout construction task will be described in Section 4.2. Most houses have mopboards at the roots of their walls. The mopboard feature is used for vehicle navigation in this study. We will describe the principle of the method we use to detect mopboard edges in omni-images in Section 4.3. In Section 4.4, we will describe the details of the proposed method. And the entire navigation strategy will be described in Section 4.5.

4.2 3D House-layout Construction by Vehicle Navigation

The major stages in proposed 3D house-layout construction by vehicle navigation include: (1) vehicle navigation by mopboard following; (2) floor-layout construction; and (3) 3-D house-layout construction. The details are described as a three-stage procedure in the following.

Stage 1 Vehicle navigation by mopboard following ---

1. Guide manually the vehicle to a start position for starting a navigation session.
2. (*Adjustment of the vehicle direction to be parallel to the nearby wall*) Perform the following steps to adjust the initial moving direction of the vehicle to be parallel to the nearest wall.

    2.1 Take an image I_0 of the nearby scene using the lower omni-camera.

    2.2 Define an *angular scanning rang*e R_0 which covers the left part of the image in which the wall presumably appears (with R_0 being from 145^{o} to 225^{o} with 0^{o} pointing to the right of the image and the angle counted in a counterclockwise direction).

    2.3 Find mopboard edge points within the scanning range R_0 in I_0 by the previously-mentioned mopboard edge point detection process (Section 4.4) to get a point set M_0.

    2.4 Fit the points in M_0 with a line L_0 by the LSE curve fitting scheme described in Section 3.3.2.

    2.5 Adjust the direction of the vehicle to the direction of L_0 (the detail is described in Section 4.5.3).

3. (*Along-wall vehicle navigation*) Perform the following steps to conduct the vehicle to navigate along wall sides and collect environment data.

    3.1 Take two images I_1 and I_2 of the nearby scene using the lower and the upper omni-cameras, respectively; and store the images.

    3.2 Define three angular scanning ranges R_L, R_R, and R_F, which cover the left, right, and front parts of image I_1, respectively, in which the walls presumably appear (with R_L being from 145^{o} to 225^{o}, R_R being from -20^{o} to 20^{o}, and R_F being from 30^{o} to 150^{o} with 0^{o} pointing to the right of the image I_1).

    3.3 Find mopboard edge points in I_1 using R_L, R_R, and R_F respectively by the previously-mentioned mopboard edge point detection process (Section 4.4) to get three point sets S_L, S_R, and S_F.

3.4 (*Transformation of the detected edge points into useful navigation information and environment data*) Perform the following steps to deal with the detected edge points.

3.4.1 Apply a pattern classification process (Section 4.5.2) on each of point sets from S_L, S_R, and S_F to find a corner edge point if it exits.

3.4.2 Based on the found corner point as a division point in collected mopboard edge points, assign the points of S_L, S_R, and S_F which are the edge points of the left, the right, and the front walls, respectively, into the corresponding sets S_L', S_R', and S_F'.

3.4.3 Add the points of S_L' into the wall data set $WallData_{index}$ according the index of the wall.

3.5 (*Detection of possible collision with a wall in front of the vehicle*) Perform the following steps to determine the distances between the walls and the vehicle.

3.5.1 Estimate the distances $dist_{L'}$, $dist_{R'}$, and $dist_{F'}$ between walls and the vehicle according to the points of S_L', S_R', and S_F', respectively.

3.5.2 Check if the vehicle is *moveable* further according to data of $dist_{L'}$, $dist_{R'}$, and $dist_{F'}$ (the detail is described in Section 4.5.4):

(1) if yes, then go to Step 3.6;

(2) else, terminate the vehicle navigation procedure.

3.6 Continue the vehicle navigation process.

3.7 Check if the vehicle arrives at the start position:

(1) if yes, exit Stage 1 and enter Stage 2.

(2) else, go to Step 2.

Stage 2 Floor-layout construction ---

4. (*Curve fitting for each segment*) Perform the following steps to fit the detected edge points, where *indexes* are used to label the walls.

4.1 Fit the points in $WallData_{index}$ with a line L_{index} by the LSE curve fitting scheme (Section 3.3.2).

4.2 Repeat Step 4.1 until all walls have corresponding fitting line equations.

4.3 Adjust the fitted lines by a global LSE optimization method to establish a floor-layout (the detail is described in Section 5.2).

4.4 Denote each edge of the floor-layout as LF_{index}.

Stage 3 3-D House-layout construction ---

5. (*Detection of objects on the walls*) Perform the following steps to detect objects on the walls.

5.1 For each floor-layout edge LF_{index}, perform the following steps.

5.1.1 Fetch the upper and lower omni-images I_{1k} and I_{2k} taken in the previous steps during vehicle navigation, which belong to LF_{index}.

5.1.2 Define a scanning range R_{1k} for I_{1k} and R_{2k} for I_{2k} according to the floor-layout edge segment widths.

5.1.3 Extract objects appearing in I_{1k} within the scanning range R_{1k} and those in I_{2k} within the scanning range R_{2k}, and store the detected objects with related information into the sets $wall\_info\_1_{index}$ and $wall\_info\_2_{index}$, respectively.

5.1.4 Repeat Steps 5.1.1 through 5.1.3 until all the omni-images which belong to LF_{index} are scanned.

5.2 Repeat 5.1 until all the floor-layout edges are exhausted.

6. (*Combination of objects on the walls*) Perform the following steps to combine the detected objects on the walls.

6.1 For each floor-layout edge LF_{index}, perform the following steps.

6.1.1 Combine the objects in $wall\_info\_1_{index}$ and $wall\_info\_2_{index}$, respectively according to their positions, resulting in two sets $wall\_info1_{index}'$ and $wall\_info2_{index}'$ of the combined objects, respectively.

6.2 Repeat 6.1 until all the floor-layout edges LF_{index} are exhausted.

7. (*Objects reorganization*) Perform the following steps to recognize doors and windows on the walls.

7.1 For each floor-layout edge LF_{index}, perform the following steps.

7.1.1 Choose an object $obj2_k'$ from $wall\_info2_{index}'$, check if there exists an object in $wall\_info1_{index}'$ with at a similar location.

(1) If exists such an object $obj1_k'$, check again if $obj1_k'$ is connected to the mopboard:

i. if yes, then recognize $obj1_k'$ together with $obj2_k'$ as a door, and add it to the set obj_door_{index};

ii. otherwise, recognize $obj1_k'$ together with $obj2_k'$ as a window, and add it to the set obj_window_{index}.

(2) If such an object does not exist, recognize $obj2_k'$as a window, and add it to the set obj_window_{index}.

7.1.2 Repeat 7.1.1 until the objects of $wall\_info2_{index}'$ are exhausted.

7.1.3 Recognize the remaining objects in $wall\_info1_{index}'$ as a window, and add it to the set obj_window_{index}.

7.2 Repeat 7.1 until all the floor-layout edge LF_{index} are exhausted.

8. (*House-layout construction*) Perform the following steps construct a 3-D house-layout.

8.1 For each floor-layout edge LF_{index}, perform the following steps.

8.1.1 Construct the wall $wall_{index}$ and draw on the wall the objects in obj_window_{index} and obj_door_{index} if any exits.

8.2 Repeat 8.1 until all the floor-layout edge LF_{index} are exhausted.

The details of the stages and the steps will be described subsequently in this chapter and in the next chapter.

4.3 3-D Data Estimation by Using the Proposed Imaging System

As mentioned previously, the positions of mopboard edge points are important information for vehicle navigation and the 3-D house layout construction. In this section, we will describe how the imaging system we proposed is used to estimate the positions of concern objects. By finding out the relations among the coordinate system used in this study, the estimated results can be transformed among different coordinate systems. The coordinate systems are described in Section 4.3.1. And the details of the 3-D data estimation process are described in Section 4.3.2.

4.3.1 Coordinate systems

The major coordinate systems used in this study have been described in Section 3.1.1. However, as shown in Figure 4.1, the vertical line L_o, which is perpendicular to the ground and pass though both of the origins of the CCS$_1$ and CCS$_2$, does not pass through the origin V of the VCS. For convenience, a local camera coordinate system CCS$_{local}$ which is defined in the following is used to describe the positions of the concerned objects in the GCS estimated by the imaging system.

The coordinates of the local camera coordinate system CCS$_{local}$ are denoted as (X_{local}, Y_{local}, Z$_{local}$) subsequently. The origin O_{local} of the CCS$_{local}$ is the intersection point of the vertical line L_o and the ground. The X_{local}-axis and the Y_{local}-axis are on the ground. The Y_{local} coincides with the V_y-axis of the VCS and the X_{local}-axis is parallel to the V_x-axis of the VCS. The Z_{local}-axis coincides with both the Z_1-axis and Z_2-axis

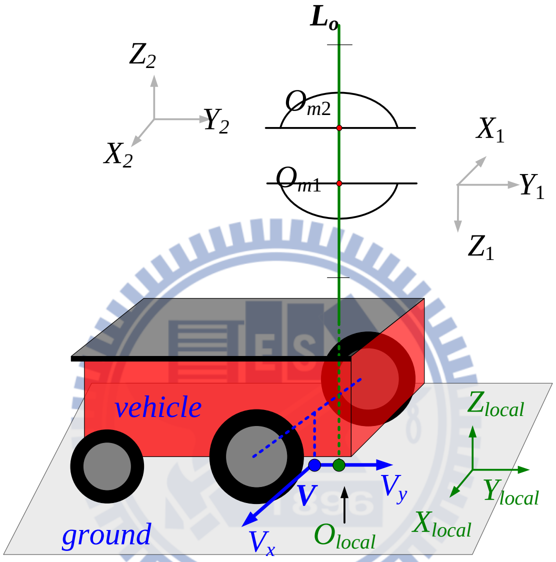


Figure 4.1 The relation between the VCS, the CCS$_1$, and the CCS$_2$.

We denote the length $\left|\overrightarrow{VO_{local}}\right|$ as l_{bias}. The relation among the GCS, the VCS, and the CCS$_{local}$ is illustrated in Figure 4.2. Assume that (x_p, y_p) represent the coordinates of the vehicle in the GCS, and that the relative rotation angle denoted as θ is the directional angle between the positive direction of the x-axis in the GCS and the positive direction of the V_x-axis in the VCS. Assume also that (x_c, y_c) represent the coordinates of the origin of the CCS$_{local}$ in the GCS. The coordinate transformation among the GCS, the VCS, and the CCS$_{local}$ can be described by the following equations:

$$V_x = X_{local}; \tag{4.1}$$

$$V_y = Y_{local} + l_{bias}. \tag{4.2}$$

Substituting Eqs. (4.1) and (4.2) into Eqs. (3.1) and (3.2), we get

$$\begin{aligned} x &= V_x \times \cos\theta - V_y \times \sin\theta + x_p \\ &= X_{local} \times \cos\theta - (Y_{local} + l_{bias}) \times \sin\theta + x_p; \end{aligned} \tag{4.3}$$

$$\begin{aligned} y &= V_x \times \sin\theta + V_y \times \cos\theta + y_p \\ &= X_{local} \times \sin\theta + (Y_{local} + l_{bias}) \times \cos\theta + y_p \end{aligned}. \tag{4.4}$$

And substituting the coordinates (X_{local}, Y_{local}) = (0, 0) into Eqs. (4.3) and (4.4), we get

$$x_c = -l_{bias} \times \sin\theta + x_p; \tag{4.5}$$

$$y_c = l_{bias} \times \cos\theta + y_p; \tag{4.6}$$

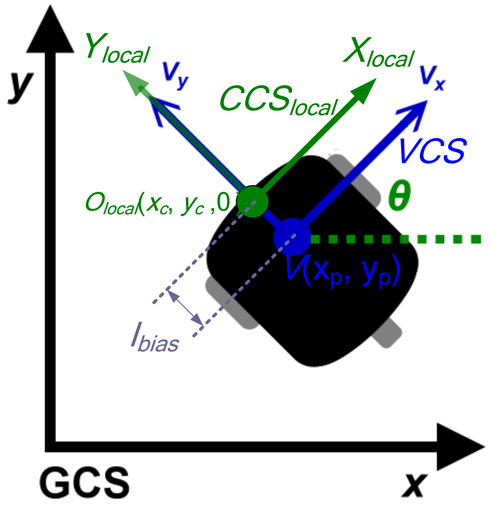


Figure 4.2 The relation among the GCS, the VCS, and the CCS$_{local}$.

4.3.2 3-D Data Acquisition

In this section, we describe how the two-camera imaging system proposed is used in this study to estimate the positions of concerned space points. Note that the two omni-cameras are connected and vertically-aligned in a bottom-to-bottom fashion, which means that none of the space points can be projected on both image planes at the same time. As a result, the most common stereo matching method, which is based on the triangulation principle obtain the relation between a 3-D point and multiple cameras, does not work at all here. We need another theory to deal with the stereo matching problem here.

