

國立交通大學

多媒體工程研究所

碩士論文

利用超音波感測及電腦視覺技術經由簡易
學習作自動車導航之研究

A Study on Mobile Robot Navigation with Simple
Learning Procedures by Ultrasonic Sensing and Computer
Vision Techniques

研究生：蔡雙圓

指導教授：蔡文祥 教授

中華民國九十七年六月

利用超音波感測及電腦視覺技術經由簡易學習作自動車導航之研究
A Study on Mobile Robot Navigation with Simple Learning Procedures
by Ultrasonic Sensing and Computer Vision Techniques

研究生：蔡雙圓

Student : Shung-Yung Tsai

指導教授：蔡文祥

Advisor : Wen-Hsiang Tsai

國立交通大學
多媒體工程研究所
碩士論文



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 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

利用超音波感測及電腦視覺技術經由簡易學習作自動車導航之研究

研究生：蔡雙圓

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

本研究提出了兩套智慧型自動車系統，其一是以視覺為基礎的自動巡邏車，具有利用跟隨人物同時做學習路徑、與人互動、二維影像分析以及路徑規劃等功能；其二為利用超音波訊號分析做智慧型導航的自動車。我們利用一台具有 PTZ 攝影機的自動車作為實驗平台，並且利用有線網路與自動車通訊，使其航行於室內環境中。針對一利用格狀圖案為校正目標的舊有影像校正方法，我們提出了一精確的十字中心自動偵測方法，增進了校正的準確性與自動化。我們也建立了一計步器校正技術，用一校正模型來減少自動車所累積的行走機械誤差。接著我們發展一利用動態調整跟隨人物衣服顏色參考值的方法，來適應跟隨人物時環境中的不均勻照度。針對跟隨人物在自動車前的快速消失，我們亦提出一利用攝影機轉動角度，紀錄使用者方向的尋找方法，並使用一增加參考圖素數量的方法，來增進偵測人物朝向的準確率。當使用者要求自動車進行學習時，自動車能夠從影像中自動找出需要被監控的物品。此外，我們利用最小化均方差以及動態選擇門檻值，來規劃跟隨人物時所學習的路徑。我們同時提出一分析串列超音波訊號的方法，使自動車能在狹窄的室內空間中進行導航。最後我們以成功的巡邏、跟隨和導覽實驗證明本系統的完整性與可行性。

A Study on Mobile Robot Navigation with Simple Learning Procedures by Ultrasonic Sensing and Computer Vision Techniques

Student: Shung-Yung Tsai Advisor: Prof. Wen-Hsiang Tsai, Ph. D.

Institute of Multimedia Engineering, College of Computer Science

National Chiao Tung University

ABSTRACT

Two intelligent autonomous vehicle systems for use in indoor environment applications are proposed. One is a vision-based vehicle system for security patrolling which has the capabilities of learning paths by person following, interaction with humans, analyzing 2D images and conducting path planning. Another is an intelligent vehicle system for navigation by analyzing ultrasonic signals. A vehicle equipped with wired control and a web PTZ camera is used as a test bed. First, a method for improving the practicability of an adopted camera calibration method is proposed, which uses a grid pattern as the calibration target and detects the centers of cross shapes on the pattern in a more precise and automatic manner. Next, a technique of odometer calibration is proposed, which uses an odometer calibration model to reduce the incremental mechanical errors the vehicle suffers. Besides, a technique for person following in an environment with variant illuminations is proposed, which adjusts the reference color value for person following dynamically. To follow a target person who turns fast in front of the vehicle, a technique of using the image information exhibited by the angle of panning of the camera to search the disappearing person is proposed. Also proposed is a technique for enhancing the accuracy of detection of the facing direction of a human by increasing the number of pixels for use in the detection work.

For object monitoring, the vehicle is enabled to find out the 2D object in the image automatically when the followed person asks it to learn the object. In addition, a minimum MSE technique for refining dynamically the path learned while following a person is proposed. Finally, a method of vehicle navigation in corridor environments by the use of the ultrasonic signal sequence is proposed. Good experimental results show the flexibility and feasibility of the proposed methods for the applications of security patrolling, person following, and people guiding in indoor environments.

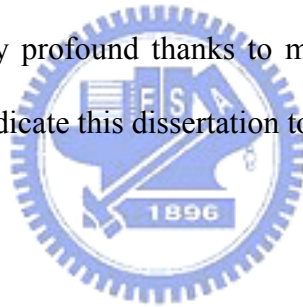


ACKNOWLEDGEMENTS

I am 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 my personal growth.

Thanks are due to Mr. Chih-Jen Wu, Mr. Kuan-Chieh Chen, Mr. Jiun-Tsung Wang, Mr. Guan-Lin Huang, Miss Hsing-Chia Chen, and Mr. Li-Jyun Lyu 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 my thesis study.

Finally, I also extend my profound thanks to my family for their lasting love, care, and encouragement. I dedicate this dissertation to my beloved parents.



CONTENTS

ABSTRACT (in Chinese)	i
ABSTRACT (in English)	ii
ACKNOWLEDGEMENTS	iv
CONTENTS	v
LIST OF FIGURES	vii
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Survey of Related Studies	3
1.3 Overview of Proposed Approach.....	4
1.4 Contributions.....	6
1.5 Thesis Organization	8
Chapter 2 System Configuration and Guidance Principles	10
2.1 Introduction.....	10
2.2 System Configuration	11
2.2.1 Hardware configuration.....	12
2.2.2 Software configuration.....	13
2.3 Learning Strategy and Major Steps in Proposed Process	14
2.4 Path Planning and Patrolling Principle and Major Steps in Proposed Process	15
2.5 Vehicle Navigation Principle and Major Steps in Proposed Process	17
Chapter 3 Camera and Odometer Calibration	21
3.1 Introduction.....	21
3.2 Camera Calibration by Cross Shape Detection.....	22
3.2.1 Idea of Proposed Camera Calibration Method.....	22
3.2.2 Coordinate Systems and Directional Angles of Camera.....	23
3.2.3 Review of Adopted Camera Calibration Method.....	25
3.2.4 Proposed Cross Shape Detection Technique.....	29
3.3 Odometer Calibration by Quadratic Functions	34
3.3.1 Idea of Odometer Calibration	34
3.3.2 Odometer Calibration Model	34
3.3.3 Curve Fitting	37
3.3.4 Navigation by Odometer Calibration.....	41

Chapter 4	Learning Procedures	43
4.1	Introduction.....	43
4.2	Human Following	43
4.2.1	Review of Adopted Method.....	43
4.2.2	Proposed Improvement of Adopted Process	45
4.2.3	Person Tracking Strategy	48
4.3	Learning of Path Data	51
4.4	Detection of Human Facing Direction.....	52
4.4.1	Review of Adopted Process	52
4.4.2	Proposed Improvement of Adopted Process	54
4.5	Learning of 2D Objects.....	57
4.5.1	Review of adopted algorithm for learning of objects	57
4.5.2	Learning 2D Objects Automatically	60
4.5.3	Finding Regions of 2D Objects.....	61
4.5.4	Finding the Horizontal Line Automatically.....	64
Chapter 5	Path Planning by Minimizing Mean Square Errors Using Ultrasonic Signals	69
5.1	Introduction.....	69
5.2	Path Planning by MSE criterion	70
5.3	Dynamic Path Decomposition	73
5.4	Vehicle Navigation by 2D Object Image Matching.....	75
Chapter 6	ALV Navigation by Ultrasonic Signal Sequences	80
6.1	Ultrasonic sensing in ALV System	80
6.2	Principle of Navigation.....	81
6.2.1	Learning Strategy.....	81
6.2.2	Navigation Strategy	82
6.3	Applications in Tour Navigation.....	86
6.4	Experimental Results	86
Chapter 7	Experimental Results and Discussions	91
7.1	Experimental Results	91
7.2	Discussions	98
Chapter 8	Conclusions and Suggestions for Future Works	100
8.1	Conclusions.....	100
8.2	Suggestions for Future Works.....	102
References		104

LIST OF FIGURES

Figure 1.1 Flowcharts of proposed systems.....	8
Figure 2.1 Equipment connection situations in this study.	12
Figure 2.2 The vehicle Pioneer 3 used in this study. (a) A front view of the vehicle. (b) A side view of the vehicle.....	13
Figure 2.3 The pan-tilt-zoom camera used in this study. (a) A perspective view of the camera. (b) A front view of the camera. (c) A left-side view of the camera.	14
Figure 2.4 An illustration of the automatic learning process by person following.....	16
Figure 2.5 An illustration of the path planning process.	18
Figure 2.6 An illustration of the patrolling process.	19
Figure 2.7 An illustration of the learning process of the vehicle navigation system. ..	19
Figure 2.8 An illustration of the navigation process.....	20
Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The global coordinate system (c) The vehicle coordinate system. (d) The spherical coordinate system.....	24
Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The global coordinate system (c) The vehicle coordinate system. (d) The spherical coordinate system. (continued)	25
Figure 3.2 The pan angle of the camera. (a) $0 \leq \theta_c \leq \pi$. (b) $0 \geq \theta_c \geq -\pi$	26
Figure 3.3 The tilt angle of the camera. (a) $0 \leq \varphi_c \leq \frac{\pi}{2}$. (b) $0 \geq \varphi_c \geq -\frac{\pi}{2}$	27
Figure 3.4 An illustration of transformation between image coordinate system (ICS) and spherical coordinate system (SCS).....	28
Figure 3.5 Camera calibration by a vertical grid board. (a) An illustration of attaching the lines on the wall. (b) The intersections seen by camera are marked by yellow points.....	28
Figure 3.6 The camera views of Figure 3.5. (a) View of Figure 3.5(a). (b) View of Figure 3.5(b).	29
Figure 3.7 Detection of the cross point of a cross shape. (a) The look of a cross shape. (b) A center point and four lines composing the cross shape. (c) A circle with four intersections with the cross shape. (d) Use of four lines to detect the cross shape.	31
Figure 3.8 The matrix used to detect the cross shape.	31
Figure 3.9 Some experimental results of cross shape detection. (a) Original Image. (b) After thresholding. (c) Find groups of centers. (d) Center detected. (e) The bmp image with camera calibration information.....	32

Figure 3.9 Some experimental results of cross shape detection. (a) Original Image. (b) After thresholding. (c) Groups of centers detected. (d) Centers detected. (e) The bmp image with camera calibration information. (continued)	33
Figure 3.10 The program window of camera calibration.	33
Figure 3.11 An illustration of the experiment.	36
Figure 3.12 The distribution of the angles of the deviations.	36
Figure 3.13 The results of line fitting of the angle of deviation.	40
Figure 3.14 An illustration of odometer calibration.....	42
Figure 4.1 An illustration of proposed learning procedures.	44
Figure 4.2 The application for person following.	46
Figure 4.3 The top-view of light and illumination in the hallway.	47
Figure 4.4 An illustration of improvement of adopted process.	48
Figure 4.5 An illustration of person tracking strategy.....	49
Figure 4.6 An example of person tracking results. (a) A vehicle is following a person. (b) The person turns fast to right. (c) The vehicle turns an angle θ_{find} to find the person. (d) The vehicle finds the person. (e) The vehicle follows the person again.	50
Figure 4.7 Drafts of maps. (a) Map of an open indoor environment. (b) Map of a hallway environment.....	51
Figure 4.8 An illustration of detection of a person's facing direction.	53
Figure 4.9 An illustration of the facing direction of the person. (a) Front. (b) Back. (c) Left. (d) Right.	53
Figure 4.10 An illustration of the scan rectangle for human facing direction detection. (a) Front. (b) Back. (c) Left. (d) Right.	56
Figure 4.11 (a) A red rectangle including the monitored object as an interesting region. (b) A user interface to specify the horizontal line.	58
Figure 4.12 The relative position $C(x_r, y_r)$ of the vehicle with respect to the start point in the world coordinate system.	58
Figure 4.13 An illustration of learning of a monitored object automatically.....	62
Figure 4.14 Sobel Operator (a) x direction (b) y direction.	64
Figure 4.15 The relationship with x-y space and γ - θ space.	65
Figure 4.16 The accumulation array A of the Hough Transform.	66
Figure 4.17 An example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line...	67
Figure 4.17 An example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line.	