One way proposed in this study is to estimate the horizontal distance to the desired object point, and subsequently to utilize both the distance and elevation angle of the space point with respect to the CCS$_{local}$ to get relevant 3-D data. As shown in Figure 4.3, assume that both space points P_1 and P_2 lie on the vertical line L which is perpendicular to the x-y plane in the GCS and that P_1 is the intersection point of L and the x-y plane in the GCS. Assume also that P_1 and P_2 lie on the light rays with the azimuth-elevation angel pairs of (θ_1, ρ_1) and (θ_2, ρ_2) with respect to the CCS$_1$ and the CCS$_2$, respectively, where O_{m1} is the focal point of the lower mirror, and O_{m2} is the focal point of the upper mirror. Assume finally that the point O_{local} with coordinates (0, 0, 0) is the origin of the CCS$_{local}$.

Referring to Figure 4.3 and by the triangulation principle, we have

$$D = \frac{MH_1}{\tan \alpha_1}, \tag{4.7}$$

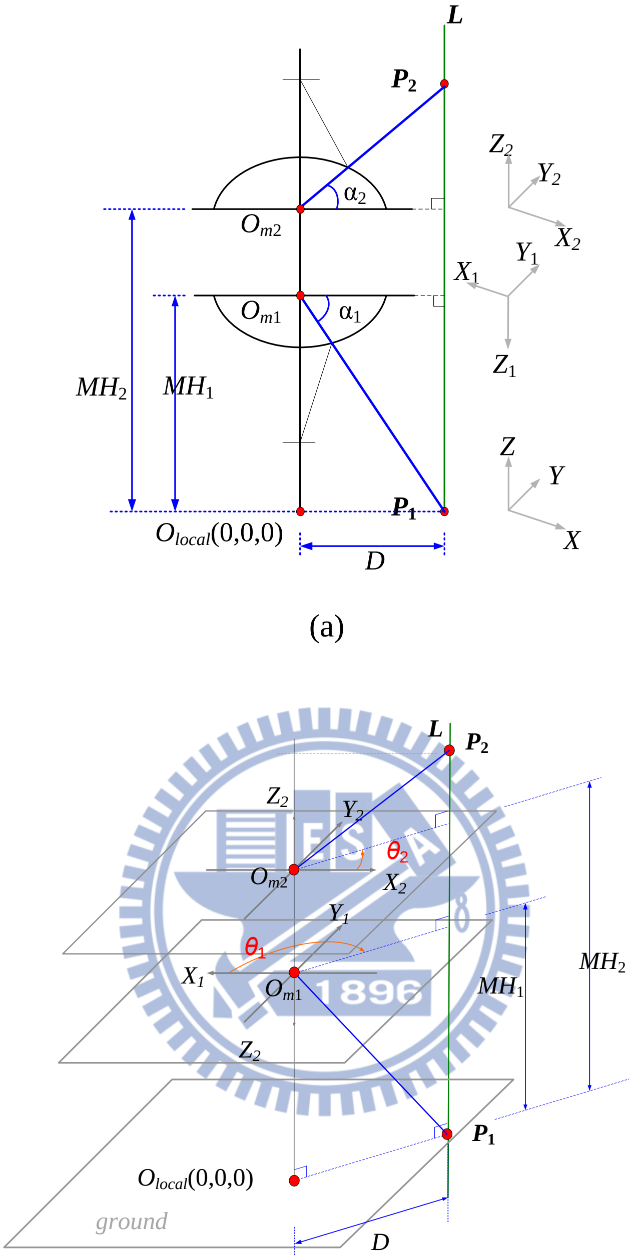
where D is the horizontal distance between O_{local} and P_1, and MH_1 is the height of the focal point O_{m1}. As a result, the height H_2 of the space point P_2 can be computed by

$$H_2 = MH_2 + D \times \tan \alpha_2, \tag{4.8}$$

where MH_2 is the height of the focal point O_{m2}.

According to the rotational invariance property of the omni-camera, we can derive the following equations:

$$\cos\theta_1 = \frac{X_1}{\sqrt{X_1^{\,2} + Y_1^2}} = \frac{u_1}{\sqrt{u_1^{\,2} + v_1^{\,2}}}; \tag{4.9}$$



(a)

(b)

Figure 4.3 (a) Computation of depth using the two-camera omni-directional imaging system. (b) Details of (a).

$$\sin\theta_1 = \frac{Y_1}{\sqrt{X_1^{\,2}+Y_1^2}} = \frac{v_1}{\sqrt{u_1^{\,2}+v_1^{\,2}}}\,; \tag{4.10}$$

$$\cos\theta_2 = \frac{X_2}{\sqrt{X_2^{\,2}+Y_2^{\,2}}} = \frac{u_2}{\sqrt{u_2^{\,2}+v_2^{\,2}}}\,; \tag{4.11}$$

$$\sin\theta_2 = \frac{Y_2}{\sqrt{{X_2}^2 + {Y_2}^2}} = \frac{v_2}{\sqrt{{u_2}^2 + {v_2}^2}}. \tag{4.12}$$

Because the vertical line *L* is perpendicular to the X_1-Y_1 and X_2-Y_2 planes and the positive direction of the X_1-axis in the CCS$_1$ is opposite to the positive direction of the X_2-axis in the CCS$_2$, it can be found that $X_1 = -X_2$ and

$$\cos\theta_1 = \frac{X_1}{\sqrt{{X_1}^2 + Y_1^2}} = \frac{u_1}{\sqrt{{u_1}^2 + {v_1}^2}} = \frac{-X_2}{\sqrt{{X_2}^2 + Y_2^2}} = \frac{-u_2}{\sqrt{{u_2}^2 + {v_2}^2}} = -\cos\theta_2; \tag{4.13}$$

$$\sin\theta_1 = \frac{Y_1}{\sqrt{{X_1}^2 + Y_1^2}} = \frac{v_1}{\sqrt{{u_1}^2 + {v_1}^2}} = \frac{Y_2}{\sqrt{{X_2}^2 + Y_2^2}} = \frac{u_2}{\sqrt{{u_2}^2 + {v_2}^2}} = \sin\theta_2. \tag{4.14}$$

Now, we can utilize Eqs. (4.7), (4.8), (4.13), and (4.14) to calculate the coordinates $(X_{p_1}, Y_{p_1}, Z_{p_1})$ of P_1 with respect to the CCS$_{local}$ as follows:

$$\begin{aligned} X_{p_1} &= -D\times\cos\theta_1 = -\frac{MH_1}{\tan\alpha_1}\times\cos\theta_1; \\ Y_{p_1} &= \quad D\times\sin\theta_1 = -\frac{MH_1}{\tan\alpha_1}\times\sin\theta_1; \\ Z_{p_1} &= \quad 0, \end{aligned} \tag{4.15}$$

and the coordinates $(X_{p_2}, Y_{p_2}, Z_{p_2})$ of P_2 with respect to the CCS$_{local}$ as follows:

$$\begin{aligned} X_{p_2} &= D\times\cos\theta_2 = \frac{MH_1}{\tan\alpha_1}\times\cos\theta_2 = -X_{p_1}; \\ Y_{p_2} &= \ D\times\sin\theta_2 = \frac{MH_1}{\tan\alpha_1}\times\sin\theta_2 = \ Y_{p_1}; \\ Z_{p_2} &= H_2 = MH_2 + D\times\tan\alpha_2 = MH_2 + \frac{MH_1}{\tan\alpha_1}\times\tan\alpha_2. \end{aligned} \tag{4.16}$$

4.4 Mopboard Detection and Location Estimation

Overall, before the navigation session is stated, a process to detect mopboard edges and to estimate their positions designed in this study is applied. One reasonable assumption is made here, that is, there is no obstacle on the uniform floor during the detection process. The process of mopboard detection in the omni-images is described in Section 4.4.1. With the detected edge pixel coordinates available, a location estimation process also designed in this study is applied subsequently, which is described in Section 4.4.2. A flowchart of the mopboard detection and location estimation processes is illustrated in Figure 4.4.

4.4.1 Proposed method for mopboard detection

According to the rotational invariance property of the omni-image, all points lying on each straight line *L*, which is parallel to the Z_1-axis in the CCS$_1$, are projected onto the image plane of the ICS$_1$ with the azimuth θ_1, forming a straight line in the ICS$_1$ and passing the origin I_1, as shown in Figure 4.5. And because mopboards occupy the bottom parts of walls, it results in a fact that the projected pixels of the mopboards on the omni-image also have an obvious band. We utilize these two properties for finding out the mopboard edge points which are located on the edge between the mopboard and the floor.

With the assumption that the house is empty, we can transform an acquired grayscale omni-image into a binary one by a pre-defined threshold and then perform erosion and dilation operation on it. In this way, the mopboard will remain in the image and the mopboard edge points can be detected by using a scanning line starting from the image center. An algorithm for the detection of the mopboard edges is described as follows.

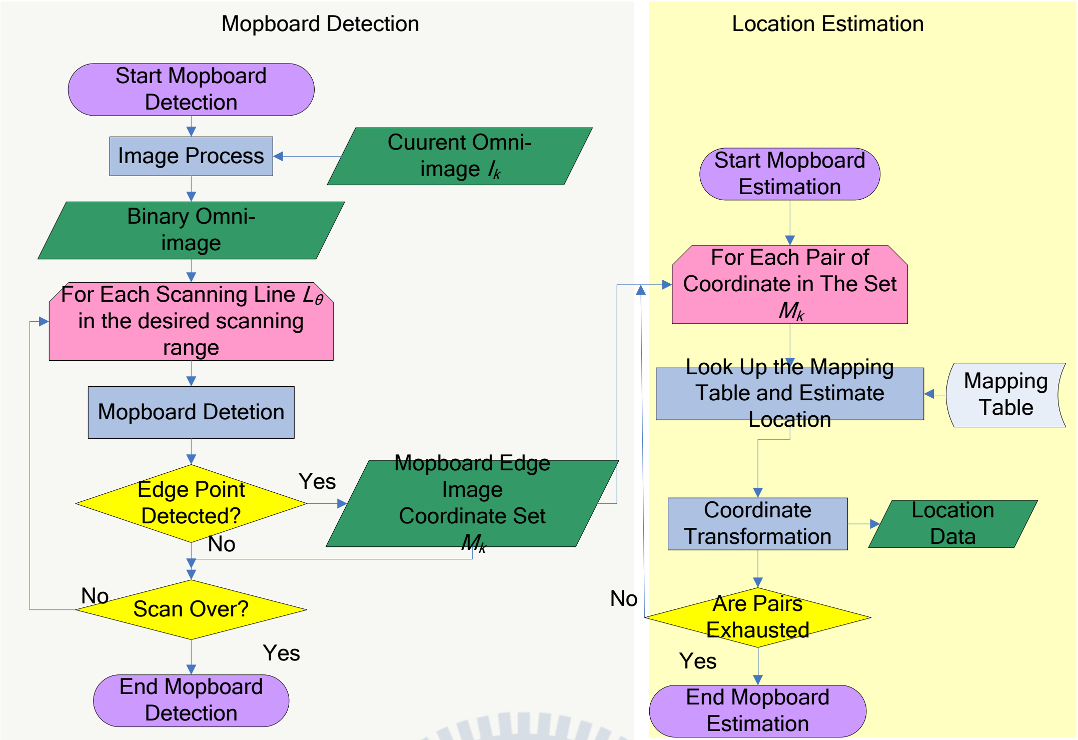


Figure 4.4 Flowchart of the mopboard detection and location estimation processes.