(continued)	67
Figure 4.18 Another example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line. ...	67
Figure 4.18 Another example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line. (continued)	68
Figure 5.1 Illustrations of path planning. (a) Backing to the start point directly. (b) Stepping on every path node precisely. (c) The planned path we desire. ...	70
Figure 5.2 Illustration of proposed path planning.....	72
Figure 5.3 An illustration of proposed dynamic path decomposition.....	74
Figure 5.4 Computation of the turning angle and move distance.....	76
Figure 5.5 An example of a planned path of an ellipse shape.....	77
Figure 5.6 An illustration of proposed patrolling process.	78
Figure 5.7 An example of experimental results. (a) The vehicle monitors an object. (b) The vehicle walks to a turning point (c) The vehicle monitors another object. (d) The vehicle goes back to the original start point.	79
Figure 6.1 The positions of ultrasonic sensors around the vehicle.	81
Figure 6.2 An illustration of the parameter of Width. (a) The hallway with two walls. (b) The hallway only has one wall.....	82
Figure 6.3 An illustration of the parameter of Directiondetect (a) Directiondetect = left. (b) Directiondetect = right. (c) Directiondetect = left + right. (d) Directiondetect = front.....	83
Figure 6.4 An illustration of the situation of adjusting the direction of the vehicle. (a) The vehicle should turn an angle to the right (b) The vehicle should turn an angle to the left.....	85
Figure 6.5 An illustration of navigation in a hallway.....	85
Figure 6.6 The navigation path. (a) An example of paths. (b) The path is transformed into an automation.....	87
Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map.....	88
Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map. (continued)	89
Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map. (continued)	90
Figure 7.1 An interface of the experiment. The green box shows the image stream and the blue box shows the input image at this moment. The yellow box	

	shows the difference image and the red box shows the draft of the map.	92
Figure 7.2	An experimental result of extraction of the clothes. (a) The input image. (b) The image of the extracted clothes.	93
Figure 7.3	An experimental result of the learning mode. (a) The input image. (b) The position of the vehicle.	93
Figure 7.4	An example result of learning a 2D object. (a) The input image. (b) The region of the 2D object. (c) The horizontal line of the 2D object.	94
Figure 7.5	An experimental result of following a person turning fast. (a) The input image. (b) The position of the vehicle.	94
Figure 7.5	An experimental result of following a person turning fast. (a) The input image. (b) The position of the vehicle. (continued).....	95
Figure 7.6	An experimental result of navigation. The blue line segments show the planning result. And the red points show the real patrolling path of the vehicle.	95
Figure 7.7	The experimental result of object monitoring and navigation path correction. (a) The numbers of monitored objects. (b) The vehicle monitors the objects. (c) The matching result and the horizontal line used for path correction.	96
Figure 7.8	An illustration of the path of following a person and monitoring objects in an experiment.	96
Figure 7.9	Interfaces of the experiment.	97
Figure 7.10	An illustration of learned data and navigation path.	97
Figure 7.11	An experimental result of navigation.	97
Figure 7.12	The experimental result of the vehicle guide a person. (a) Navigation in a hallway. (b) Navigation at a turning point.	98

Chapter 1

Introduction

1.1 Motivation

In recent years, autonomous land vehicles with ultrasonic sensors and vision-based robots have played helpful roles in human life. They can help us to do various tasks, such as:

- (1) keeping company with old people like a nurse to take care of them;
- (2) monitoring concerned objects;
- (3) navigating in desired environments automatically;
- (4) guiding people in indoor environment, and so on.

In security monitoring of objects, if we use stationary cameras, we have to install many of them and there might still be corners where the cameras cannot cover. But if we use an autonomous vehicle equipped with a video camera, which has the ability of automatic learning and human following, then the problem can be solved because the vehicle can follow people everywhere and learns the path that we hope it patrol, just like a mother teaching her child where to go.

While a vehicle follows a person, the vehicle has to detect the person in front of it continually. However, this is not easy because the brightness could not be all the same on the path, it could change to dark or bright slowly, and the person in the camera view might be darker or brighter than in the view a moment ago. The vehicle has to keep finding out the person to follow under variant brightness.

When the vehicle is patrolling, it may seem not smart if it walks totally the same

as the trajectory it learned, because this trajectory could be rough and there is no need to step on every spot of it precisely. We hope the vehicle could reconstruct the path into a smooth route, but not too smooth to miss some important points we want it to patrol.

In playing the role of a guide, it might be dangerous when the vehicle guides people in a narrow environment because the vehicle could hit the wall. But if we use an autonomous land vehicle equipped with ultrasonic sensors, the problem can be solved because the vehicle can then detect the distance to the wall using ultrasonic signals and keep navigating in the middle of the path safely.

For the application of person guidance, the vehicle needs often the additional functions of making audio introductions to the environment or the surrounding objects, and making warning announcements to people who stand on the way that the vehicle want to pass through.

In summary, the research goal of this study is to design an intelligent autonomous vehicle system with two kinds of capability. One is to learn the traversed path and objects which to be monitored, while following a person, and then navigate to the start point by automatic path correction by the use of computer vision and ultrasonic sensing techniques. For this goal, it is desired to design a system to be capable of automatic learning and path planning. The other capability is to navigate in a narrow indoor environment by ultrasonic sensing. It is desired to design a system to be capable of environment analysis in narrow indoor environments. By these functions, a variety of applications of the vehicle can be carried out, like being used as an autonomous patrolling assistant or a security guard, as well as a guide in an office environment, an exhibition area, or a tour route.

1.2 Survey of Related Studies

To achieve the mission of person following in indoor environments, the function of human detection is required at first for finding a targeted person in the person following process. Skin color is an important feature of humans. In the study of human skin colors, many methods have been proposed to build skin color models. Wang and Tsai [1] proposed a method which uses an elliptic skin model to detect human faces by color and shape features in images. They also proposed another method which detects a person not using the skin color, but using the colors of the clothes which the person wears. A study of human detection under variant illumination was conducted by Chen and Tsai [2]. They used many sets of color values of C_b and C_r to define *a function of skin-color reference models*. The function was then used to separate human skin regions from background.

Besides, some systems for person following have been proposed. Ku and Tsai [3] proposed a sequential pattern recognition method to decide the location of a person with respect to a vehicle and to detect a rectangular shape attached on the back of the person to achieve smooth person following. However, the person has to appear in the image all the time and the road has to be wide enough for this method to work. In applications of person following, Kwolek [4] proposed a method that determines the position of a mobile robot by laser readings. The tracking of the human head is done by a particle filtering technique using the features of color, depth, gradient, and shape. Morioka, Lee, Hashimoto [5] studied person following to provide people with services by the use of a human collaborative robot which tracks the back and shoulder of a person. There are some different studies for human detection and following by using different sensors. Treptow, Cielniak, and Duckett [6] proposed a method which tracks and identifies people from thermal and gray images.

For object monitoring, the vehicle has to learn the features of the concerned object and match the features to determine whether the object is the same as the previously learned one. Lowe [7] used a scale-invariant detector to find the extrema in the difference-of-Gaussian scale-space. He then created a *scale-invariant feature transform* (SIFT) descriptor to match key points using a Euclidean distance metric in an efficient best-bin first algorithm which can identify the nearest neighbors of points in high dimensional spaces.

For path planning, a method for planning the navigation task for a mobile robot under dynamic and unknown environments was proposed by Xiao, Liao, and Zhou [8]. For the topic of robot navigation, Chen and Tsai [9] proposed a method with a simplified SIFT for monitoring objects and used the position and feature information of objects to conduct vehicle navigation. Davison [10] proposed a method for monocular vision-based robot navigation.



1.3 Overview of Proposed Approach

The goal of this study, as mentioned previously, is to design two kinds of intelligent capabilities for autonomous land vehicles. One is to use computer vision and ultrasonic sensing techniques for person following and patrolling. The other is to use ultrasonic sensing for navigation in narrow indoor environments. The overall frameworks of proposed systems for these two capabilities are illustrated in Figure 1.1(a) and Figure 1.1(b).

The first proposed approach to person following and patrolling using a vision-based autonomous vehicle with an ultrasonic sensing system includes seven major stages. The first stage is camera calibration for measuring the distance between

the person and the vehicle by a so-called *angular mapping* technique [1] in which each point in the image represents a unique light ray from the viewpoint into the camera. By the angular information of light rays and the height of the camera, we can know the relative distances of targets in images. The details will be described in Chapter 3.

The second stage is human detection. An elliptic skin model to detect human faces by color and shape features in images was proposed by Wang and Tsai [1]. But this model is suitable only in a limited range of luminance. An improved skin color model which can adjust its elliptic center to adapt to the change of luminance was proposed by Chen and Tsai [9]. However the skin color detection is not suitable for everyone. We solve the problem by using cloth detection instead of skin color detection in this study.

The third stage is person following. It not only conducts the basic person following function but also deals with some unexpected situations. For example, it deals with environments where luminance is not uniform. This study solves these problems by dynamically adjusting the values of C_b and C_r of the clothes worn by the person to detect the person in the camera frame.

The fourth stage is learning the path data during person following. The vehicle needs to learn two kinds of data. One is the data of a path node which are used for path planning in the next cycle. The other is object monitoring data which are used to conduct security patrolling. The details will be described in Chapter 4.

The fifth stage is path planning by the mean square error (MSE) method using the path data collected during person following. The vehicle reconstructs the path it has learned into a smooth path. In addition, it is also wanted in this stage to automatically adjust the threshold of the mean square error for adaptation to different environments like wide or narrow ones. The details will be described in Chapter 5.

The sixth and seven stages compose the navigation phase. The vehicle uses the information learned and the path planned before for executing the mission of patrolling. It monitors objects using a simplified SIFT method [9] and a navigation technique using 2D object image matching.

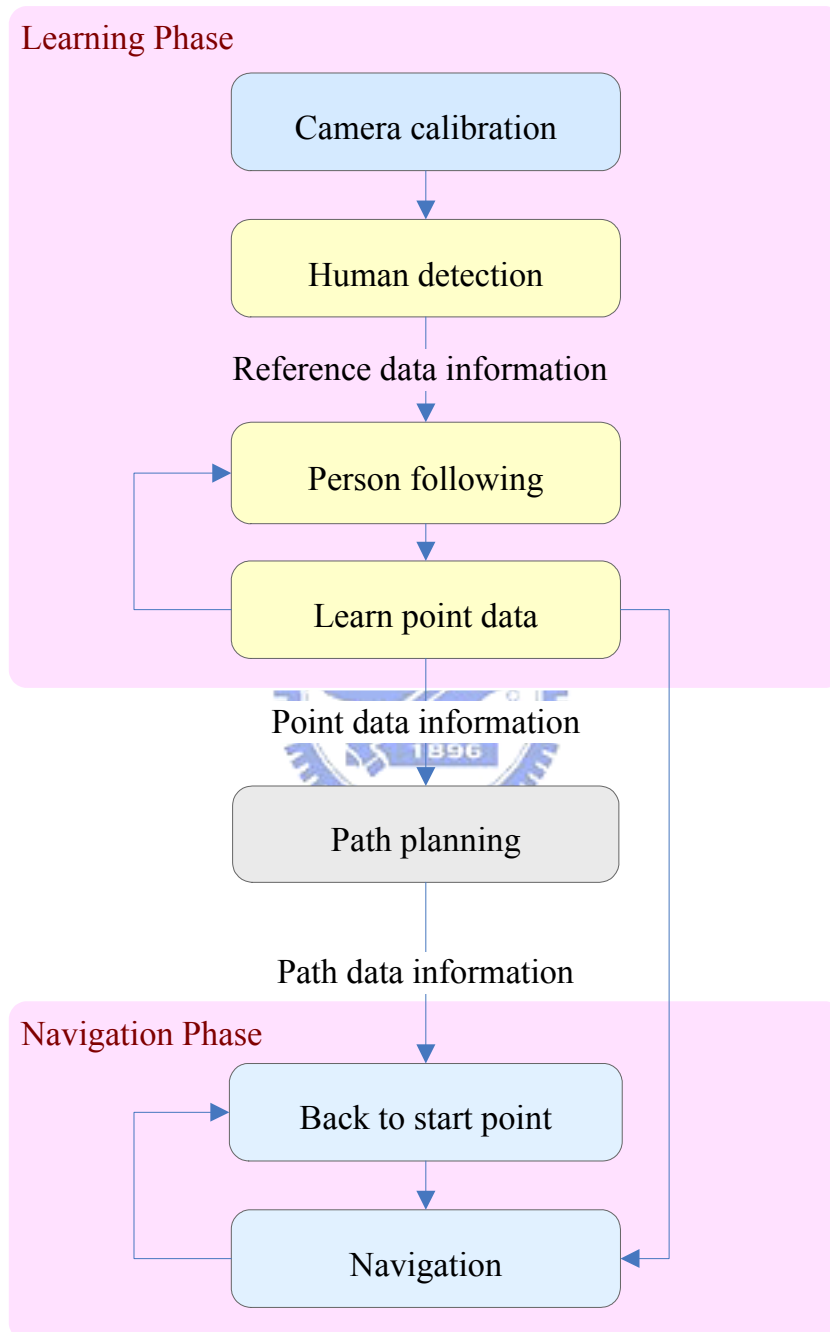
The second proposed approach to autonomous land vehicle guidance in indoor environments with ultrasonic sensors includes two major phases. The first is the learning phase, and the second is the guidance phase. In the learning phase, the vehicle needs to learn the information of path data and point data. In the guidance phase, the vehicle analyzes the signal detected from the ultrasonic sensor and guides people on the path which was learned in the learning phase. The details will be described in Chapter 6.

1.4 Contributions

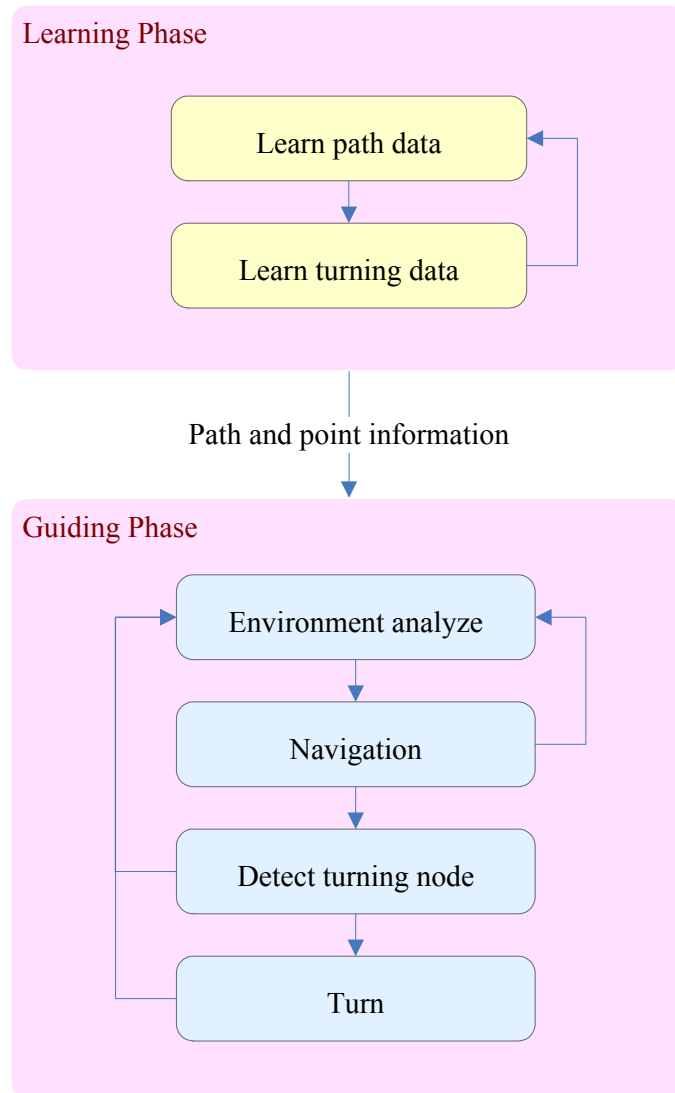


The major contributions of this study are summarized as follows.

- (1) A method for improving the practicability of camera calibration is proposed.
- (2) A method for person following in environments where the luminance is not uniform is proposed.
- (3) A method for detecting a person who turns fast and disappears is proposed.
- (4) A path planning method which can adapt to different environments is proposed.
- (5) A method of computing the angle to calibrate the odometer of the vehicle is proposed.
- (6) A guidance method using sequences of ultrasonic signals is proposed.



(a) Person following and patrolling technique.



(b) Tour guiding in narrow indoor environments technique.

Figure 1.1 Flowcharts of proposed systems.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we describe the system configuration of the vehicle and the principles of proposed learning,

human following, and guidance techniques. In Chapter 3, the proposed the method for improving the practicability of camera calibration, the method for calibration of the odometer, and the method of using the reference data for distance computation are described. In Chapter 4, a method for improving the practicability of human facing direction detection is described. The proposed techniques for path planning are described in Chapter 5. The method for learning the path of guidance is described in Chapter 6. Some satisfactory experimental results are shown in Chapter 7. Finally, some conclusions and suggestions for future works are given in Chapter 8.



Chapter 2

System Configuration and Guidance Principles

2.1 Introduction

There had many studies about using machines to reduce the burden of humans to do the work of patrolling. The autonomous land vehicle with camera equipments is commonly used for this kind of work because cameras could “see” objects which need be monitored in replacement of human eyes. Usually, we have to key in the path of a patrolling vehicle manually into a computer and such work of assigning a path to a vehicle is not easy to perform for a user who is not good at machine handling. This inconvenience can be resolved if we have an autonomous vehicle equipped with a camera which has the abilities of following people and remembering the path it walked through. With such an intelligent vehicle, we can then bring it to walk through the path once, and the vehicle can remember the path and walk the way back to the start point by itself. Additionally, we can change the patrol path adaptively for various application needs easily.

On the other hand, when we visit an unfamiliar place like an office in a building, we often need a guide to bring us from the outside of the elevator into the office. It is desirable too that the guide could give some introduction to the environment. We can use a robot instead of a human being to do this job. However, when the environment is not broad, then the robot could hit the wall. The problem could be solved if we have

an autonomous land vehicle with ultrasonic sensors, which can keep the vehicle navigating in the middle of the path using the ultrasonic signals provided by the sensors.

The entire hardware equipments and software used in this study are described in Section 2.2. The system of person following and patrolling includes two major stages to reach its goal. The first is the learning process which makes the vehicle more intelligent. In Section 2.3, we will introduce the principle of the proposed learning technique. We will describe how to follow a person and learn the information of paths. The second stage is the path planning process which can refine the learned path. In Section 2.4, we will describe the major steps of refining the path obtained from the learning process. For the system of the guiding vehicle, we will introduce the principles of the learning and guiding strategies in Section 2.5.



2.2 System Configuration

In our system, we use the Pioneer 3, a vehicle made by ActiveMedia Robotics Technologies Inc., as a test bed. The vehicle is equipped with a pan-tilt-zoom camera of which we can adjust some parameters by a graphical user interface, such as the image resolution, the image format, and so on.

Because the wireless signal might be interfered by unknown signals sometimes, we control the vehicle by a network cable connecting our computer and the vehicle in our system. In this way we can have stable communication among the vehicle, the program, and the PTZ camera. A diagram illustrating this configuration is shown in Figure 2.1.

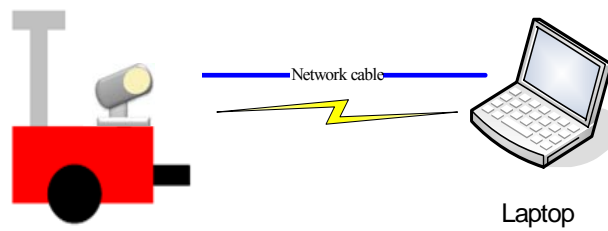


Figure 2.1 Equipment connection situations in this study.

2.2.1 Hardware configuration

The hardware equipments we use in our system include three parts. The first part is a laptop which we use to run our program. A kernel program can be executed on the laptop to control the vehicle by issuing commands to the vehicle. It also can be used to get the status information of the vehicle.

The second part is the vehicle which has an aluminum body. The size of it is 44cm×38cm×22cm with three wheels of the same diameter of 16.5cm. In the vehicle, there are three 12V batteries each of which, by one charge, supplies power to the vehicle to run 18-24 hours. The vehicle can reach a forward speed of 160cm per second and a rotation speed of 300 degrees per second. The embedded control system can be used to control the vehicle to move forward or backward and turn around by the user's commands. The appearance of the vehicle is shown in Figure 2.2.

The third part is a digital IP camera with panning, tilting, and zooming (PTZ) capabilities. The PTZ IP camera used in this study is an AXIS 213 PTZ made by AXIS, as shown in Figure 2.3. This is a camera with a height of 130mm, a width of 104mm, a depth of 130mm, and a weight of 700g. The pan angle range is 340 degrees and the tilt angle range is 100 degrees. It has 26x optical zooming and 12x digital

zooming capabilities. The image captured in our experiments is of the resolution of 320×240 pixels for the reason of raising image processing efficiency. Moreover, the camera is directly connected to a laptop by a network cable for transmission of the captured image.

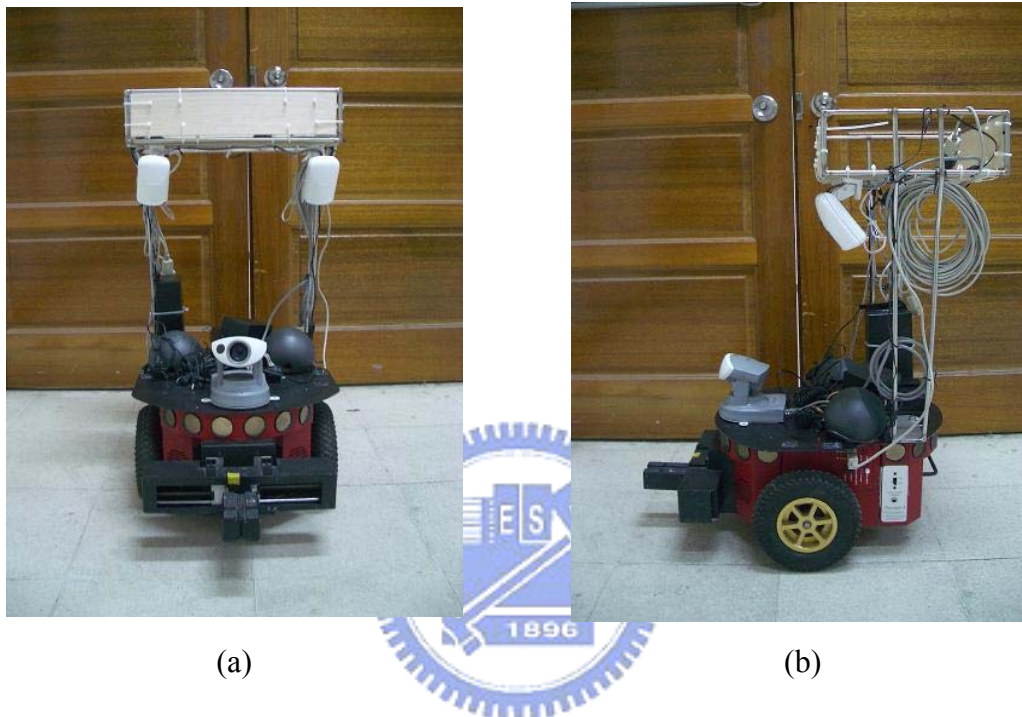


Figure 2.2 The vehicle Pioneer 3 used in this study. (a) A front view of the vehicle. (b) A side view of the vehicle.

2.2.2 Software configuration

The ActiveMedia Robotics provides an application interface ARIA to control the vehicle used in this study. ARIA is an object-oriented interface which is usable under Linux or Win32 in the C⁺⁺ language and can dynamically control the velocity, heading, and other navigation settings of the vehicle. We use the ARIA to communicate with the embedded system of the vehicle. And we use the Borland C⁺⁺ Builder as the development tool in our experiments.



(a)



(b)



(c)

Figure 2.3 The pan-tilt-zoom camera used in this study. (a) A perspective view of the camera. (b) A front view of the camera. (c) A left-side view of the camera.

2.3 Learning Strategy and Major Steps in Proposed Process

The proposed learning strategy is based on human following [2]. When the vehicle follows a person, it uses the information of the clothes which the person wears.

Two kinds of situations should be handled. The first is that the vehicle can find the person by the clothes feature and the second is that the vehicle cannot find the person.

In the first situation, the vehicle will keep following the person and avoid losing the observation of the person. In the second situation, the vehicle cannot follow the person anymore. Because the camera on the vehicle always pans to follow the person, the vehicle can keep knowing that the person is on which side with respect to the vehicle (left or right). If the person disappears in front of the vehicle, then the vehicle can find out the side on which the disappearing person was. After that, the vehicle will turn an angle to the correct its direction to search for the disappearing person in the acquired image. But if the vehicle turns for a pre-set number of times and still cannot find the person, then the system stops the learning strategy.

Our system needs to learn two major kinds of data at the learning strategy. The first part is path data. These data are used in the path planning. The second part is the information of those objects which the person wants the vehicle to monitor. The person only needs to stand in front of the object and looks at it. Then the vehicle will go forward to learn the information of the object. These data are used in the patrolling process. The major steps are shown in Figure 2.4.

2.4 Path Planning and Patrolling Principle and Major Steps in Proposed Process

After the vehicle learns the path and the object data in the learning stage, the vehicle refines the path and stretches the rough part of the path by analyzing these

data using a binary-cut line fitting technique. The path planning process has three major steps. The first step is to retain the nodes where the monitored objects are located. After doing the first step, the path is separated into some segments. Then, we collect these path segments by pushing them into an empty queue.

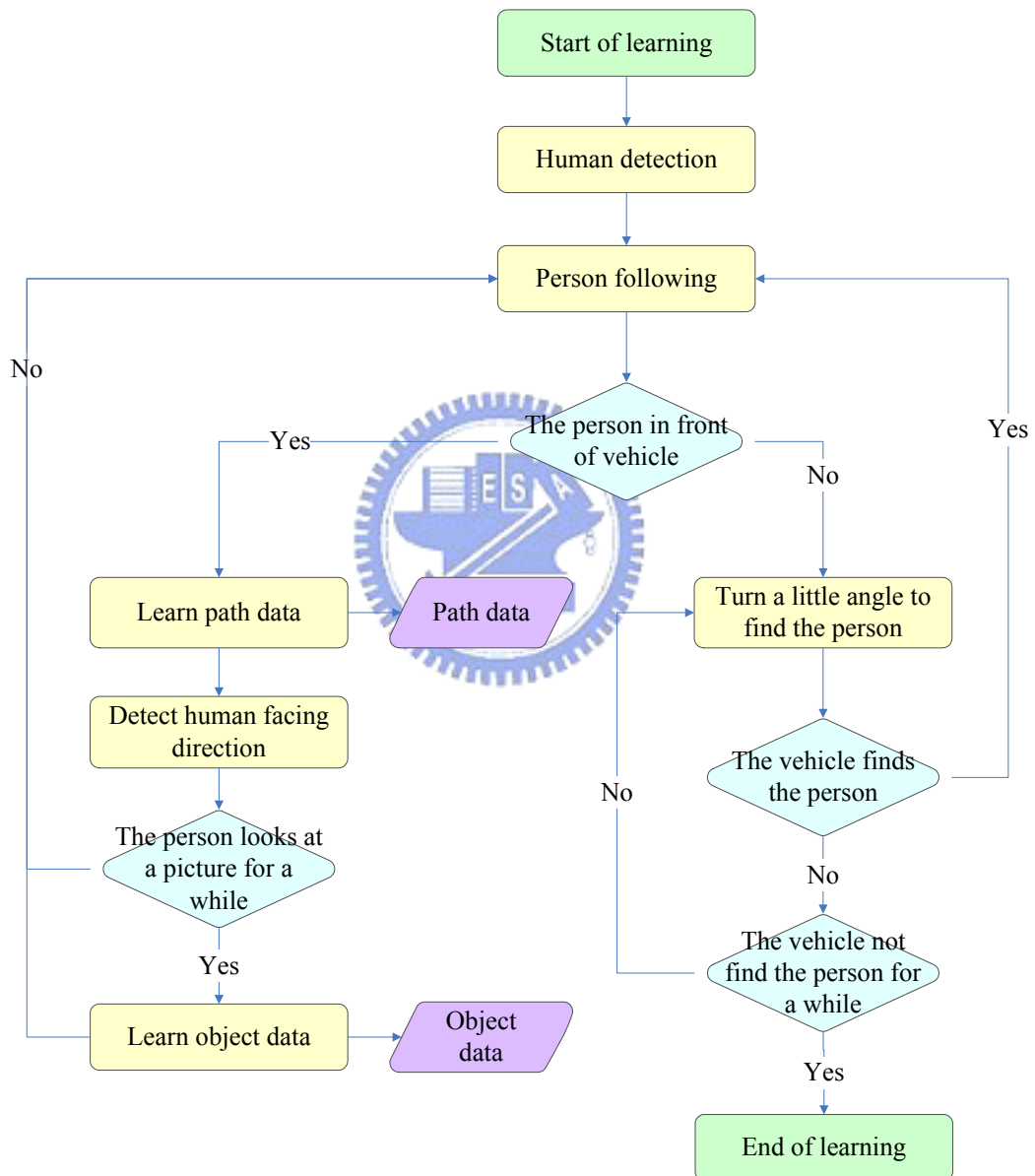


Figure 2.4 An illustration of the automatic learning process by person following.