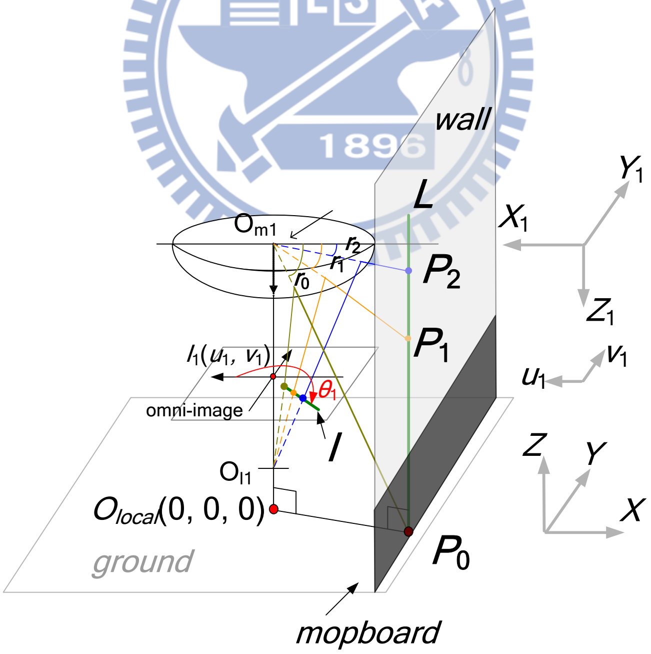
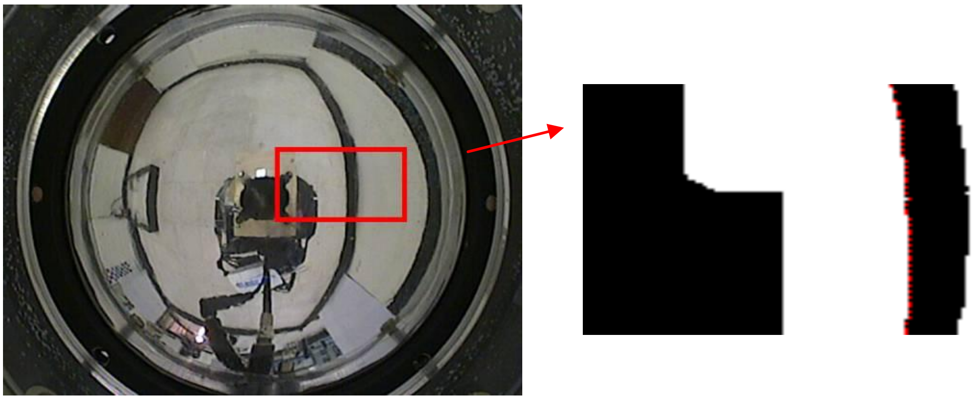


Figure 4.5 The vertical line L in the OCS$_{local}$ and the corresponding line l in the ICS$_1$.



(a) (b)

Figure 4.6 Mopboard edge detecting in an omni-image. (a) Original omni-image. (b) Detected edge pixels (in red color).

Algorithm 4.1 Mopboard detection.

Input: An omni-image I_k taken from the lower omni-camera; a desired scanning range; and a vehicle *mask*.

Output: A set M_k of image coordinates of mopboard edges.

Steps:

Step 1. Transform I_k into a gray version I_G.

Step 2. Reset the gray values g_{pi} of each pixel p_i in the gray image I_G by comparing g_{pi} with a pre-defined threshold T as follows:

if $g_{pi} \geq T$, set g_{pi} as a white pixel;

else, set g_{pi} as a black pixel.

Step 3. Perform erosion on I_G to obtain a new image I_G'.

Step 4. Perform dilation on I_G' to obtain a second new image I_G''.

Step 5. For each scan line L_θ in a pre-selected angle scanning range with certain angular scanning steps, perform the following steps.

5.1 Start from the image center of I_G'', traverse along the line L_θ, and find the first intersection black pixel p of I_G'' with L_θ.

5.2 Check if p is followed by 10 or more consecutive black pixels along L_θ: if so, then add the coordinates of p into M_k (which is set empty initially).

Step 6. Repeat Step 5 until the scan range is exhausted.

4.4.2 Mopboard edge points location estimation

After detecting the mopboard edge points, we have the set M_k which includes the image coordinates of the detected mopboard edge points. Denote the set M_k in more detail by

$$M_k = \{(u_t, v_t) \mid (u_t, v_t) \text{ is the detected mopboard edge coordinates in ICS}_1\}. \qquad (4.17)$$

In order to estimate the corresponding locations of these detected edge points, we first choose an element (u_t, v_t) from M_k and calculate the corresponding radius r_t and azimuth θ_t by transforming (u_t, v_t) into the corresponding polar coordinates (r_t, θ_t). Then, we determine a column index of the pano-mapping table by finding out which *azimuth interval* the θ_t lies on. For example, if there exists i satisfying the following equation:

$$\theta_i = i \times (2\pi / M) \le \theta_t < (i+1) \times (2\pi / M) = \theta_{i+1}, \qquad (4.18)$$

then the column index is taken to be i. Also, we compare r_t with r_s in the entry E_{is} of the pano-mapping table, and keep the nearest radius r_j. As a result, the corresponding ρ_j can be found out. Once the ρ_j is known, the 3-D position estimation task described in Section 4.3 can be carried out. In summary, the process for location estimation is described as follows.

Algorithm 4.2: Mopboard edge location estimation.

Input: A set M_k of mopboard edges specified by image coordinates.

Output: The corresponding mopboard edge locations in the GCS.

Steps:

Step 1. Take a mopboard edge point t with image coordinates (u_t, v_t) from the set M_k.

Step 2. Use (u_t, v_t) to look up the pano-mapping table to find the corresponding elevation angel ρ_t.

Step 3. Calculate the coordinates of point t in the CCS$_{local}$ by Eq. (4.15).

Step 4. Transform the coordinates from the CCS$_{local}$ to the GCS by Eqs. (4.3) and (4.4).

Step 5. Repeat Step 1 until all the mopboard edge points in M_k are exhausted.

4.5 Vehicle Navigation Strategy

When the vehicle starts navigation, how to transform the detected mopboard edge points into correct navigation information and environment data is an important issue. In Section 4.5.1, we will describe the main idea of the navigation strategy adopted in this study. The vehicle navigates based on keeping a fixed distance to each wall all the time; therefore, it is indispensable to assign each extracted mopboard edge point correctly to the wall adjacent to the mopboard. We propose a pattern classification technique to accomplish the task. The classification process is described in Section 4.5.2. In order to keep the vehicle's navigation path parallel to each wall, we use line fitting to adjust the direction of the vehicle. The adjustment process is described in Section 4.5.3. The proposed navigation process is described in Section 4.5.4.

4.5.1 Idea of proposed navigation strategy

For vehicle navigation, the distance to the wall is important navigation information. The main idea is to navigate and keep the navigation path parallel to the wall. In this way, we choose three regions *Region*$_L$, *Region*$_R$, and *Region*$_F$ on the CCS$_{local}$, as shown in Figure 4.7, so that the walls which are close to the vehicle can be detected. And we define the three sets S_L, S_R , and S_F by

$$\begin{aligned} S_L &= \{P_L \mid P_L \in M_{detected} \text{ and } P_L \in Region_L\} \\ S_R &= \{P_R \mid P_R \in M_{detected} \text{ and } P_R \in Region_R\} , \\ S_F &= \{P_F \mid P_F \in M_{detected} \text{ and } P_F \in Region_F\} \end{aligned} \tag{4.19}$$

where the set $M_{detected}$ includes the detected mopboard edge points with coordinates in the CCS$_{local}$.

However, for example, the edge points which are on the left wall with respect to the vehicle may appear both in *Region*$_L$ and in *Region*$_F$ at the same time, resulting in the wrong distance estimation. In order to transform the detected mopboard edge point locations into useful navigation information and environment data, a pattern classification technique is proposed and described subsequently. Once we can assign each of the edge points to its corresponding wall by this classification technique, we can adjust the direction of the vehicle to keep the vehicle's navigation path parallel to each wall and estimate the distances to the nearby walls. Let the new sets derived from Eq. (4.19) after the pattern classification process be described as follows:

$$\begin{aligned} S_L' &= \{P_L' \mid P_L' \in S_L \text{ and } P_L' \in Wall_L\}; \\ S_R' &= \{P_R' \mid P_R' \in S_R \text{ and } P_R' \in Wall_R\}; . \\ S_F' &= \{P_F' \mid P_F' \in S_F \text{ and } P_F' \in Wall_F\}. \end{aligned} \tag{4.20}$$

As a result, the distances to each wall can be estimated more accurately, and are used as navigation information to guide the vehicle.

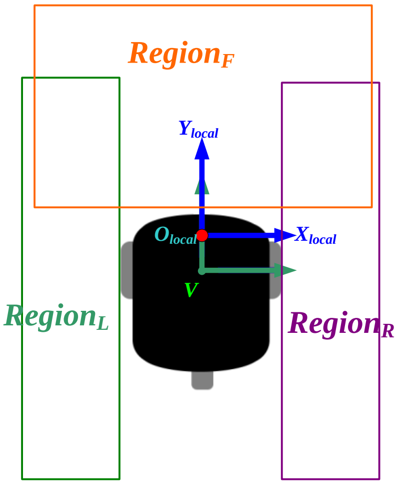


Figure 4.7 An illustration of the pre-defined regions.

4.5.2 Proposed pattern classification technique of mopboard data

As mentioned above, the estimated positions of the detected mopboard edge points are derived in an order according to the scanning order in the omni-image. Based on this order, the classification process would be simplified because any of these estimated points can be easily divided into two parts.

The purpose of using a pattern classification technique in this study is to find an edge point locating at the corner of two adjacent walls that can divide the edge points into two parts. With a property that the adjacent walls are perpendicular to each other, we can conduct a pattern classification process by the LSE criterion to find such a corner point, if it exists.

Let such a corner point, denoted as S_k, divides the set of edge points in different walls into two sets, denote as S_{k1} and S_{k2}. A reasonable assumption is made here that the two lines which are used to fit the points in S_{k1} and S_{k2} by the LSE criterion, respectively, are perpendicular to each other. In this way, such a corner point will yield a minimum error. The detailed process for finding the corner point is described in the following.

At first, we describe the set $M_{detected}$ more specifically by

$$M_{detect} = \{m_{\theta 1}, m_{\theta 2}, \ldots, m_{\theta n}\}, \tag{4.21}$$

where $m_{\theta t}$ means the estimated point with the scanning azimuth angle θ_t and $\theta_1 < \theta_2 < \ldots < \theta_n$. And we rewrite Eq. (4.19) as follows:

$$\begin{aligned} S_L &= \{P_L \mid P_L \in M_{detected} \text{ and } P_L \in Region_L\} = \{P_{L_{\theta 1}}, P_{L_{\theta 2}}, \ldots, P_{L_{\theta n_L}}\} \\ S_R &= \{P_R \mid P_R \in M_{detected} \text{ and } P_R \in Region_R\} = \{P_{R_{\theta 1}}, P_{R_{\theta 2}}, \ldots, P_{R_{\theta n_R}}\}, \\ S_F &= \{P_F \mid P_F \in M_{detected} \text{ and } P_F \in Region_F\} = \{P_{F_{\theta 1}}, P_{F_{\theta 2}}, \ldots, P_{F_{\theta n_F}}\} \end{aligned} \tag{4.22}$$

where $L_{\theta 1} < L_{\theta 2} < \ldots < L_{\theta n_L}$, $R_{\theta 1} < R_{\theta 2} < \ldots < R_{\theta n_R}$, and $F_{\theta 1} < F_{\theta 2} < \ldots < F_{\theta n_F}$. Then, we apply the classification task on the above sets. For example, if we want to classify the set S_L, the main idea is (1) to find a point $P_{L\theta t}$ which belongs to the set S_L that divides the set S_L into two sets, S_{L1_t} and S_{L2_t} as follows:

$$S_{L1\_t} = \{P_{L\theta 1}, \ldots, P_{L\theta t}\},\ S_{L2\_t} = \{P_{L\theta t}, \ldots, P_{L\theta n_L}\}, \tag{4.23}$$

and (2) to calculate the corresponding line equations $line_{L1\_t}$ and $line_{L2\_t}$, respectively, in the following:

$$line_{L1\_t} : a_{L1\_t}x + b_{L1\_t}y + c_{L1\_t} = 0; \tag{4.24}$$

$$line_{L2\_t} : a_{L2\_t}x + b_{L2\_t}y + c_{L2\_t} = 0. \tag{4.25}$$

Subsequently, first choose $line_{L1\_t}$ as the major line, and generate a line $line_{L1\_2\_t}$ which is perpendicular with $line_{L1\_t}$ and pass through the mean of the coordinates in the set S_{L2_t}, described as:

$$line_{L1\_2\_t} : a_{L1\_2\_t}x + b_{L1\_2\_t}y + c_{L1\_2\_t} = 0. \tag{4.26}$$

In the same way, choose *line*$_{L2\_t}$ as the major line, and generate a line *line*$_{L2\_1\_t}$ which is perpendicular with *line*$_{L2\_t}$ and pass through the mean of the coordinates in the set S_{L1_t}, described as:

$$line_{L2\_1\_t} : a_{L2\_1\_t}x + b_{L2\_1\_t}y + c_{L2\_1\_t} = 0. \qquad (4.27)$$