The second step that is to pop out one segment from the queue and check the segment to see whether it is smooth enough or not. If the answer is no, then go to the third step to separate the segment again into two small parts and push them into the queue mentioned before. We then execute the second step again and repeatedly, until the queue is empty. We save the segments which are smooth enough to be the path data, for use in the patrolling process. The major steps are illustrated in Figure 2.5.

After doing the steps of the path planning, the next process is navigation to the start point. The navigation process includes three major steps. The first step is to read and sort the path data obtained from the path planning. The second step is to go forward according to the path data, and check the current position to see whether there is an object to be monitored there or not. If the answer is yes, then go to the third step to match the monitored object. We repeat the second and the third steps until the vehicle arrives at the start point. The major steps are illustrated in Figure 2.6.



2.5 Vehicle Navigation Principle and Major Steps in Proposed Process

Our system has two major processes in the vehicle navigation system. The first is the learning process. The vehicle learns the information of a navigation path in this process. We divide the navigation path into several segments, and the divided points are the positions where the vehicle needs to turn. Also, the vehicle needs to learn two kinds of major information of every segment of the navigation path. One is the information of the environment of the path where the vehicle navigates and the other is the information of a turning node which the vehicle passes by. The major steps are

illustrated in Figure 2.7.

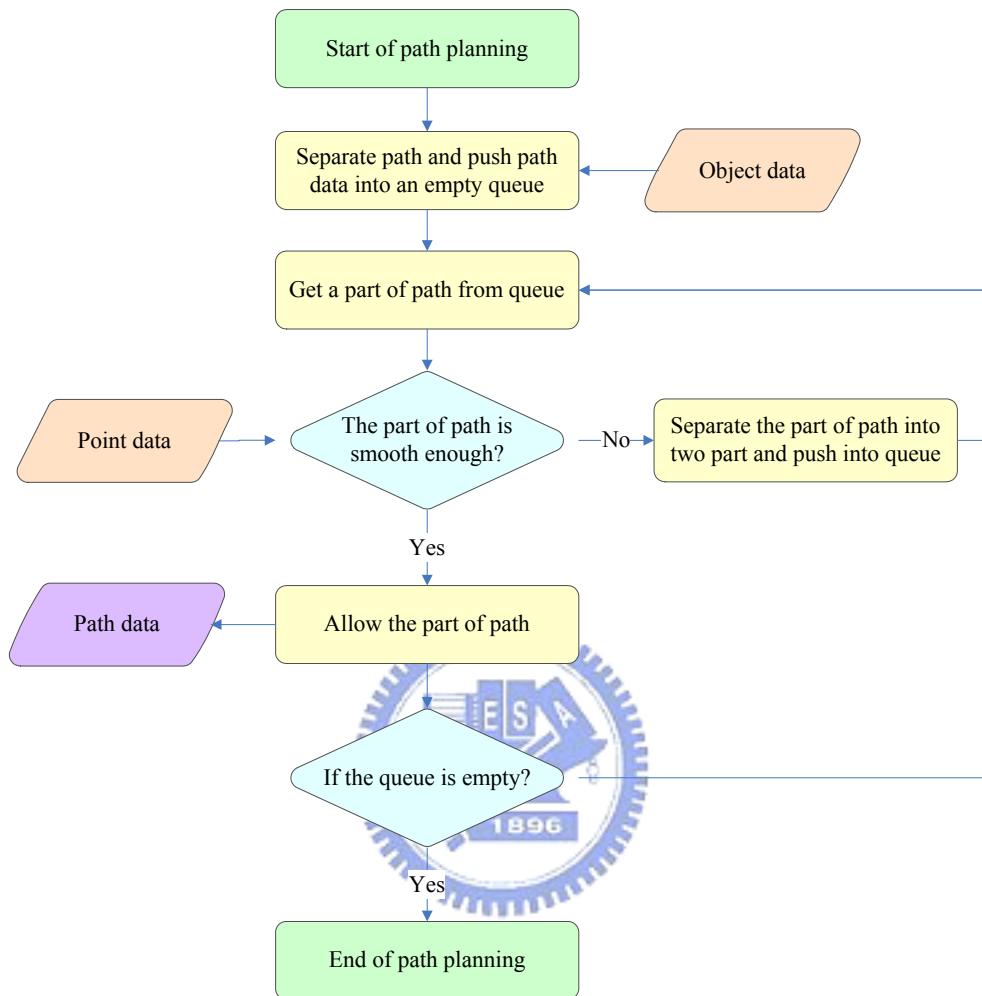


Figure 2.5 An illustration of the path planning process.

The second part of our system is the navigation process. After reading the data of the navigation path which is separated into several segments by the turning nodes obtained from the learning process, the vehicle can now navigate and guide, if necessary, a visitor on the path, by performing two tasks at every segment. One is navigation in the middle of the hallway and the other is finding the position of the turning node. These major steps are illustrated in Figure 2.8.

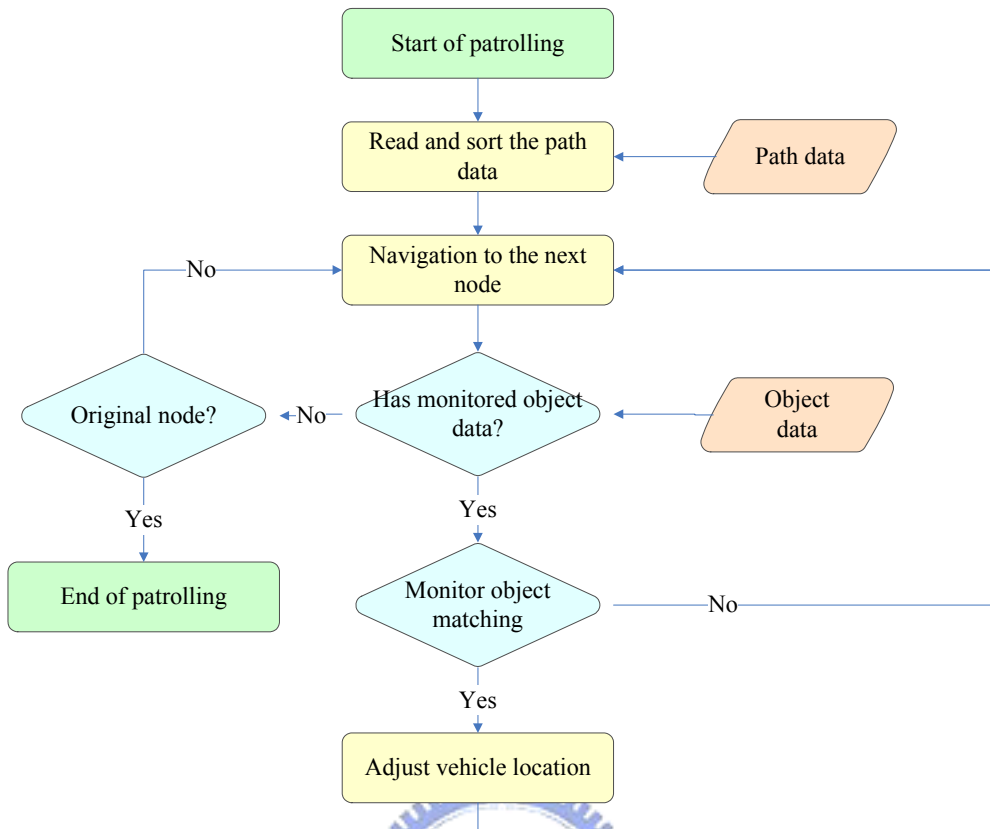


Figure 2.6 An illustration of the patrolling process.

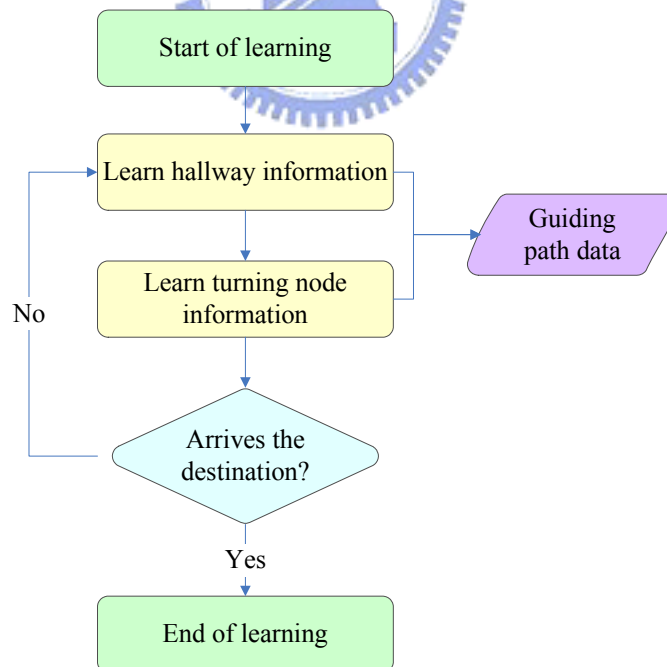


Figure 2.7 An illustration of the learning process of the vehicle navigation system.

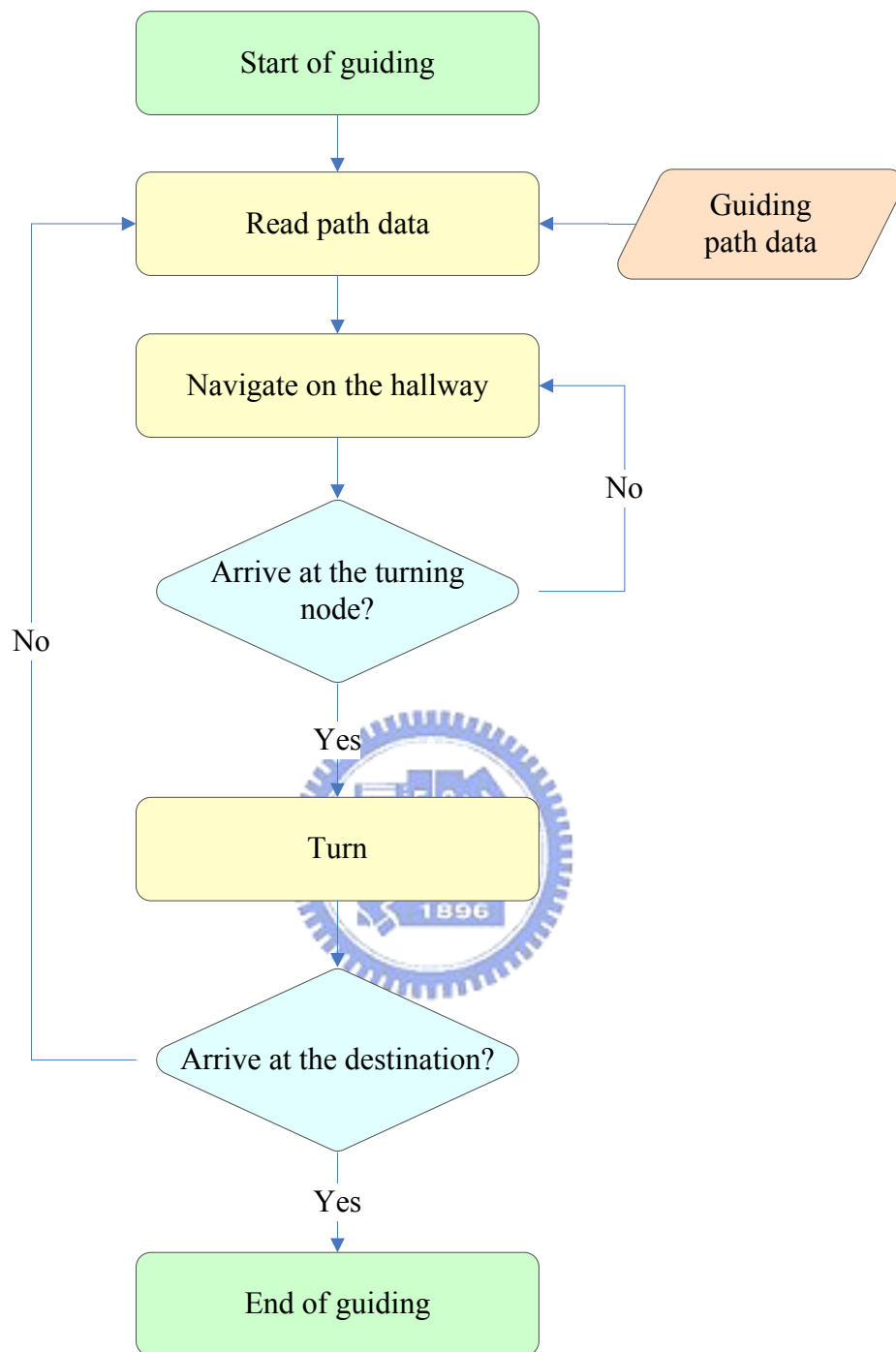


Figure 2.8 An illustration of the navigation process.