Then, calculate the errors $ERROR_{L1\_t}$ and $ERROR_{L2\_t}$ respectively by

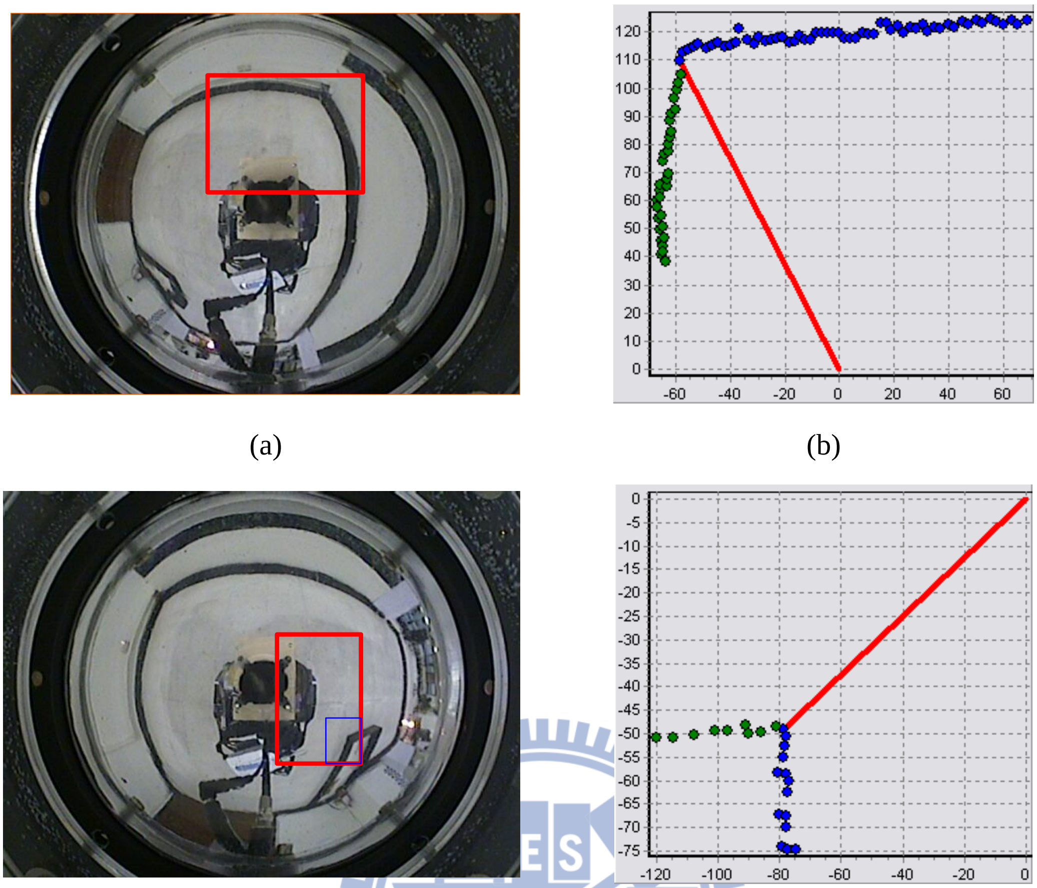
$$\begin{aligned} ERROR_{L1\_t} = & \sum_{n=1}^{t}\left(\frac{\left|a_{L1\_t} \times x_{L\theta n} + b_{L1\_t} \times y_{L\theta n} + c_{L1\_t}\right|}{\sqrt{a_{L1\_t}{}^{2} + b_{L1\_t}{}^{2}}}\right)^{2} \\ & + \sum_{m=t}^{n_L}\left(\frac{\left|a_{L1\_2\_t} \times x_{L\theta m} + b_{L1\_2\_t} \times y_{L\theta m} + c_{L1\_2\_t}\right|}{\sqrt{a_{L1\_2\_t}{}^{2} + b_{L1\_2\_t}{}^{2}}}\right)^{2}; \end{aligned} \qquad (4.28)$$

$$\begin{aligned} ERROR_{L2\_t} = & \sum_{n=1}^{t}\left(\frac{\left|a_{L2\_1\_t} \times x_{L\theta n} + b_{L2\_1\_t} \times y_{L\theta n} + c_{L2\_1\_t}\right|}{\sqrt{a_{L2\_1\_t}{}^{2} + b_{L2\_1\_t}{}^{2}}}\right)^{2} \\ & + \sum_{m=t}^{n_L}\left(\frac{\left|a_{L2\_t} \times x_{L\theta m} + b_{L2\_t} \times y_{L\theta m} + c_{L2\_t}\right|}{\sqrt{a_{L2\_t}{}^{2} + b_{L2\_t}{}^{2}}}\right)^{2}. \end{aligned} \qquad (4.29)$$

In this way, by the equation

$$i = \arg\min_{t\in[2,n_L-1]}\left(\min\left(ERROR_{L1\_t}, ERROR_{L2\_t}\right)\right), \qquad (4.30)$$

we can find the point $P_{L\theta i}$ which classifies the set S_L properly and then the set S_L' can be derived. Note that if *i* equal to 2 or $n_L - 1$, it means that all of points in the set S_L belong to the same wall. The results are shown in Figure 4.8; the algorithm is described in the next section. Note that the positive direction of the X_{Local}-axis in the CCS$_{Local}$ is opposite to the positive direction of the X_1-axis in the CCS$_1$.



(a) (b)

(c) (d)

Figure 4.8 Classification of mopboard data. (a) Original omni-image. (b) Result of classification of the Front Region of (a). (c) Another original omni-image. (d) Result of classification of the Left Region of (c).

4.5.3 Use of line fitting for vehicle direction adjustment

Before driving the vehicle to go forward or to turn around, it is desired that the vehicle can keep its path to parallel with the walls. Otherwise, the vehicle may hit the wall. For this purpose, the detected mopboard edges can be used, by a line fitting process, to generate a straight line which describes the direction of these detected edges. With the direction of the line, the vehicle can adjust its pose such that it can keep an appropriate distance with respect to the wall. The entire process is described in the following algorithm and illustrated in Figure 4.9.

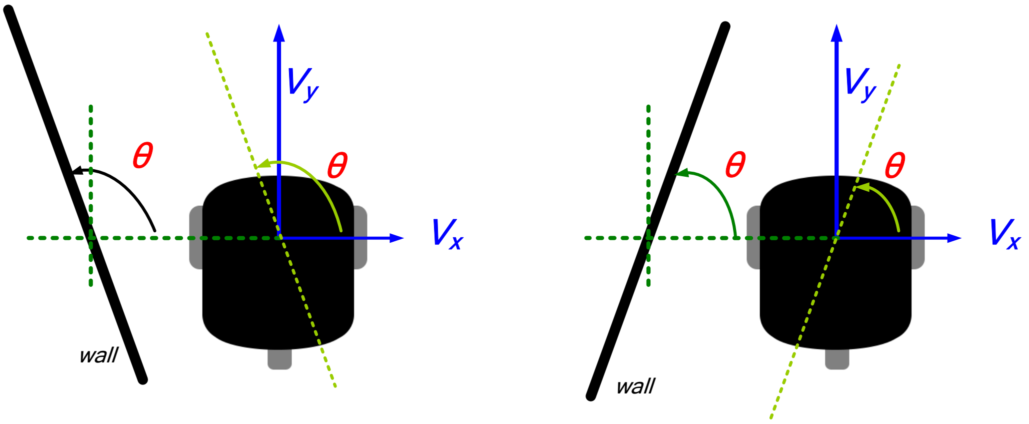


Figure 4.9 Illustration of the vehicle and the wall. (a) $\theta > 90^{\circ}$. (b) $\theta < 90^{\circ}$.

Algorithm 4.3: Adjustment of the vehicle direction to be parallel to the nearby wall.

Input: An omni-image I_0 of the nearby scene taken by the lower omni-camera. A scanning range R_0.

*Output:*none

Steps:

Step 1. Find mopboard edge points in I_0 using R_0 by the previously-mentioned mopboard edge point detection process (Section 4.3) to get a point set M_0.

Step 2. Fit the points in M_0 with a line L_0: $ax + by + c = 0$ by the LSE curve fitting scheme described in Section 3.3.2..

Step 3. Adjust the direction of the vehicle to the direction of L_0 by the following steps.

3.1 Find the direction of L_0, denote as $\theta_{L_0}$with respect to the positive direction of the V_x-axis of the VCS.

3.2 Compare θ_{L_0} with 90°:

(1) if $\theta_{L_0} > 90^{\circ}$, then turn to the left for an angle of $\theta_{L_0}-90^{\circ}$;

(2) else, if $\theta_{L_0} < 90^{\circ}$, then turn to the right for an angle of $90^{\circ} - \theta_{L_0}$;

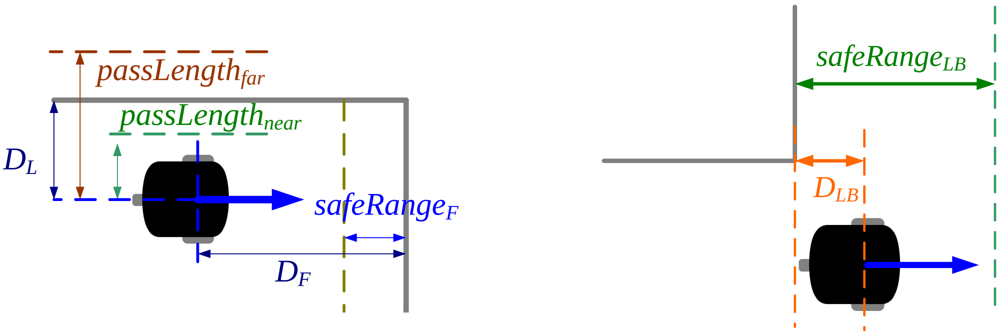
(3) otherwise, make no adjustment to the direction of the vehicle.

4.5.4 Proposed navigation strategy and process

Before the vehicle starts a navigation session, it will estimate the distances to the nearby walls which are within the relevant angular scanning ranges. Based on the distances, we can know if it can turn to the left, turn to the right, or go forward, and then issue an appropriate control instruction to the vehicle to drive it to navigate safely. The major rules for controlling the vehicle are described as follows.

1. For the work of the imaging system, keeping an appropriate distance with the wall is needed.
2. The vehicle can go forward until the left wall is not detected further or until it is blocked by the frontal wall.

Both situations are illustrated in Figure 4.10, and the algorithm of issuing a control instruction to the vehicle for each navigation cycle is described as follows. Note that the distance D_{LB} is assigned a value only under the condition that the vehicle exceeds the corner point, as shown in Figure 4.10(b). For gathering enough information, it is desired that vehicle go forward a fixed distance $moveDist_{fix}$ if it is moveable, and then collect information. Referring Figure 4.10 for notations used in the algorithm. D_L, D_F, and D_R are the distances between the vehicle and the left wall, the frontal wall, and the right wall, respectively, if the corresponding walls are detected. The threshold values $passLength_{near}$ and $passLength_{far}$ are used to decide whether the vehicle can move forward or not.



(a) (b)

Figure 4.10 Illustration of the safe ranges. (a) Approaching to the frontal wall. (b) Exceeding the corner.

Algorithm 4.4: Control decision for vehicle navigation.

Input: The distances D_L, D_F, D_R, D_{LB} and pre-selected length or distance values, $passLength_{near}$, $passLength_{far}$, $moveDist_{fix}$, $safeRange_F$, and $safeRange_{LB}$, used as threshold values.

*Output:*none

Steps:

Step 1. If the vehicle turned to the left in the last navigation cycle, go to Step 2.2; otherwise, go to Step 2.

Step 2. Perform the following steps to check whether or not the vehicle should go forward.

2.1 If $passLength_{near} \leqq D_L \leqq passLength_{far}$, continue; otherwise, go to Step 3.1

2.2 If $safeRange_F \leqq D_F$, do the following steps; otherwise, go to Step 3.2.

i. If $(D_F - safeRange_F) \geqq moveDist_{fix}$, go forward for the distance of $moveDist_{fix}$ and then exit;

ii. Otherwise, go forward for the distance of $D_F - safeRange_F$ and then exit.

Step 3. Perform the following steps to check whether or not the vehicle should turn around.

3.1 Check if $D_{LB} \geqq safeRange_{LB}$:

i. if yes, turn to the left for the angle of 90^{o}, and then exit;

ii. otherwise, go forward for the distance of $safeRange_{LB} - D_{LB}$, and then turn to the left for the angle of 90^{o}, and finally exit.

3.2 Turn to the right for the angle of 90^{o}, and then exit.

Chapter 5
Automatic Construction of House Layout by Autonomous Vehicle Navigation

5.1 Introduction

The goal of this study is to construct a 3-D house-layout of the room space. In Chapter 4, we described how the vehicle conducts navigation and gathers the environment information that includes the detected mopboard edge points and the omni-images taken by the imaging system. With the mopboard edge points, we propose in this study a global optimization method to generate the floor-layout to fit them, and the details will be described in Section 5.2. Besides, objects on walls such as doors and windows are also concerned. Based on the floor-layout, we conduct analysis of the omni-image to detect objects on walls. The entire process is described in Section 5.3.