Chapter 3

Camera and Odometer Calibration

3.1 Introduction

The camera and the odometer are the two important equipments of the vehicle used in this study. In this chapter, we describe the proposed calibration methods for these two equipments.

Distance information is important and useful in the person following process. The vehicle of our system computes the distance information by analyzing images captured from the camera. But there is ambiguity in the inverse mapping from 2D image coordinates to 3D world positions. Wang and Tsai [1] proposed a method of angular mapping for camera calibration to compute the distance between a vehicle and a person. Chen and Tsai [9] proposed a method of area tracking to deal with the situation where the vehicle cannot see the entire clothes of the person. In Section 3.2, we will review these methods and propose another method which can improve the practicability of these camera calibration methods.

For vehicle navigation in indoor environments, the vehicle location is the most important information to guide the vehicle in correct paths. Though the information provided by the odometer of the vehicle is precise enough for inferring the vehicle location for most applications, it cannot be used solely for the navigation process because the incremental mechanical errors might result in imprecise odometer data and so in deviations from the planned navigation path. Hence, in order to keep the navigation in track, a vision-based odometer calibration technique is desirable to

eliminate the errors. In Section 3.3, we will propose a technique for this purpose.

3.2 Camera Calibration by Cross Shape Detection

3.2.1 Idea of Proposed Camera Calibration Method

The distance information is indispensable for a person-following vehicle. The vehicle can avoid striking a person by using the distance information to keep a safe distance to the person. And the vehicle can go forward to see more clearly for avoiding losing information of this person by using the distance information when the person is far from the vehicle. Our system computes the distance information by analyzing the images captured from the camera. Through imaging with the camera, 3D world coordinate systems are mapped into 2D image coordinate systems. However, there is ambiguity in the inverse mapping from a 2D image point to its corresponding 3D world location because each point in the image is the projection result of a light ray onto the image sensor. The light ray can be described by a longitude angle and a latitude angle in the 3D world space. To define the corresponding longitude and latitude angles (or simply called *longitude* and *latitude* in the sequel) of each point in images, Wang and Tsai [1] proposed a method of 2D mapping to achieve the goal of *angular-mapping camera calibration*.

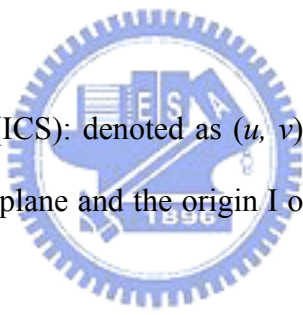
However, some steps repeat and incur error easily in their method. We will propose a method which simplifies the steps of camera calibration and is suitable for common users. In Section 3.2.3, we will review the method of angular-mapping

camera calibration. In Section 3.2.4, we will describe the proposed method.

Before describing the above-mentioned methods, we first introduce the definitions of the coordinate systems and the directional angle of the camera used in this study in Section 3.2.2.

3.2.2 Coordinate Systems and Directional Angles of Camera

Four coordinate systems are utilized in this study which describes the relative locations between the vehicle and encountered objects. The coordinate systems are shown in Figure 3.1. The definitions of all the coordinate systems are stated in the following.

- 
- (1) Image coordinate system (ICS): denoted as (u, v) . The uv -plane of the system is coincident 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 x -axis and the y -axis are defined to lie to on the ground, and the origin G of the global coordinate system is a pre-defined point on the ground. In this study, we define G as the starting position of the person-following process.
 - (3) Vehicle coordinate system (VCS): denoted as (V_x, V_y) . The V_xV_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 system is parallel to the line segment joining the two driving wheels and through the origin V . The V_x -axis is perpendicular to the x -axis and goes through V .
 - (4) Spherical coordinate system (SCS): denoted as (ρ, θ, φ) . This system is proposed

by Wang and Tsai [1]. It is a 3D polar coordinate system and we explain this system in terms of the 3D Cartesian coordinate system with coordinates (i, j, k) for convenience. The origin S of the spherical system, which is also the origin of the Cartesian system, is the optical center of the camera. The ij -plane of the Cartesian system is parallel to the uv -plane in the ICS. A point P at coordinates (i, j, k) in the Cartesian space is parallel to the uv -plane in the ICS. A point P at coordinates (i, j, k) in the Cartesian space is represented by a 3-tuple (ρ, θ, φ) in the spherical space. The value ρ with $\rho \geq 0$ is the distance between the point P and the origin S . The longitude θ is the angle between the positive k -axis and the line from the origin S to the point P projected onto the ik -plane. The latitude φ is the angle between the ik -plane and the line from the origin S to the point P .

Two kinds of directional angles of a camera are used in this study. One is the pan angle and the other the tilt angle. The pan angle of the camera is defined in the VCS, denoted by θ_c . It represents the degree of horizontal rotation of the camera and is important for coordinate transformation.

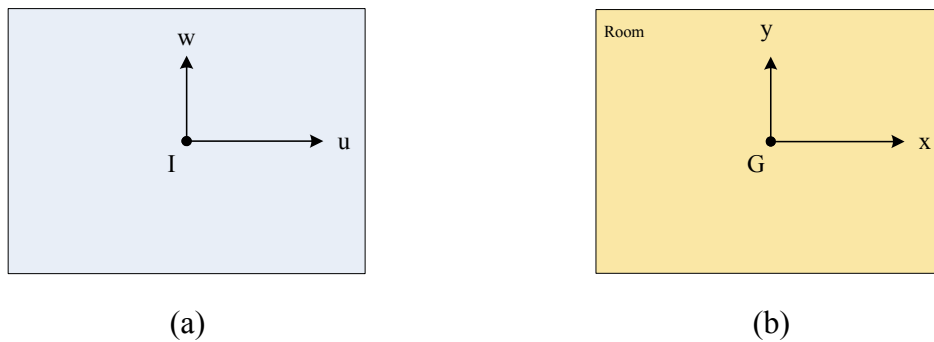


Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The global coordinate system (c) The vehicle coordinate system. (d) The spherical coordinate system.

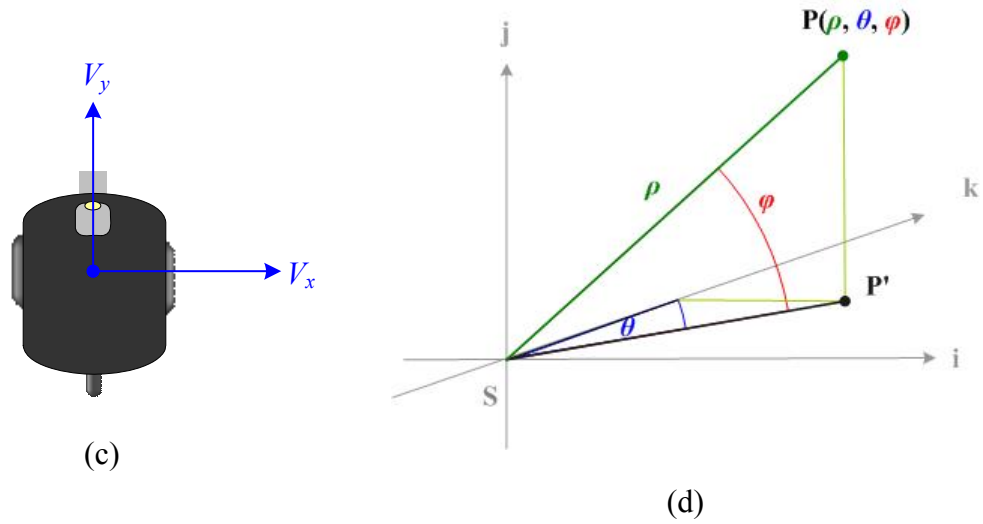


Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The global coordinate system (c) The vehicle coordinate system. (d) The spherical coordinate system. (continued)

We define the direction of the y -axis to be zero. The value of θ_c is exactly the angle between the camera direction and the direction of the y -axis. The range of θ_c is between 0 and π if θ_c is in the first and fourth quadrants and between 0 and $-\pi$ if θ_c is in the second and third quadrants, as shown in Figure 3.2. The tilt angle of the camera is defined as the angle between the optical axis of the camera and the ground. The angle, denoted as φ_c , represents the vertical tilting of the camera. We define the angle to be zero when the optical axis of the camera is parallel to the ground. The range of φ_c is between 0 and $\pi/2$ if the camera tilts up, and is between 0 and $-\pi/2$, else, as shown in Figure 3.3.

3.2.3 Review of Adopted Camera Calibration Method

Wang and Tsai [1] proposed a nonlinear angular mapping method to precisely obtain the angular transformation from the real world to the image. By the angular

information of the light rays and the height of the camera, we can know the relative distances of targets in images.

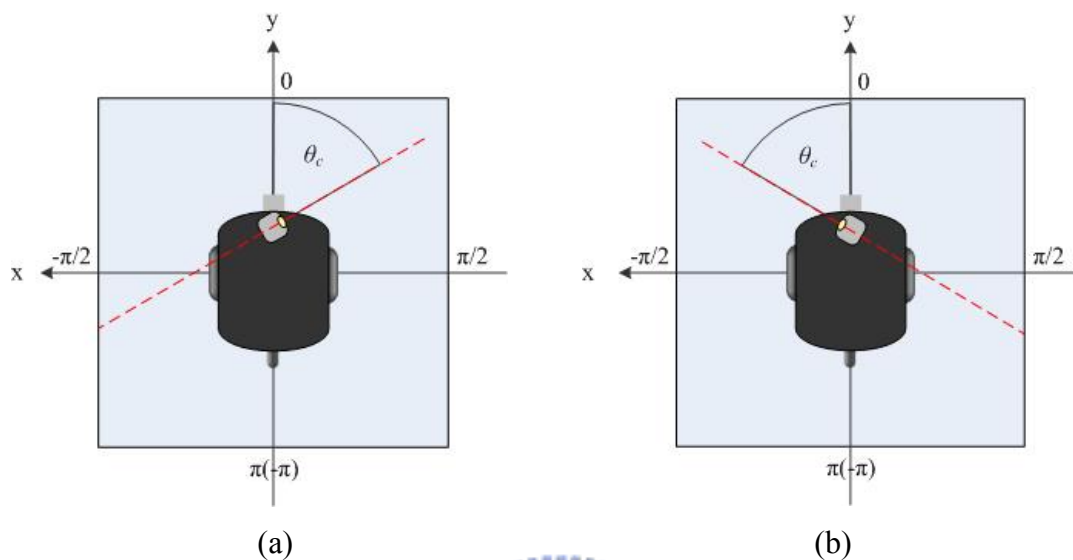


Figure 3.2 The pan angle of the camera. (a) $0 \leq \theta_c \leq \pi$. (b) $0 \geq \theta_c \geq -\pi$.

The coordinate system used in this method is shown in Figure 3.4, which includes the previously-mentioned image coordinate system (ICS) described by image coordinates (u, v) and the spherical coordinate system (SCS) described by parameters (ρ, θ, φ) . The latter is a 3D polar coordinate system which can be explained in terms of the 3D Cartesian coordinate system with coordinates (i, j, k) . The ij -plane of the Cartesian system is parallel to the uv -plane in the ICS. The origin S of the SCS, which is also the origin of the Cartesian system, is the optical center of the camera. A point P at coordinates (i, j, k) in the Cartesian space is represented by a 3-tuple (ρ, θ, φ) in the SCS where ρ is the distance between the point P and the origin S . The longitude θ is the angle between the positive k -axis and the line from the origin S to the point P projected onto the ik -plane, and the latitude φ is the angle between the ik -plane and the line from the origin S to the point P .

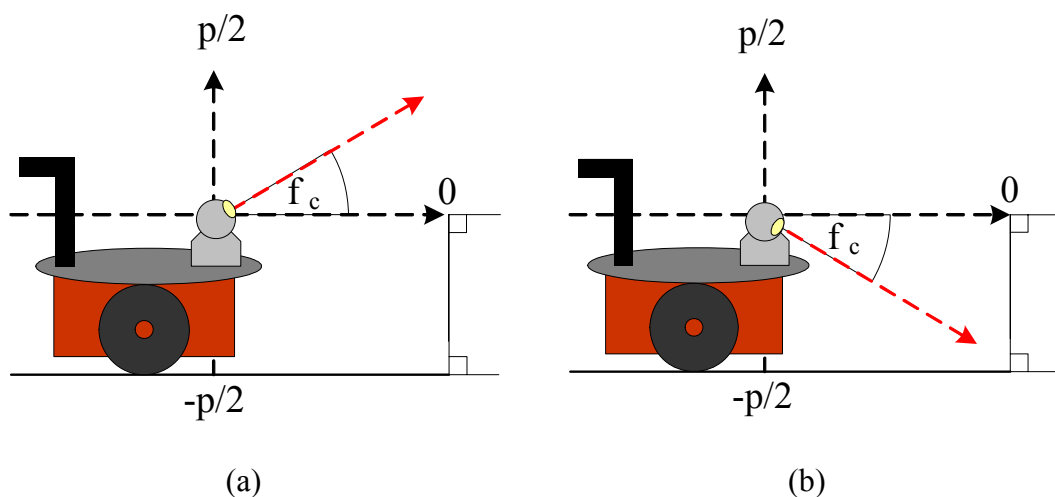


Figure 3.3 The tilt angle of the camera. (a) $0 \leq \varphi_c \leq \frac{\pi}{2}$. (b) $0 \geq \varphi_c \geq -\frac{\pi}{2}$.

A grid board is used in this method. It has m vertical lines and n horizontal lines, and is attached on a wall which is perpendicular to the ground. Because the longitude and the latitude values of the intersection points in the grid have been known in advance, the longitude and the latitude values of the other pixels in the image can be computed by an interpolation method. In this way, the longitude and the latitude values of each pixel in the image can be obtained. In Figure 3.5, we can see the respective positions of the grid board and the camera use in this method. And the views from the camera are shown in Figure 3.6. The intersections of the lines are marked by yellow points.

After knowing the longitude and the latitude values of the yellow points in the image by previously-mentioned nonlinear angular mapping, we can use an interpolation method to compute the longitude and the latitude values of the other pixels in the image.

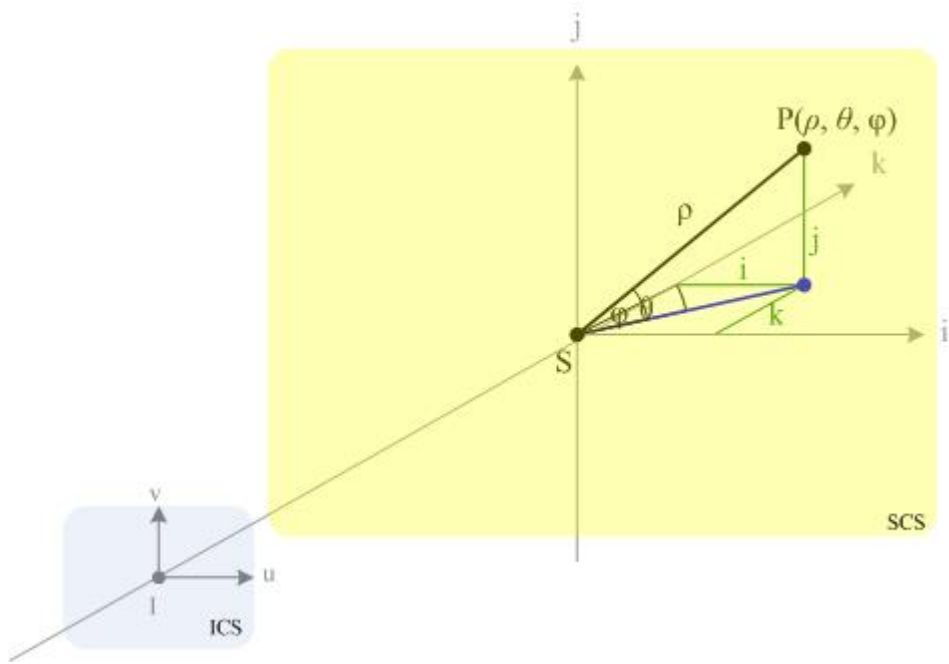


Figure 3.4 An illustration of transformation between image coordinate system (ICS) and spherical coordinate system (SCS).

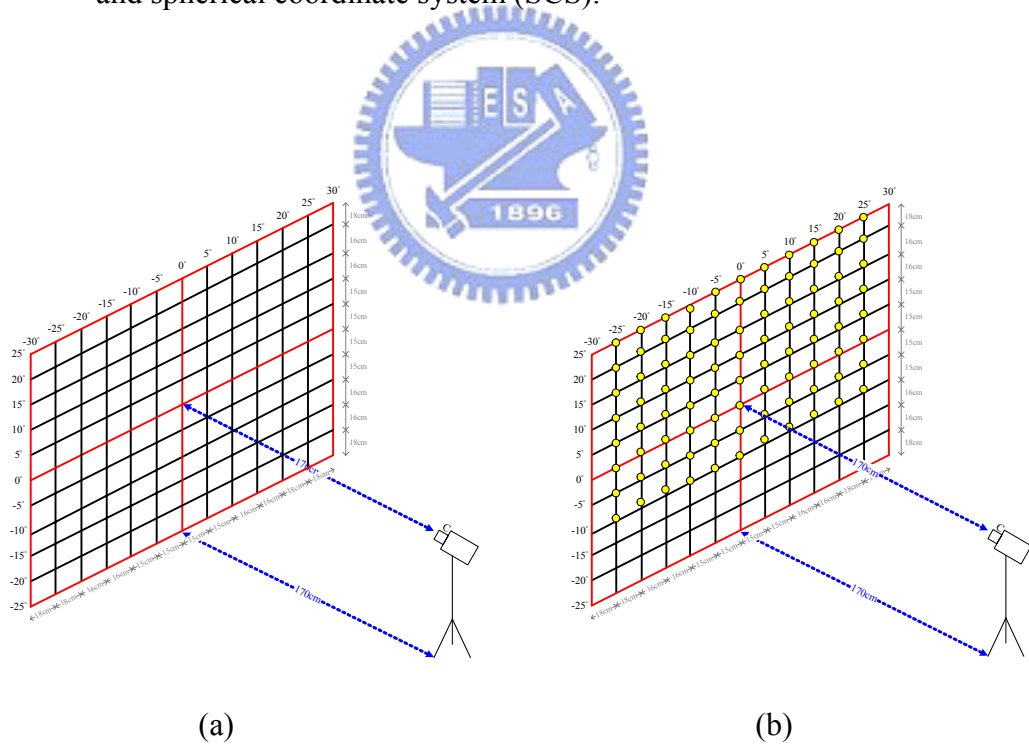
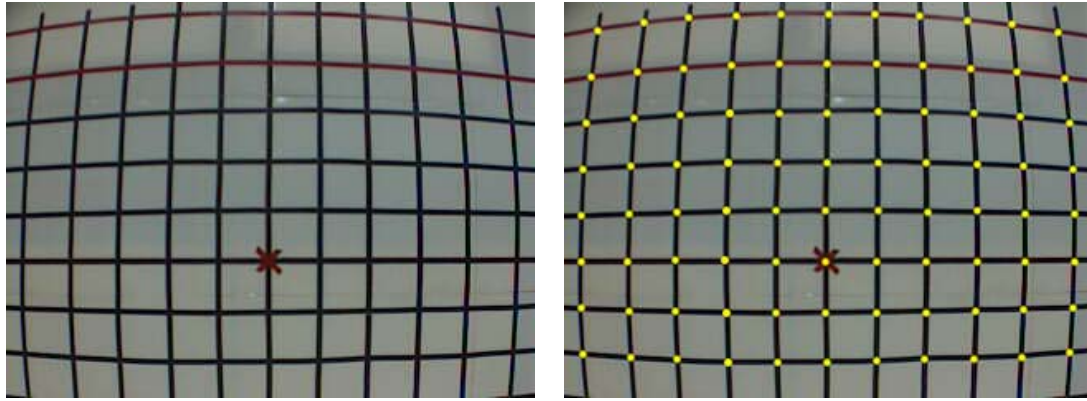


Figure 3.5 Camera calibration by a vertical grid board. (a) An illustration of attaching the lines on the wall. (b) The intersections seen by camera are marked by yellow points.



(a)

(b)

Figure 3.6 The camera views of Figure 3.5. (a) View of Figure 3.5(a). (b) View of Figure 3.5(b).

3.2.4 Proposed Cross Shape Detection Technique

By using the camera calibration method which is mentioned in Section 3.2.2, we can convert the coordinate values of every pixel in the image coordinate system (ICS) into the spherical coordinate system (SCS). An important step is to mark the intersection points in an image of the grid board. However, it is required to point out every intersection point manually in this method. The work is repeated many times and is prone to incur errors. To deal with this problem, we propose a technique of cross shape detection which can be adopted to locate the intersection points in the image automatically.

A look of a cross shape which has two lines intersecting each other is shown in Figure 3.7(a). For detecting the cross shape, we have to find out its property. A different definition of cross shape is a shape which has a center point C with four lines L_1 , L_2 , L_3 and L_4 emerging out of the center in four directions, as shown in Figure 3.7(b). Let C denote the center. After we draw a circle, it can be divided into four

curve parts, P_1P_2 , P_3P_4 , P_5P_6 and P_7P_8 , intersecting the four lines of the cross shape respectively, as shown in Figure 3.7(c). By a careful observation around the circle, we can discover a common property of the eight intersection points, P_1 to P_8 , that is, the positions of the eight points are located at places where the cross shape changes its color from white to black or from black to white. That means if we choose a point on the grid board randomly, let the point be the center to draw a circle, and find out the color changes eight times through the circle, then we can decide that the point is the center point of a cross shape. On the contrary, if we choose a point on the grid board randomly, let the point be the center to draw a circle, and find out that the color changing times are not equal to eight, then we can say that the point is not the center point of the cross shape.

We can simplify this technique by using four lines instead of using the circle, as shown in Figure 3.7(d). That is, we check every point of the foreground on the grid board by using a matrix, which is shown in Figure 3.8, to scan the four lines composed of yellow cells and count the times of color changing. If the color of the grid board changes eight times, then we can decide C to be the center of the cross shape. The detailed process is described in the following as an algorithm.

An experimental result of cross shape detection is shown in Figure 3.9. Figure 3.9 shows the image captured from the camera. Figure 3.9(b) is the thresholding result of Figure 3.9(a). In Figure 3.9(c), many groups of centers of cross shapes are detected. Figure 3.9(d) shows the final result in which the real center of every cross shape have been found out. We conducted the calibration work and saved the longitude and the latitude values of every pixel in the image into a bmp image as shown in Figure 3.9(e). Figure 3.10 shows the program windows for camera calibration.

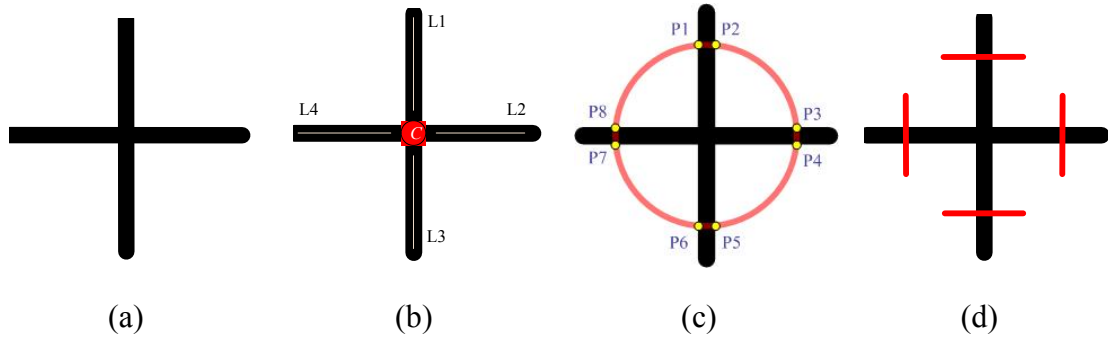


Figure 3.7 Detection of the cross point of a cross shape. (a) The look of a cross shape. (b) A center point and four lines composing the cross shape. (c) A circle with four intersections with the cross shape. (d) Use of four lines to detect the cross shape.

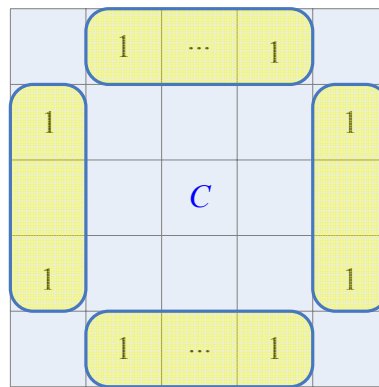


Figure 3.8 The matrix used to detect the cross shape.

Algorithm 3.1. *Computing the intersection points in a calibration target image.*

Input: An image taken from camera I_c , and the gray version I_g of I_c .

Output: A set S of intersection points of the image I_c , with coordinates $(x_{c1}, y_{c1}), (x_{c2}, y_{c2}), \dots, (x_{cn}, y_{cn})$

Steps:

Step 1. Calculate a threshold value T of the gray value which can differentiate the background and the foreground of I_c .

Step 2. Reset the gray value g_{pi} of every pixel p_i in the image I_g in the following way:

If $g_{pi} > T$ is satisfied, set g_{pi} as a foreground point;
else,

if $g_{pi} \leq T$ is satisfied, set g_{pi} as a background point.