5.2 Proposed Global Optimization Method for Floor-layout Construction

5.2.1 Idea of proposed method

In order to create the floor layout, we use a straight line to fit the detected edge points which belongs to the same wall by the LSE curve fitting scheme, and we do this for each wall, respectively. However, due to the error of the odometer of the vehicle or possible position estimation errors from the imaging system, the adjacent fitted lines may be not mutually perpendicular, as shown in Figure 5.1, though in fact the adjacent walls are so.

One way out, as adopted in this study, is to use straight lines which are perpendicular or parallel to each other to fit the edge points in the beginning. In this way, how to fine-tune all of the fitting lines such that the adjacent fitting lines not only are mutually perpendicular but also can fit the edge points with the minimum tolerance is an important task. A global LSE optimization method is proposed to solve this problem, and the details are described in the following section.

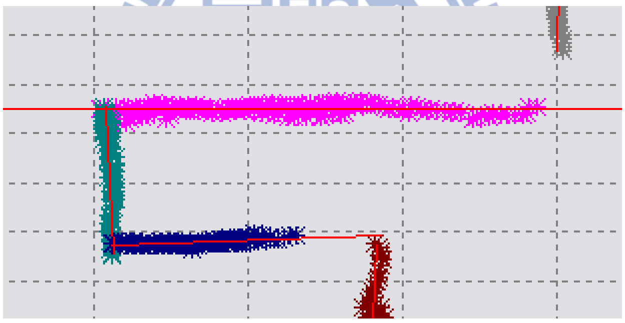


Figure 5.1 The edge points of different walls (in different colors) and the fitting lines.

5.2.2 Details of proposed method

As mentioned, it is reasonable to use two mutually perpendicular lines to fit the edge points which belong to adjacent mutually perpendicular walls. It means that if we choose one fitting line and adjust the direction of it, the directions of other fitting lines will change, too. As a result, using the fitting lines to fit the edge points will incur fitting errors. Based on this idea, we propose a global LSE optimization method to minimize the fitting error and the entire process is described in the following algorithm.

Algorithm 5.1: Floor-layout construction.

Input: n sets of detected mopboard edge points, S_1, S_2, …, S_n of n walls, W_1, W_2, …, W_n.

Output: A floor layout.

Steps:

Step 1. (*Line fitting for each wall*) Fit the points in S_k of W_k with a line L_k by the LSE curve fitting scheme and compute its mean point M_k where $1 \leq k \leq n$.

Step 2. (*Optimal fitting with respect to a chosen fitting line*) Perform the following steps to obtain *globally* optimal fitting with respect to a selected fitting line of a certain wall.

2.1 Choose a fitting line L_k, starting from k =1 until k = n, and compute its direction angle θ_k.

2.2 *Adjust* θ_k by adding a small angle ε (initially adding –10° and then adding 0.1° each time later until adding +10°), resulting in θ_k', to generate a new line L_k' such that L_k' passes the point M_k with direction angle θ_k'.

2.3 (*Generation of other fitting lines*) Generate a sequence of lines L_{k+1}', L_{k+2}', …, $L_{(k+n-2)\text{mod } n}'$ with each L_{k+i}' perpendicular to its former line $L_{k+(i-1)}'$ and passing its original mean point M_{k+i} (i.e., every two neighboring lines are mutually perpendicular).

2.4 (*Computing the sum of fitting errors of all lines*) Compute the error e_i of fitting all the points in S_i of W_i to line L_i' obtained in the last two steps (Steps 2.2 and 2.3), and sum the errors up to get a *total error* e_k' for L_k'.

2.5 Repeat Steps 2.2 through 2.4 until the range of angular adjustment, (–10°, + 10°), is exhausted.

2.6 Find the minimum of all the total errors e_k' and denote it as $e_{\min, k}$.

2.7 Repeat Steps 2.1 through 2.6 to compute the $e_{\min, k}$ for all $k = 1, 2, \ldots, n$.

2.8 Find the global minimum error $e_{\min, k_o}$ as the minimum of all the $e_{\min, k}$.

Step 3. Take all the lines with adjusted angles corresponding to $e_{\min, k_o}$ as the desired floor layout.

An example of the results yielded by the above algorithm is shown in Figure 5.2.

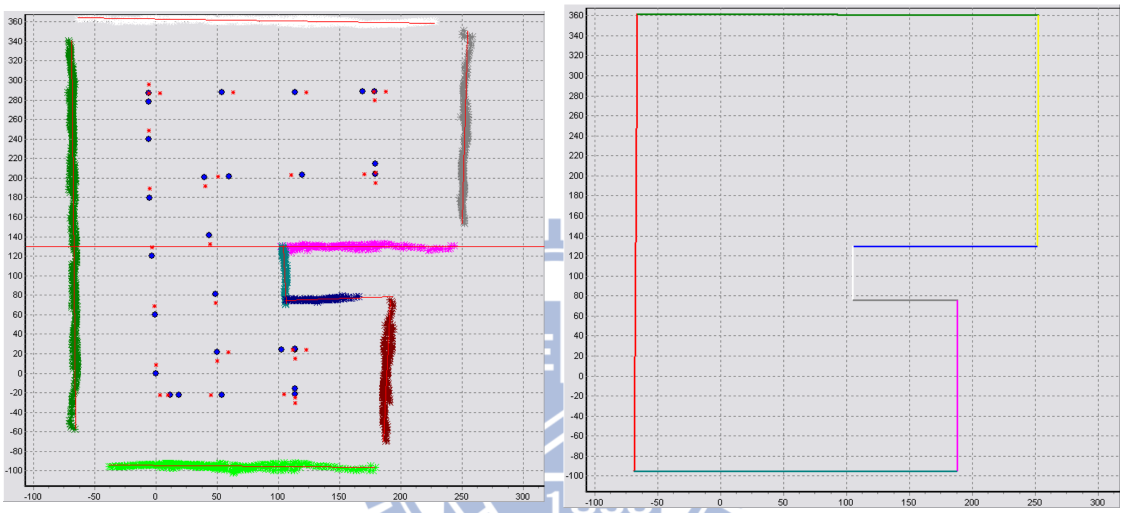


Figure 5.2 Illustration of floor-layout construction. (a) Original edge point data. (b) A floor-layout of (a).

5.3 Analysis of Environment Data in Different Omni-images

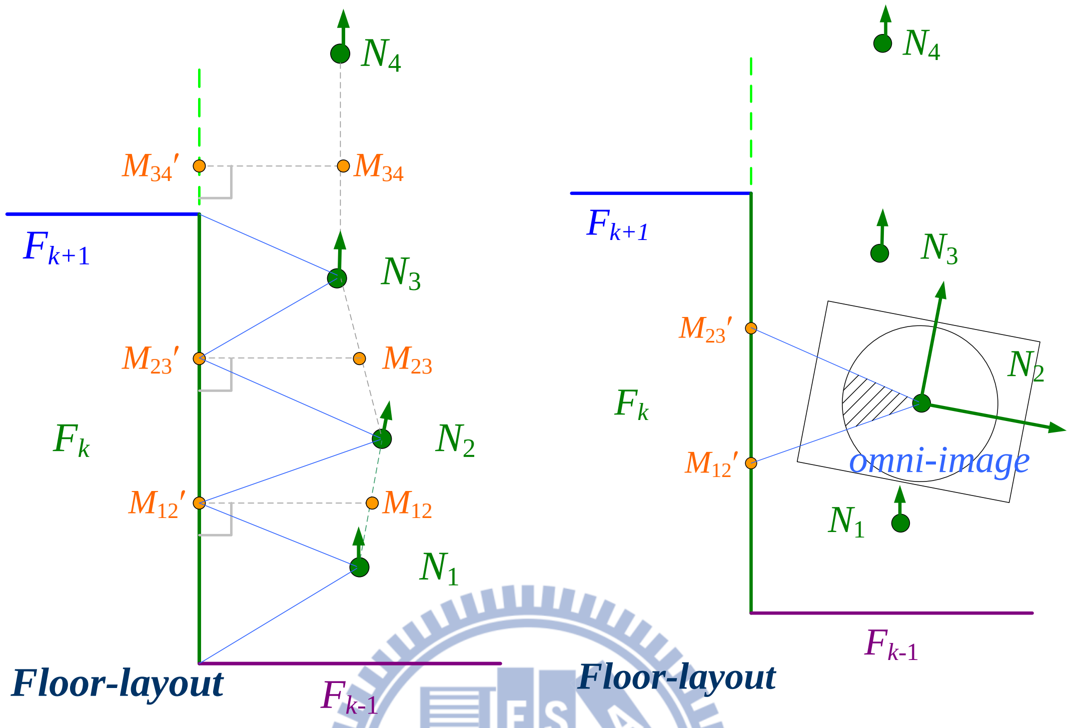
5.3.1 Introduction

Only creating a floor layout is insufficient for use as a 3-D model of the indoor room space. The objects on walls such as doors and windows must also be detected and be drawn to appear in the desired 3-D room model. Therefore, it is indispensible to analyze the omni-images taken by the imaging system to extract such objects.

However, an omni-image covers a large range, but the further the distance is, the larger the estimation error is. How to retrieve the desired information is an important task. For this, we propose a method to determine a scanning range with two direction angles for each pair of omni-images based on the floor layout edge equation. The details will be described in Section 5.3.2. With the scanning region for each omni-image, we can retrieve appropriate 3-D information from different omni-images. Note that each object which is detected by the scanning region of each omni-image is regarded as an individual one. Therefore, adjacent objects which appear in the consecutive omni-images taken from the same omni-camera (the upper or the lower one) must be combined into one based on, e.g., the information of their positions. Also, due to the configuration of the imaging system, some objects on the wall, such as windows and doors, may appear in the pair of omni-images (the upper and the lower ones). To solve it, we propose a method to recognize doors and windows from the combined objects, and the method is described in Section 5.3.3.

5.3.2 Proposed method for retrieving accurate data

As shown in Figure 5.3, for each wall W_k, there are multiple navigation imaging spots N_1, N_2, …, and N_m for taking omni-images as the environment information. In order to determine a scanning region for each omni-image taken at N_1 through N_m, we calculate the midpoint M_{ij} of N_i and N_j for all i = 1, 2, …, n – 1 and j = i + 1. Then, we project each midpoint M_{ij} onto the line L_k which is the result of the floor-layout construction process mentioned previously, resulting in a projection point M_{ij}'. And based on the positions of the projection points and the navigation imaging spots, we can determine the cover ranges. With the cover range and the direction of the vehicle at the navigation spot, we can determine the scanning range with two direction angles for each pair of omni-images.



(a) (b)

Figure 5.3 Illustration of determining scanning range. (a) Decide the cover range. (b) Relevant scanning range on omni-image.

However, if we conduct the detection of objects within each scanning range, the mopboard will be detected as an object on the wall. Because the objects to be detected are doors and windows, and the mopboard is regarded as a feature of a wall, the mopboard should not appear in the detected objects. For this reason, we define a scanning region excluding the mopboard in the scanning range. In order to exclude the mopboard part in the scanning range, we use the pano-mapping table in a reverse way to estimate the image coordinates of a concerned space point at a known position. More specifically, we use the distance to a detected edge point, and a pre-defined height of the mopboard to calculate the elevation of a point which is on the top edge of the mopboard with respect to the CCS$_1$. Then, we can look up the pano-mapping table to find its radius in the omni-image. As a result, the mopboard part can be excluded in the omni-image, forming a desired scanning region. In this way, the object detection process can be carried out within this region. The detailed algorithm is described as follows where $D_{k,\theta}$ is the distance to the detected mopboard edge point which is on the direction of angle θ with respect to the CCS$_1$, and k is the index of the imaging spot of each wall.

Algorithm 5.2: Determination of scanning region of a wall.

Input: The floor-layout line F_k of W_k, omni-images $I_{1,1}$, $I_{1,2}$, …, and $I_{1,m}$; $I_{2,1}$, $I_{2,2}$, …,and $I_{2,m}$; navigation imaging spots N_1, N_2, …, and N_m; the distance $D_{t,s}$ estimated from mopboard detection where $1 \leq t \leq m$ and $1 \leq s \leq 360$; a constant H which is the height of the mopboard; pre-defined scanning radii r_{max} and r_{min}.

Output: The scanning regions $R_{1,i}$ for $I_{1,i}$ and $R_{2,i}$ for $I_{1,i}$, for all $i = 1, 2, …, m$.

Steps:

Step 1. (*Midpoint calculation and projection*) Perform the following steps to calculate the midpoints.

1.1 Calculate the midpoint M_{ij} of N_i and N_j, where $j = i - 1$.

1.2 Project M_{ij} onto the floor layout edge F_k, and find the projection point M_{ij}'.

1.3 Repeat Step 1.1 through 1.2 for all i from 1 to $m - 1$.

Step 2. (*Determination of cover range*) Perform the following steps to determine the cover range.

2.1 Find consecutive projection points which are on F_k, assuming that the indexes of the consecutive points M_{ij}' are found to be p through q:

2.2 For all $i = p, p + 1, …, q - 1$, check i according to the following rules.

(A) if $i = p$, then form the cover range R_i by N_i, M_{ij}', and the endpoint of F_k

which is nearest to M_{ij}', where $j = i + 1$;

(B) else, if $i = q - 1$, then form the cover range R_i by N_q, M_{si}', and the endpoint of F_k which is nearest to N_q, where $s = i - 1$;

(C) otherwise, form the cover range R_i by N_j, M_{ij}', and M_{js}', where $j = i + 1$, and $s = j + 1$.

Step 3. (*Determination of scanning region in omni-image*) Perform the following steps to determine a region excluding the mopboard portion.

3.1 Choose $I_{1,k}$ and $I_{2,k}$, starting with $k = 1$ until $k = m$, and the range $R_{1,k}$ and $R_{2,k}$ with two direction angles, respectively.

3.2 Choose the distance $D_{k,s}$, staring with $s = 1$

3.3 Check the value of s: if s is in the interval of the two direction angles, then compute the radii $r_{1k,s}$ and $r_{2k,s}$ by using the pano-mapping table in a reverse way and H, respectively; otherwise, do nothing.

3.4 Increase s by 1 and check the value of s: if $s \leq 360$, then go to Step 3.2; otherwise, continue.

3.5 Generate the scanning regions $R_{1,k}'$ and $R_{2,k}'$ which are subjected to the radii $r_{1k,s}$ and $r_{2k,s}$, and the pre-defined values of r_{max} and r_{min}, respectively, for all s such that $1 \leq s \leq 360$.

3.6 Repeat Steps 3.1 through 3.5 until all the scanning regions are determined.

5.3.3 Proposed method for door and window detection

With the above-mentioned scanning region, we can detect the objects in each omni-image by the proposed two-way angular scanning scheme which extends the method in Step 5 of Algorithm 4.1. The two-way scheme is used instead of the original one described in Algorithm 4.1 because the two-way scheme can achieve a better estimation. In the two-way angular scanning scheme, we first perform Steps 1 through 4 of Algorithm 4.1 for the pair of images, then traverse along the line L_θ from the outer boundary to the inner one and from the inner to the outer boundary within the scanning regions, and find the first intersection black pixel which is followed by 10 or more consecutive black pixels, respectively. The first found black pixel in each direction will be regarded as an element of the boundary of the detected object, as shown in Figure 5.4.

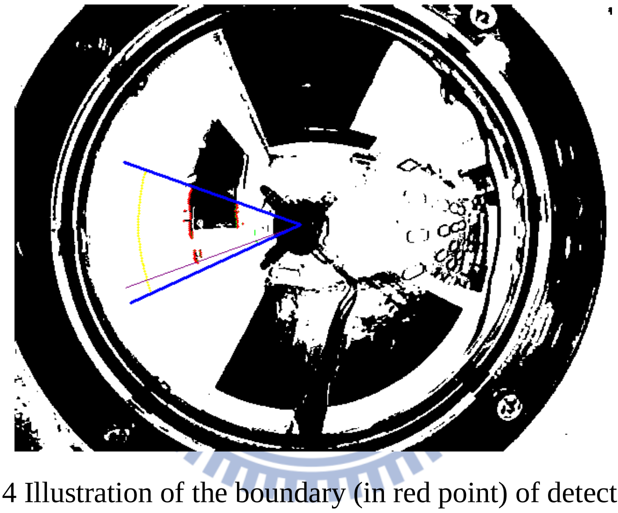


Figure 5.4 Illustration of the boundary (in red point) of detected object.

According to the two first scanned pixels which are detected by traversing along the scanning line with opposite directions, we utilize the information within the scanning line bounded by the two scanned pixels to determine whether the two pixels are boundary points of the object or not. Besides, objects may occupy the omni-image and are detected for some certain continuous angular interval. We utilize the property to combine the detected objects which are detected from certain continuous scanning lines into an individual one, as shown in Figure 5.5. In this way, there may be some individual objects in the scanning region. As a result, according to these pixels which are boundary points of the object, and by utilizing the 3-D position estimation method described in Section 4.3.2 by looking up the pano-mapping table, the average heights at the bottom and the top of the object can be estimated.

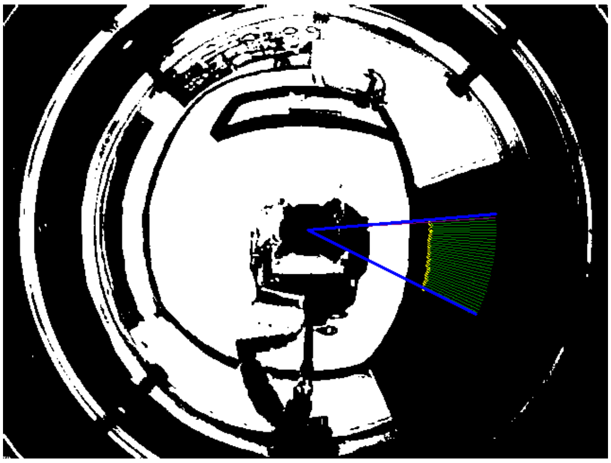


Figure 5.5 Illustration of the detected objects within the scanning region.

There are some major rules for combing the detected objects.

1. First, we traverse along the scanning line L_θ within the scanning region R' from opposite directions and find the pixels p_i and p_o of the object boundaries, respectively.
2. By counting the numbers n_b and n_{sum} of black pixels and all pixels between the two detected pixels along L_θ, if found, and by a threshold and the ratio $\frac{n_b}{n_{sum}}$, we can determine whether the pixels p_i and p_o are on the boundary of the object or not.
3. In R', if all the pixels p_i and p_o for each scanning line L_θ in an angular interval are all on the boundary of an object, then such a largest interval can used to describe the same object. Besides, in R', those combined objects may be combined again according to their positions.

And with the above rules, all individual objects can be determined in the same scanning region of an omni-image.

However, an object on a wall may appear to cross two or more scanning regions of corresponding source omni-images taken by the same upper or lower omni-camera. Besides, an object also has a possibility to appear in the pairs of omni-images simultaneously. For the above reason, we have to combine those detected individual objects which belong to the same one. Here, we denote O_1 and O_2 as the sets of the individual objects detected from the omni-images taken by the lower and upper omni-cameras, respectively. As shown in Figure 5.6, at first, we conduct the combination task for all the objects in O_1 and O_2, according to their positions to form two new sets O_1' and O_2', respectively. Then, we conduct the reorganization task. The process is described in the following algorithm.

Algorithm 5.3: Objects reorganization

Input: Sets O_1' and O_2' including the combined objects on walls.

Output: The window objects set O_W, and the door objects set O_D.

Steps:

Step 1. (*Objects recognition for each wall*) For each floor-layout edge F_k, perform the following steps to recognize the objects on wall W_k

1.1 Choose an object $o_{2,i}$ from O_2', to find an object in O_1' at a similar location.

(1) If such object $o_{1,j}$ is found, then check if it is connected to the mopboard by its location:

i. if yes, then recognize $o_{1,j}$ together with $o_{2,i}$ as a door, and add it to O_D;

ii. otherwise, recognize $o_{1,j}$ together with $o_{2,i}$ as a window, and add it to O_W.

(2) If such an object is not found, recognize $o_{2,i}$ as a window, and add it to O_W.

1.2 Repeat Step 1.1 until the objects of O_2' are exhausted.

1.3 Recognize the remaining objects in O_1' as widows, and add it to O_W.

Step 2. Repeat Step 1 until all the floor-layout edges are exhausted.

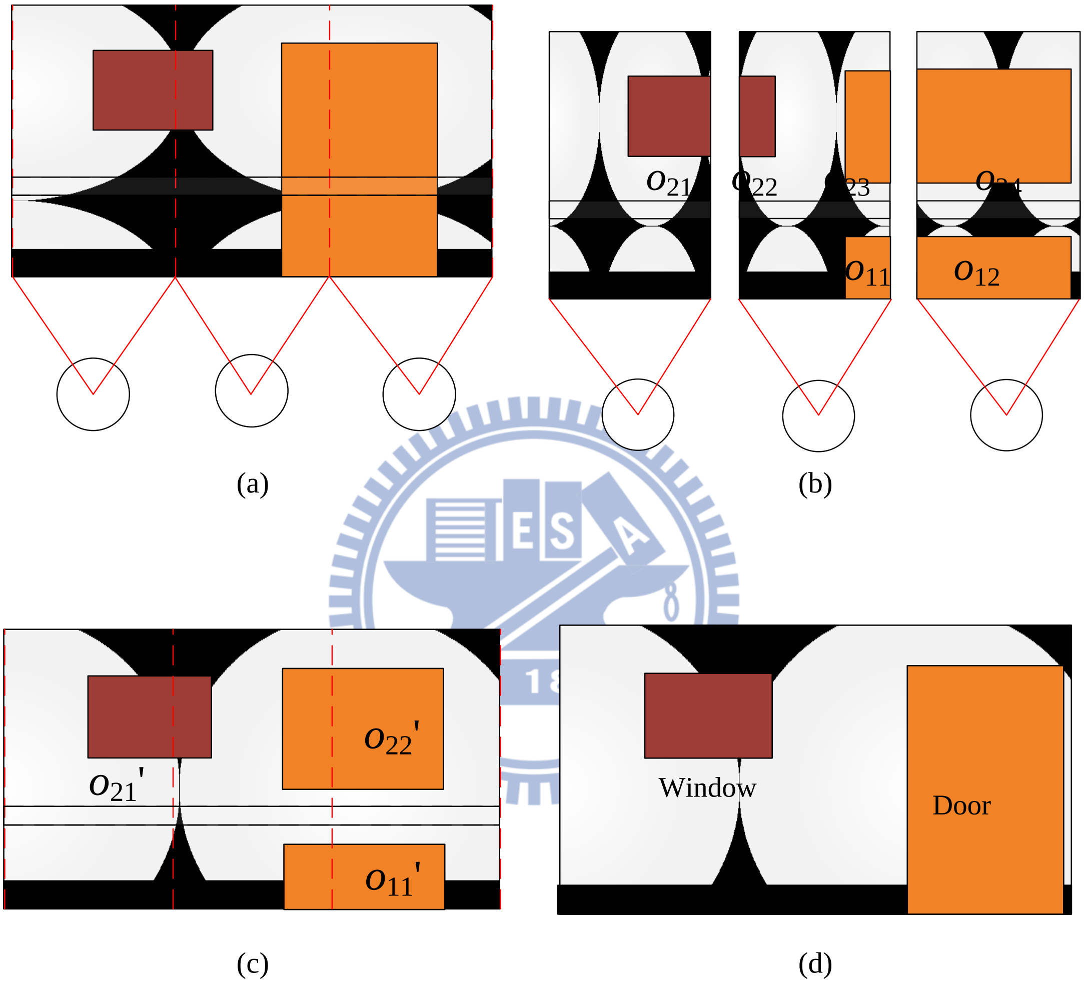
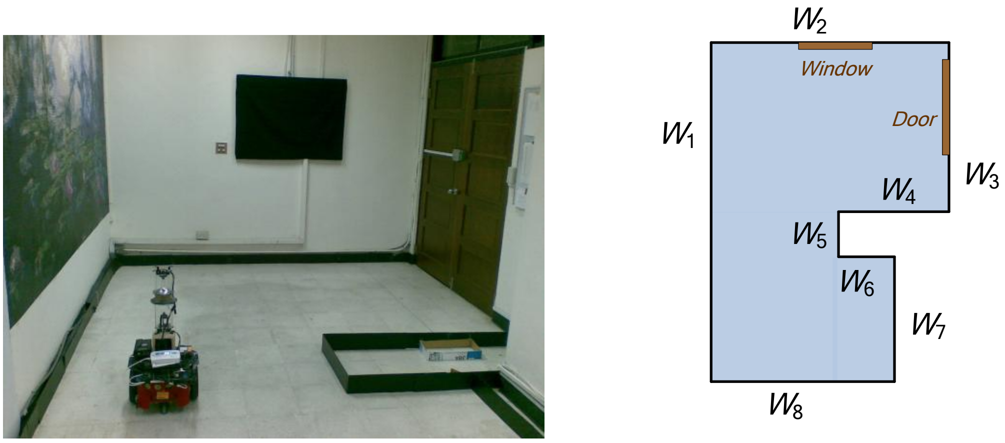


Figure 5.6 Illustration of object combinations. (a) Scanning regions. (b) Individual objects. (c) Combined objects for each omni-camera. (d) Reorganization of objects.