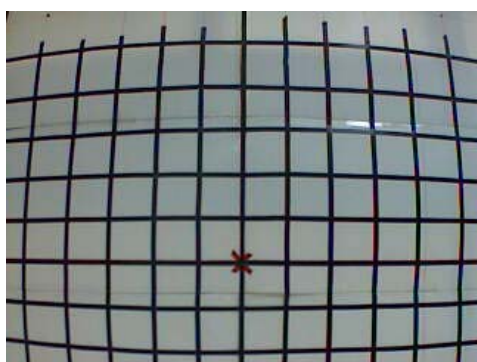
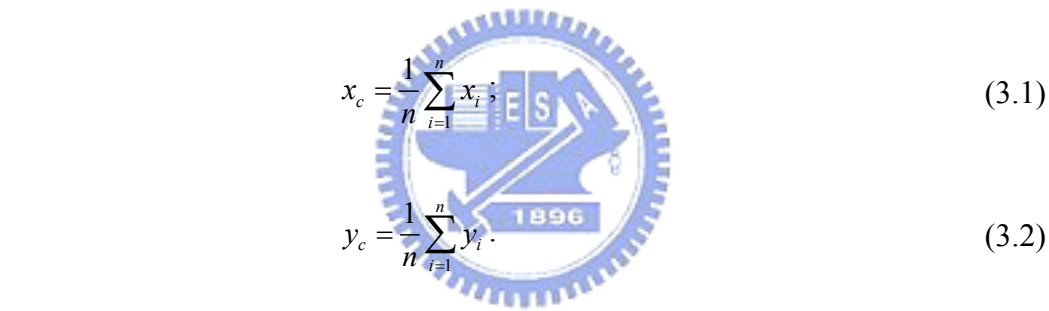
Step 3. Calculate the times t_i of color changing around every pixel of the foreground.

Step 4. Delete the point p_i whose value t_i is not equal to eight, and consider the remaining points as centers.

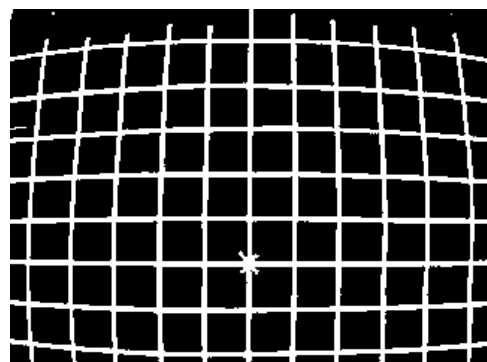
Step 5. Calculate the centroid of every group of centers and find out the intersection points by Eqs. (3.1) and (3.2) below:

$$x_c = \frac{1}{n} \sum_{i=1}^n x_i; \tag{3.1}$$

$$y_c = \frac{1}{n} \sum_{i=1}^n y_i. \tag{3.2}$$

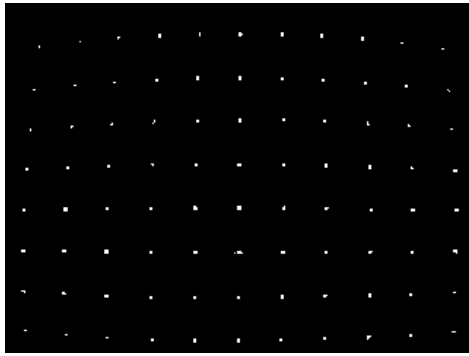


(a)

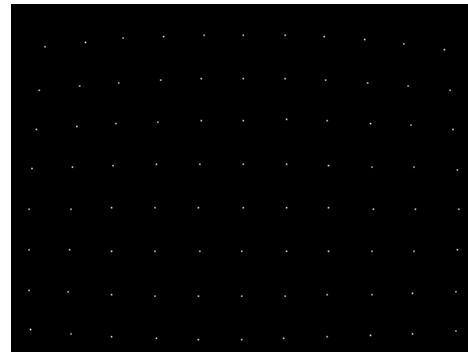


(b)

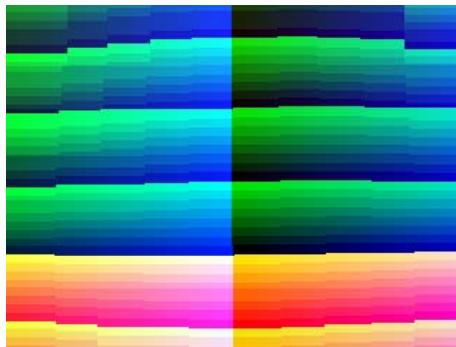
Figure 3.9 Some experimental results of cross shape detection. (a) Original Image. (b) After thresholding. (c) Find groups of centers. (d) Center detected. (e) The bmp image with camera calibration information.



(c)



(d)



(e)

Figure 3.9 Some experimental results of cross shape detection. (a) Original Image. (b) After thresholding. (c) Groups of centers detected. (d) Centers detected. (e) The bmp image with camera calibration information. (continued)

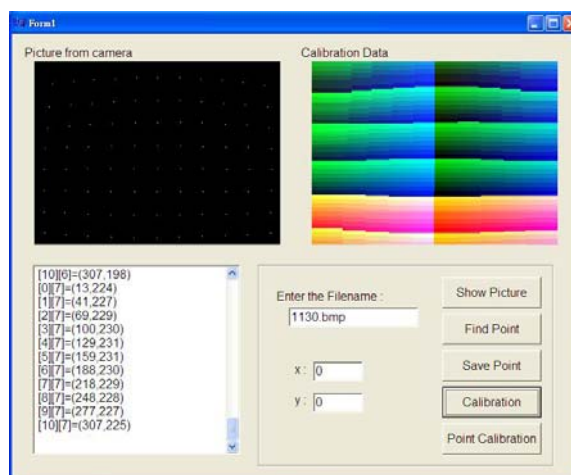


Figure 3.10 The program window of camera calibration.

3.3 Odometer Calibration by Quadratic Functions

3.3.1 Idea of Odometer Calibration

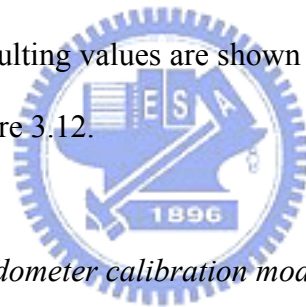
The vehicle moving direction is another important information factor for guiding the vehicle to navigate in indoor environments. The direction information is provided by the odometer of the vehicle. However, the vehicle cannot navigate by using the odometer information only because the incremental mechanical errors might result in imprecise odometer data. Hence, in order to keep the navigation in the path, odometer calibration must be carried out to eliminate the errors.

The vehicle we use in this study has mechanical errors, as mentioned. It deviates gradually to left when it moves forward on a straight line, as observed in our experiments. In this study, we propose a technique to collect the data of the deviation, and analyze the data to build an odometer calibration model. In Section 3.3.2, we will describe the details of the odometer calibration model we adopt which can eliminate the mechanical errors. After finding out the deviation values in every different distance, we want to apply this model to compute the deviations of all distances by a curve fitting scheme which we will discuss in Section 3.3.3. The use of such calibration results is described in Section 3.3.4.

3.3.2 Odometer Calibration Model

Before building an odometer calibration model, we prepare some equipment for our experiment. We use a sticky tape, a measuring tape, an autonomous land vehicle, and a computer. First, we fix the initial position O of the vehicle and mark the position

by pasting a sticky tape on the ground. Second, we fix the initial direction of the vehicle and paste a straight line L along the direction on the ground by using a stick tape. Third, we send commands to drive the vehicle forward on a straight line, and then commands to stop the vehicle. Fourth, we mark the terminal position T of the vehicle by pasting a piece of sticky tape on the ground. Fifth, we find the node P on the straight line L which is the vertical projection of the terminal position T . Sixth, we measure the distance D_1 between O and T which is the move distance of the vehicle, and the distance D_2 between T and P which is the deviation produced by mechanical errors. Seventh, we compute the angle θ of the inverse sine value of D_2/D_1 which is the angle of the deviation. We repeat the steps at least twenty times and let the distance the vehicle moves be different every time. An illustration of the experiment is shown in Figure 3.11. The resulting values are shown in Table 3.1 and the distribution of the results is shown in Figure 3.12.



Algorithm 3.2. *Building an odometer calibration model.*

Input: None.

Output: An odometer calibration model.

Steps:

- Step 1. Fix the initial position O and initial direction line L of the vehicle.
- Step 2. Send commands to let the vehicle move forward and then stop.
- Step 3. Mark the terminal position T of the vehicle.
- Step 4. Find the vertical projection node P of the terminal position T of the vehicle on the straight line L .
- Step 5. Measure the distance D_1 between O and T and the distance D_2 between T and P .

- Step 6. Compute the angle θ of inverse sine value of D_2/D_1 .
- Step 7. Repeat Step 1 to Step 6 at least twenty times and let the distance the vehicle moves every time be different.

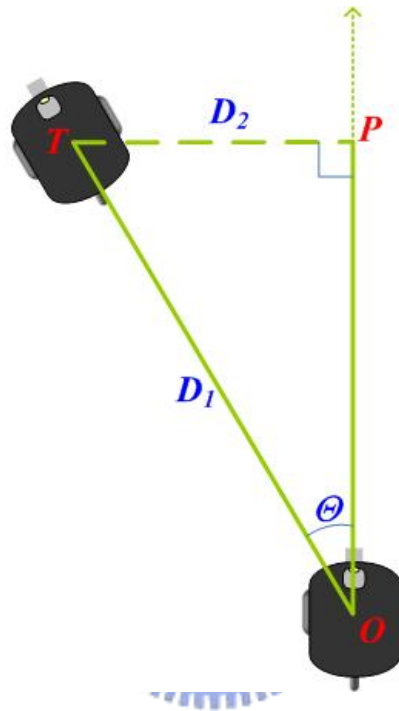


Figure 3.11 An illustration of the experiment.

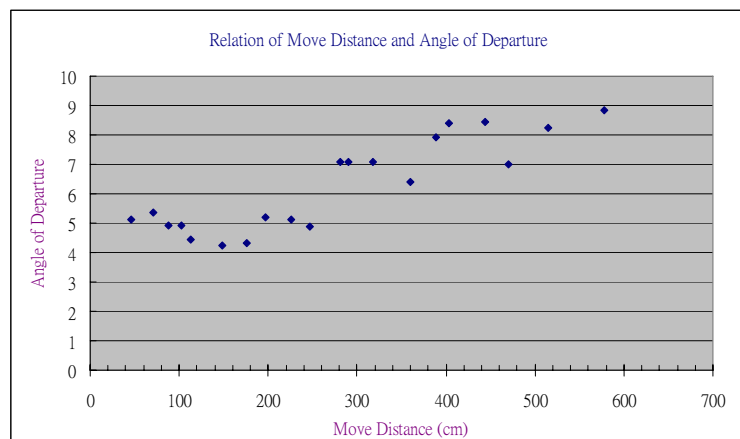


Figure 3.12 The distribution of the angles of the deviations.

Table 3.1 *The results of the experiment of building an odometer calibration model.*

Move Distance (cm)	Distance of Deviation (cm)	Angle of Deviation
45.6	1.3	5.13
70.3	2.1	5.38
87.9	2.4	4.91
102.7	2.8	4.91
113.1	2.8	4.46
148.7	3.5	4.24
175.6	4.2	4.31
197.6	5.7	5.19
225.9	6.4	5.10
246.5	6.7	4.89
281.6	11.1	7.10
290.7	11.4	7.06
317.6	12.5	7.09
359.3	12.8	6.41
389.4	17.1	7.91
403.8	18.8	8.38
443.4	20.8	8.45
469.9	18.3	7.01
515.1	23.6	8.25
577.3	28.3	8.83

3.3.3 Curve Fitting

After measuring the values of the angles of deviations, we found out that the

distribution of data has a trend which may be roughly described as a curve of the second order with respect to the vehicle move distance value. Therefore, we use a least squares error (LSE) fitting method to fit the data with a curve. And then we can use the resulting curve for odometer calibration while patrolling. The method is explained as follows.