Chapter 6
Experimental Results and Discussions

6.1 Experimental Results

In this chapter, we will show some experimental results of the proposed automatic house-layout construction system by autonomous vehicle navigation based on mopboard following in indoor environments. The experimental environment was an empty indoor room with mopboards at the bottom of the walls, in the Computer Vision Laboratory of the Department of Computer Science in Engineering 3 Building at National Chiao Tung University, Taiwan, and is shown in Figure 6.1(a). For convenience, we denote walls as the notations shown in Figure 6.1(b).



(a) (b)

Figure 6.1 The experimental environment. (a) Side view. (b) Illustration of the environment.

At first, the vehicle was placed appropriately near wall W_1 with its direction not precisely parallel to it. When the vehicle started a navigation cycle, the vehicle adjusted its direction to keep its navigation path parallel to each wall by utilizing the imaging system to conduct the mopboard detection process as described previously. Figure 6.2 shows an example of the resulting images of mopboard detection in which the red dots show the detected mopboard edge points. After this action, the imaging system will gather the environment data and conduct the mopboard detection process again to estimate the locations of the mopboard edge points. And according to the estimated distances between the vehicle and the walls, the vehicle decided to go forward or turn around. Figure 6.3(a) shows the result of the estimated mopboard edge points which cover two adjacent walls. Figure 6.3(b) shows the result of the pattern classification procedure.

After the vehicle finished the navigation session, the floor layout was constructed with the globally optimal fitting method described previously. Figure 6.4(a) shows the estimated mopboard edge points of all walls and Figure 6.4(b) shows the floor layout which is constructed by the globally optimal fitting method.

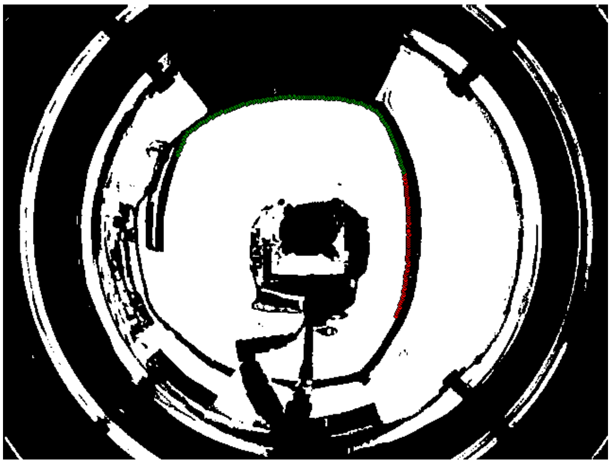
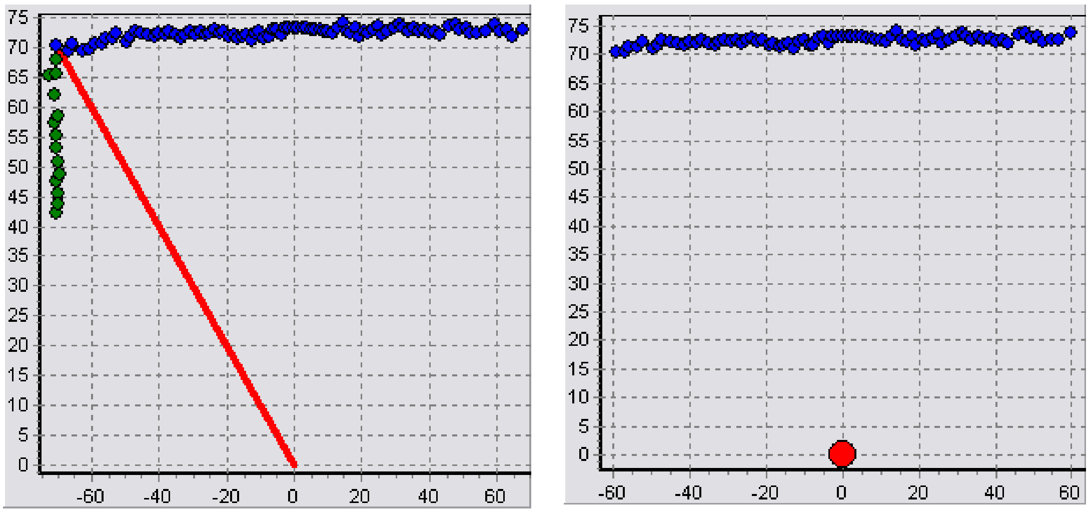


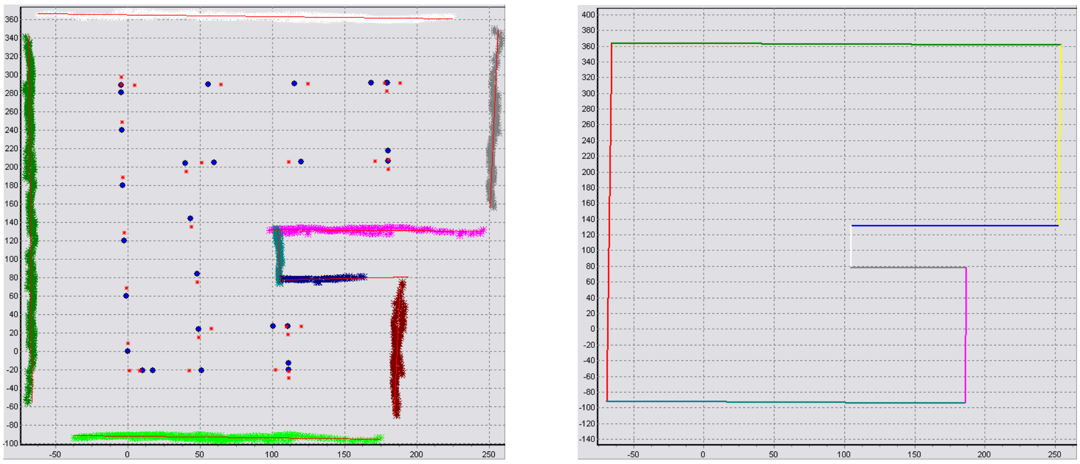
Figure 6.2 Detected mopboard edge points.



(a) (b)

Figure 6.3 Classification of mopboard edge points (a) The detected mopboard points. (b) Result of the classification (the points belonging to the upper wall).

In Table 6.1, we show the errors in percentage between the actual widths of the walls and the estimated data of 9 times of navigations using the proposed system. From the table, we see that the average error of the wall widths is 2.745%. Also, we compute the error in percentage of the estimated total perimeter of the floor layout with respect to the real data, and it is 0.23%. Both error percentages show that the precision of the proposed system is satisfactory for real applications.



(a) (b)

Figure 6.4 Illustration of global optimization. (a) The estimated mopboard edge points of all walls. (b) A floor layout fitting the points in (a).

Table 6.1 Precision of estimated wall widths and their error percentages.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Exp. No.	Wall No.	Real wall width	Estimated wall width	Estimated wall width error % [(1) – (2)]/(2)	Average estimated wall width error %	Total measured perimeter (Real perimeter = 1720.7)	Estimated perimeter error %
1	1	323	318.7	1.336	2.345	1714.5	0.36
	2	229.4	231.5	0.922			
	3	150	149.5	0.365			
	4	61.2	54.14	11.54			
	5	79.6	81.15	1.951			
	6	169	171.8	1.644			
	7	249	250.4	0.556			
	8	459.5	457.4	0.449			
2	1	323	322.4	0.194	2.586	1720.1	0.03
	2	229.4	231.1	0.744			
	3	150	150.1	0.091			
	4	61.2	53.02	13.36			
	5	79.6	81.09	1.875			
	6	169	172.5	2.05			
	7	249	253.3	1.739			
	8	459.5	456.6	0.632			
3	1	323	320	0.924	2.948	1720.3	0.02
	2	229.4	231.2	0.787			
	3	150	146.2	2.533			
	4	61.2	54.54	10.88			
	5	79.6	82.32	3.418			
	6	169	172.1	1.817			
	7	249	256.1	2.866			
	8	459.5	457.8	0.365			
4	1	323	321	0.624	3.129	1726.7	0.35
	2	229.4	232	1.146			
	3	150	144.4	3.741			
	4	61.2	55.42	9.445			
	5	79.6	82.81	4.036			
	6	169	172.1	1.847			
	7	249	259.4	4.18			
	8	459.5	459.6	0.015			
5	1	323	318.5	1.393	2.09	1714.3	0.37
	2	229.4	231.2	0.795			
	3	150	148.9	0.74			
	4	61.2	55	10.12			
	5	79.6	80.56	1.209			
	6	169	171.8	1.677			
	7	249	250.2	0.471			
	8	459.5	458.1	0.313			
6	1	323	319.5	1.075	3.137	1716	0.28
	2	229.4	230.7	0.562			
	3	150	147.3	1.802			
	4	61.2	53.18	13.11			
	5	79.6	82.5	3.647			
	6	169	172.1	1.825			
	7	249	254.7	2.303			
	8	459.5	456	0.772			
7	1	323	318.5	1.382	2.387	1719.2	0.09
	2	229.4	231.7	1.012			
	3	150	144.1	3.902			
	4	61.2	56.87	7.072			
	5	79.6	80.81	1.518			
	6	169	171.6	1.561			
	7	249	255.2	2.489			
	8	459.5	460.2	0.159			
8	1	323	319.3	1.155	2.749	1716.7	0.23
	2	229.4	231.7	1.014			
	3	150	146.2	2.524			
	4	61.2	54.56	10.85			
	5	79.6	81.54	2.442			
	6	169	171.2	1.331			
	7	249	254.6	2.248			
	8	459.5	457.5	0.428			
9	1	323	319.2	1.172	3.335	1718.4	0.13
	2	229.4	231.1	0.719			
	3	150	145.9	2.702			
	4	61.2	53.95	11.85			
	5	79.6	83.57	4.987			
	6	169	171.4	1.428			
	7	249	256.8	3.148			
	8	459.5	456.4	0.673			
average					2.745		0.21

After the floor layout was constructed, the system proceeded to extract doors and windows, if any. In our experimental environment, we do have a door and a simulated window. The door is high enough to be seen in both the images taken by the lower and the upper omni-cameras. An example of the images of the door and the door detection result is shown in Figure 6.5. In addition, we simulated a window in our experimental environment by creating a black frame and attaching it on a wall. The window is not so low that only upper omni-camera can "see" it. An example of the image of the window and the window detection result is shown in Figure 6.6.

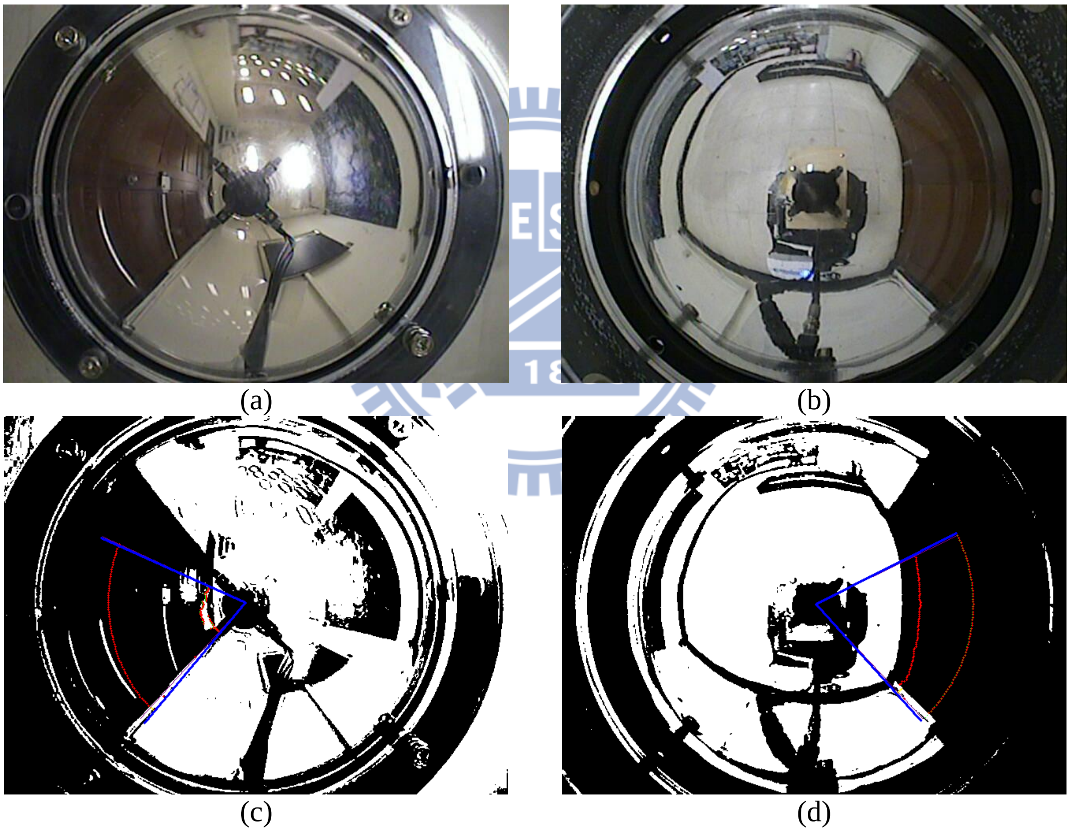
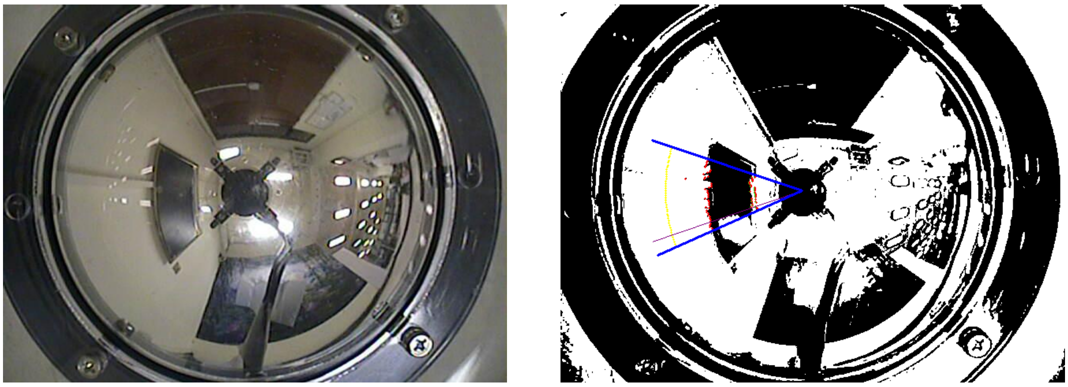

(a)
(b)
(c)
(d)

Figure 6.5 Images and door detection result. (a) Image of the door taken by the upper omni-camera. (b) Image of the door taken by the lower omni-camera. (c) Door detection result of (a). (d) Door detection result of (b).

After door and window detections, they were merged into the floor-layout data to form the final house layout. An example of house-layout construction in graphic form is shown in Figure 6.7, in which the house layout is displayed from two views, one from the top of the environment and the other from the back of the wall on which the window appears.



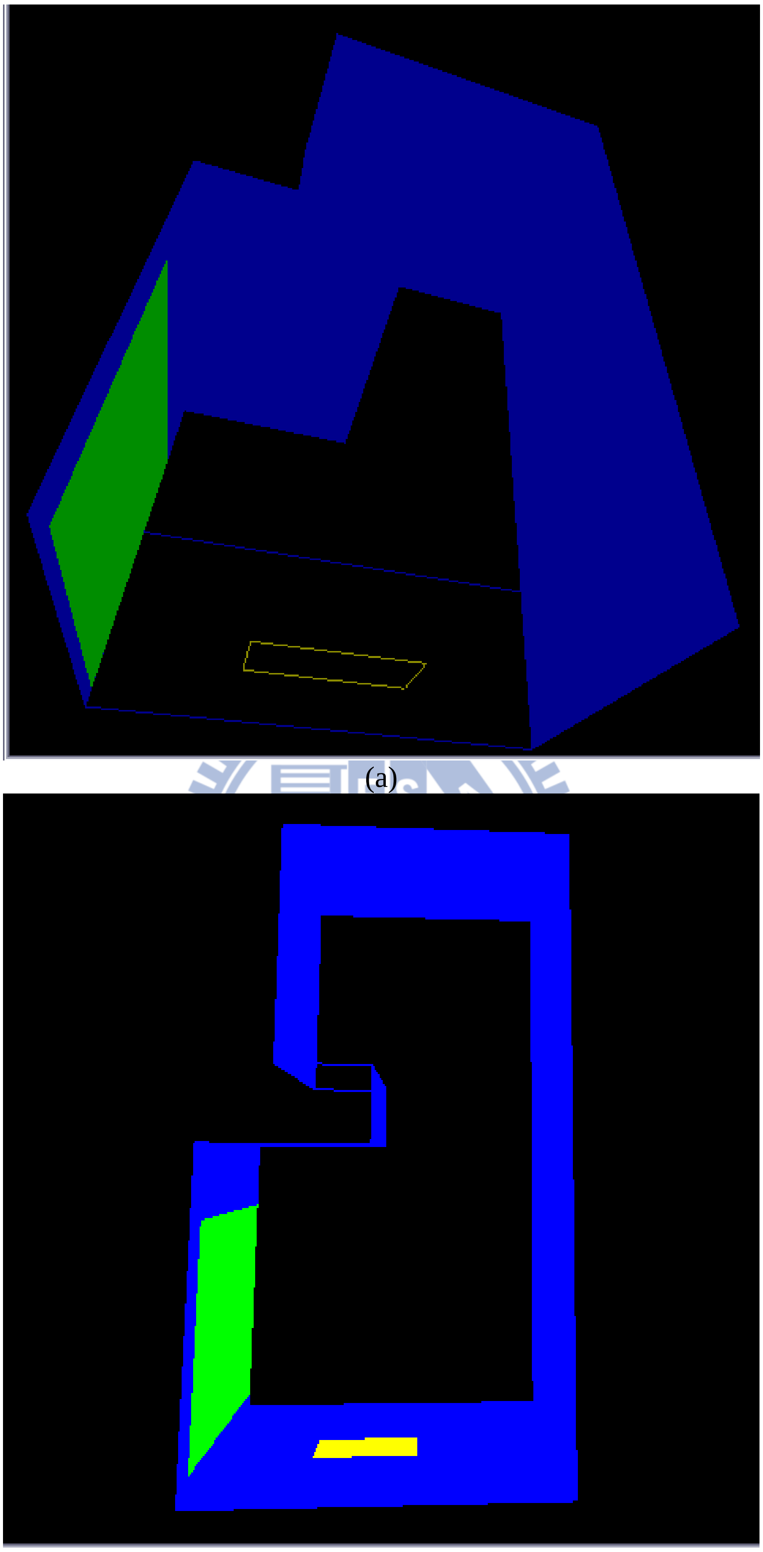
(a) (b)

Figure 6.6 Images and window detection result. (a) Image of the window taken by the upper omni-camera. (b) Window detection result of (a).

6.2 Discussions

From our experiments and the results, we see that the goal of the study — automatic house-layout construction by autonomous vehicle navigation without path learning — has been achieved. An inconvenience found in this study is that the two-camera omni-directional imaging system designed for this research is not high enough so that windows high above will not appear clearly (with its image part smeared by the plastic top cover of the upper omni-camera) when the vehicle navigates too close to the wall root. A possible solution is to construct a more transparent and spherical-shaped cover made possibly of glass.

Due to unavailability of an empty and sufficiently-large house space for conducting the experiment, the environment we used for our experiments was not totally closed and half of it was enclosed by simulated walls. In the future, more experiments should be conducted in more realistic room spaces.



(a)

(b)

Figure 6.7 Graphic display of constructed house layout. (a) Viewing from the top (green rectangle is a door and yellow one is a window). (b) Viewing from the back of the window.

Chapter 7
Conclusions and Suggestions for Future Works

7.1 Conclusions

A system for automatic house-layout construction by vision-based autonomous vehicle navigation in an empty indoor room space has been proposed. To achieve acquisition of environment images, a new type of omni-directional camera has been designed for this study, which consists of two omni-cameras aligned coaxially and back to back, with the upper camera taking images of the upper semi-spherical space of the environment and the lower camera taking images of the lower semi-spherical space. A so-called pano-mapping table [7] is used for computing the depth data of space feature points.

The proposed automatic house layout construction process consists of three major stages: (1) vehicle navigation by mopboard following; (2) floor layout construction; and (3) 3-D house layout construction. In the first stage, a vehicle is navigated to follow the mopboards at the roots of the walls in the house. A pattern classification technique has been proposed for classifying the mopboard points detected by an image processing scheme applied directly on the omni-image. Each group of mopboard points so classified are then fitted by a line using an LSE criterion. The line is used to represent the related wall.

In the second stage, a global optimization method has been proposed to construct a floor layout from all the wall lines in a sense of minimizing the total line fitting error.

In the last stage, doors and windows are detected from the omni-images taken in the navigation session. An algorithm has been proposed to match rectangular areas appearing in the lower and upper omni-images taken by the respective cameras, to decide the existence of doors or windows. And then the detected door and window data are merged into the wall line data to get a complete 3-D data set for the house. Finally, the data set is transformed into a graphic form for 3-D display of the house from any viewpoint.

The entire house layout construction process is fully automatic, requiring no human involvement, and so is very convenient for real applications. The experimental results show the feasibility of the proposed method.

7.2 Suggestions for Future Works

There exist several interesting topics for future research, which are listed in the following.

1. The proposed two-camera omni-directional imaging system may also be used for other applications like environment image collection and 3-D environment model construction.
2. More in-house objects, like paintings, furniture, poles, and so on, may be extracted from omni-images of the house environment for more complete construction of the house layout.
3. More applications of the proposed methods, like house dimension measuring, unknown environment exploration, automatic house cleaning, etc., may be investigated.
4. More techniques for acquiring the corner and line information from house ceilings using the proposed omni-camera system may be developed.

References

[1] Z. Zhu, “Omnidirectional stereo vision,” *Proceedings of Workshop on Omnidirectional Vision in the 10th IEEE ICAR*, pp. 1-12, Budapest, Hungary, Aug 2001.

[2] A. Ohya, A. Kosaka, and A. Kak, “Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing,” *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, pp. 969-978, 1998.

[3] J. Gluckman, S. K. Nayar, and K. J. Thoresz, “Real-time omnidirectional and panoramic stereo,” *Proceedings of DARPA98*, pp.299–303, 1998.

[4] H. Ukida, N. Yamato, Y Tanimoto, T Sano, and H. Yamamoto, “Omni-directional 3D measurement by hyperbolic mirror cameras and pattern projection,” *Proceedings of 2008 IEEE Conference on Instrumentation & Measurement Technology*, Victoria, BC, Canada, pp.365-370, May. 12-15, 2008.

[5] B. S. Kim, Y. M. Park and K. W. Lee, “An experiment of 3D reconstruction using laser range finder and CCD camera,” *Proceedings of IEEE 2005 International Geoscience and Remote Sensing Symposium*, Seoul, Korea, pp.1442-1445, July 25-29, 2005.

[6] S. Kim and S. Y. Oh, “SLAM in indoor environments using omni-directional vertical and horizontal line features,” *Journal of Intelligent and Robotic Systems*, v.51 n.1, p.31-43, January 2008.

[7] S. W. Jeng and W. H. Tsai, “Using pano-mapping tables for unwrapping of omni-images into panoramic and perspective-view images,” *Journal of IET Image Processing*, Vol. 1, No. 2, pp. 149-155, June 2007.

[8] K. L. Chiang. “Security patrolling and danger condition monitoring in indoor environments by vision-based autonomous vehicle navigation,” *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2005.

[9] J. Y. Wang and W. H. Tsai, “A study on indoor security surveillance by vision-based autonomous vehicles with omni-cameras on house ceilings,” *M. S. Thesis,* Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June. 2009.

[10] M. C. Chen and W. H. Tsai, "Vision-based security patrolling in indoor environments using autonomous vehicles, ”*Proceedings of 2005 Conference on Computer Vision, Graphics and Image Processing,* pp. 811-818, Taipei, Taiwan, Republic of China, August 2005.

[11] C. J. Wu and W. H. Tsai, “Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings,” *Robotics and Autonomous Systems*, Vol. 57, No. 5, pp. 546-555, May 2009.

[12] P. Biber, S. Fleck, and T. Duckett, “3D modeling of indoor environments for a robotic security guard,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 3, pp. 124-130, 2005.

[13] S. B. Kang and R. Szeliski, “3-d scene data recovery using omnidirectional multi-baseline stereo,” Int. J. Comput. Vision, p. 167~183, Oct 1997.

[14] J. I. Meguro, J, I, Takiguchi, and Y, A, T, Hashizume, “3D reconstruction using multibaseline omnidirectional motion stereo based on GPS/dead-reckoning compound navigation system,” *International Journal of Robotics Research*, Vol. 26, No. 6, pp. 625-636, June 2007.