First, we generalize a straight line to a curve of the k th-degree polynomial:

$$y = a_0 + a_1 x + \dots + a_k x^k, \quad (3.3)$$

with the residual given by

$$R^2 \equiv \sum_{i=1}^n \left[y_i - (a_0 + a_1 x_i + \dots + a_k x_i^k) \right]^2. \quad (3.4)$$

The partial derivatives of R^2 are

$$\frac{\partial(R^2)}{\partial a_0} = -2 \sum_{i=1}^n \left[y - (a_0 + a_1 x + \dots + a_k x^k) \right] = 0; \quad (3.5)$$

$$\frac{\partial(R^2)}{\partial a_1} = -2 \sum_{i=1}^n \left[y - (a_0 + a_1 x + \dots + a_k x^k) \right] x = 0; \quad (3.6)$$

$$\frac{\partial(R^2)}{\partial a_k} = -2 \sum_{i=1}^n \left[y - (a_0 + a_1 x + \dots + a_k x^k) \right] x^k = 0. \quad (3.7)$$

These lead to the following equations:

$$a_0 n + a_1 \sum_{i=1}^n x_i + \dots + a_k \sum_{i=1}^n x_i^k = \sum_{i=1}^n y_i; \quad (3.8)$$

$$a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + \dots + a_k \sum_{i=1}^n x_i^{k+1} = \sum_{i=1}^n x_i y_i; \quad (3.9)$$

$$\cdot a_0 \sum_{i=1}^n x_i^k + a_1 \sum_{i=1}^n x_i^{k+1} + \dots + a_k \sum_{i=1}^n x_i^{2k} = \sum_{i=1}^n x_i^k y, \quad (3.10)$$

or, in matrix form,

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \dots & \sum_{i=1}^n x_i^k \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \dots & \sum_{i=1}^n x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_i^k & \sum_{i=1}^n x_i^{k+1} & \dots & \sum_{i=1}^n x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \vdots \\ \sum_{i=1}^n x_i^k y \end{bmatrix}. \quad (3.11)$$

This is a Vandermonde matrix. We can also obtain the matrix for a least squares fit by writing:

$$\begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \quad (3.12)$$

Pre-multiplying both sides by the transpose of the first matrix then gives

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \quad (3.13)$$

So, we have

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \dots & \sum_{i=1}^n x_i^k \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \dots & \sum_{i=1}^n x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_i^k & \sum_{i=1}^n x_i^{k+1} & \dots & \sum_{i=1}^n x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \vdots \\ \sum_{i=1}^n x_i^k y \end{bmatrix}. \quad (3.14)$$

As before, given n points, a fitting of them with polynomial coefficients a_0, a_1, \dots ,

a_k gives

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^k \\ 1 & x_2 & \cdots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}. \quad (3.15)$$

In matrix notations, the equation for a polynomial fit is given by

$$Y = XA. \quad (3.16)$$

This can be solved by pre-multiplying by the matrix transpose as follows:

$$X^T Y = X^T X A. \quad (3.17)$$

This matrix equation can be solved numerically, or can be inverted directly if it is well formed, to yield the solution vector:

$$A = (X^T X)^{-1} X^T Y. \quad (3.18)$$

The result of curve-fitting of the previously-mentioned set of angle data of deviations is shown in Eq. (3.19) below and illustrated in Figure 3.13.

$$f(x) = 0.00000476 \times x^2 + 0.00592048 x + 4.16437951. \quad (3.19)$$

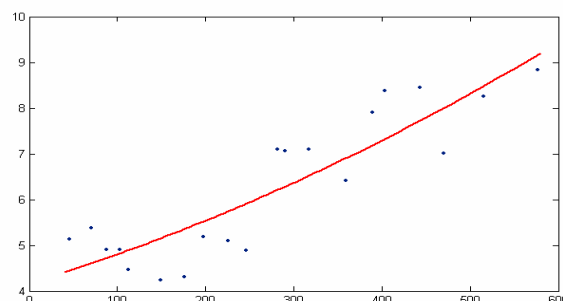


Figure 3.13 The results of line fitting of the angle of deviation.

3.3.4 Navigation by Odometer Calibration

In Section 3.3.2 and Section 3.3.3, we have described the process of building the odometer calibration model and found a quadratic function which can show the relationship between the distance the vehicle moves and the angles of deviations caused by the mechanical errors. In this section, we describe the process about using the quadratic function to calibrate the odometer while navigating the vehicle in indoor environment.

We guide the vehicle most of the time while navigation by the command: “move to front.” But the vehicle always has a leftward deviation while moving forward. To deal with this problem, we use the odometer calibration model to balance the deviation. First, we have to know the distance D we want the vehicle to move forward. Second, we substitute the value D into Eq. (3.19) to get the angle of deviation θ . That means if we command the vehicle to move to D centimeters ahead of the original position, then the vehicle should be instructed to deviate rightward for the angle of θ . Therefore, we issue a command of right turn of angle θ before commanding the vehicle to move forward. In this way, we can balance the deviation. The major steps are shown in Figure 3.14.

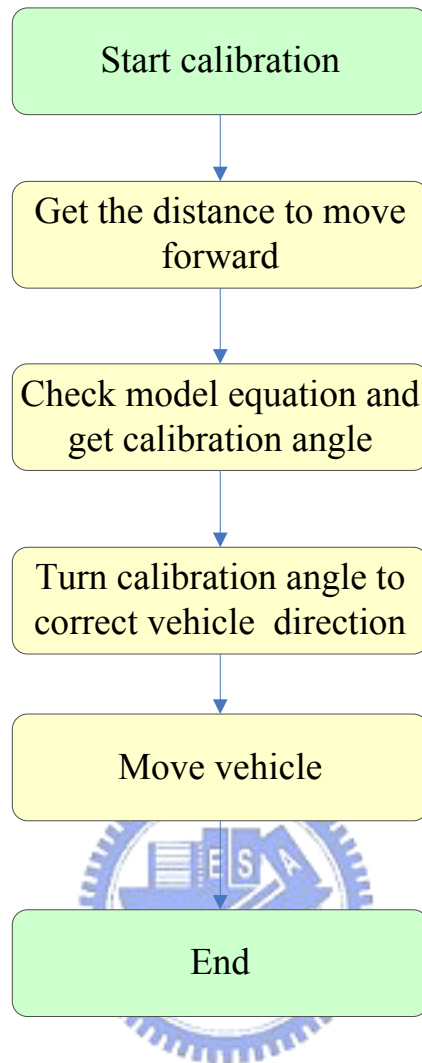


Figure 3.14 An illustration of odometer calibration.

Chapter 4

Learning Procedures

4.1 Introduction

In this chapter, we introduce the details of the proposed learning procedures. Two kinds of environment features are used for learning in this study. The first is path data. In the procedures, the vehicle learns path data while following a person. An adopted method and an improved method of human following are described in Section 4.2. In Section 4.3, we describe how to gather path data when the vehicle follows a person in an indoor environment.

The second feature is 2D object. In the learning procedures, the vehicle learns 2D objects if it detects the situation that the person stops and faces to the left or right side. An adopted method and an improved method of human facing direction detection are described in Section 4.4. For object monitoring, how to detect the 2D object automatically is the key issue, and we propose a method for it in Section 4.5. An illustration of the learning procedures is shown in Figure 4.1.

4.2 Human Following

4.2.1 Review of Adopted Method

Wang and Tsai [1] proposed a person following method which can conduct a

work of following a specified person by motion analysis of human clothes. The system user learned the clothes of the target person manually. The user decided the start point and the boundary box in the image for region growing of the clothes. After learning the image part of the person's clothes, the system will enter the human tracking process. The major steps are shown in Figure 4.2.

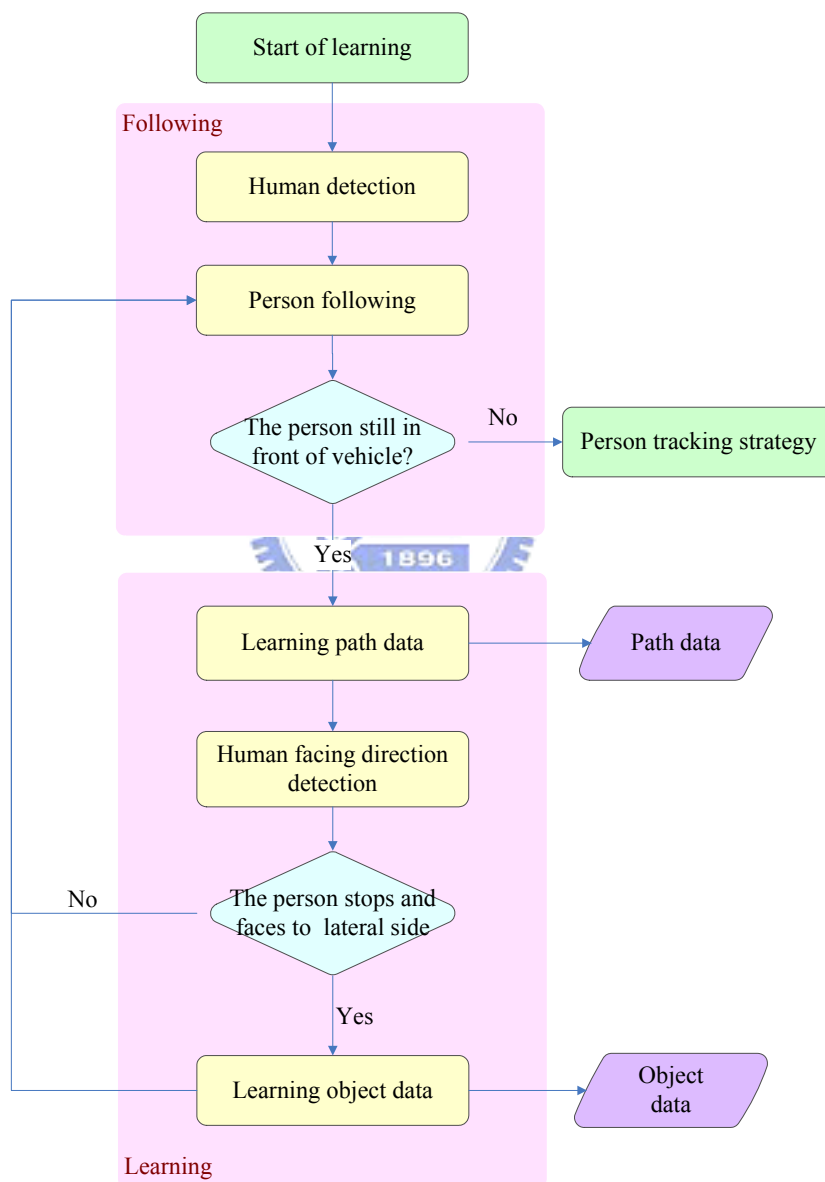


Figure 4.1 An illustration of proposed learning procedures.

To detect the location of the target person by clothes, the system uses a *clothes*

intersection region to predict the direction of the target person. The proposed method only computes the directional variation of the target person. The detail of the proposed clothes region intersection is described in the following algorithm.

Algorithm 4.1. *Clothes region intersection.*

Input: Clothes image $I_{clothes}$, and the initial region $R_{initial}$ which is the target clothes region.

Output: The current region $R_{current}$ of the person's clothes in the image.

Steps:

Step 1. Capture an image $I_{current}$.

Step 2. Subtract $I_{clothes}$ from $I_{current}$ pixel by pixel in $R_{initial}$, and get a new image $I_{intersect}$ as the result, which is the intersection of the two images in the region $R_{intersect}$.

Step 3. Calculate the centroid of $I_{intersect}$ in $R_{intersect}$ and that of $I_{clothes}$ in $R_{initial}$, denoted as $C_{intersect}$ and $C_{initial}$, respectively.

Step 4. Compute $C_{current}$, the centroid of the clothes region location in the current image, by $C_{intersect} = \frac{C_{initial} + C_{current}}{2}$. Also calculate the region $R_{current}$ by $C_{current}$ with its width and height being those of $R_{initial}$, respectively.

Step 5. Let $R_{initial} = R_{current}$, and repeat the steps.

4.2.2 Proposed Improvement of Adopted Process

The vehicle can follow persons by the method mentioned previously in indoor environments with uniform illumination. Unfortunately, the illumination in common indoor environments is not actually uniform, especially in the hallway environment

where there are more than one light and the illumination between two lights is darker than the illumination under either light. A vertical view of the lighting condition under two lights is shown in Figure 4.3.

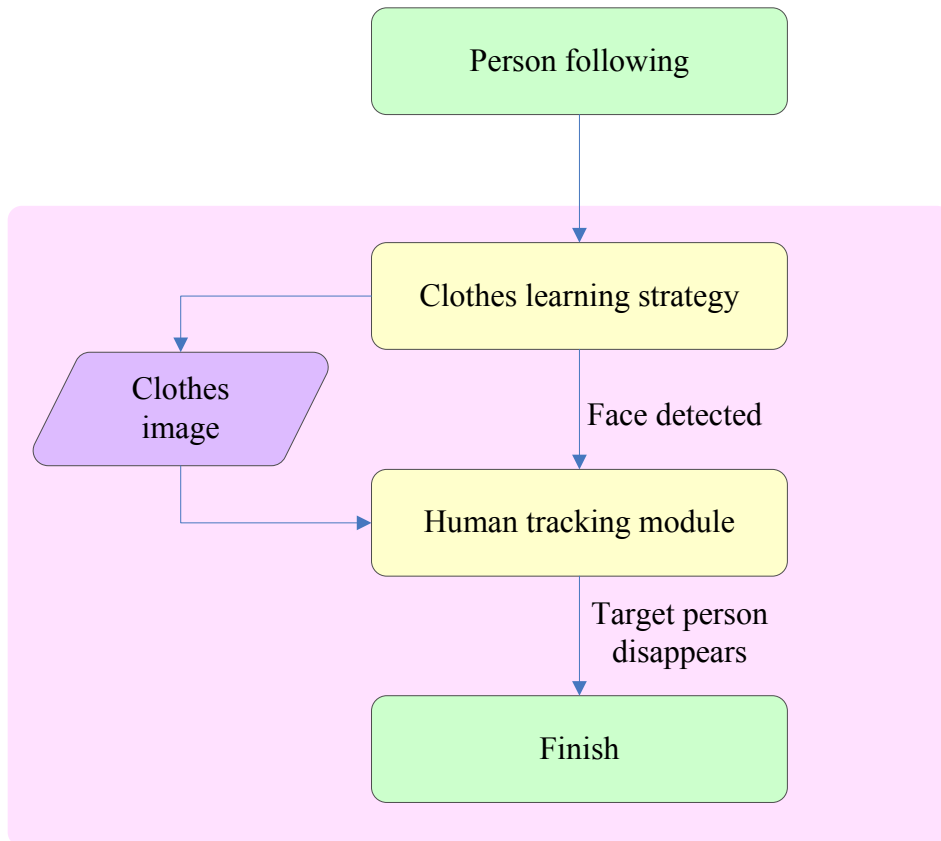


Figure 4.2 The application for person following.

The method mentioned in Section 4.2.1 uses the value of C_b and C_r which are detected at the initial time. However, if the vehicle follows a person and keeps using the value detected at the initial time, then it may lose the tracking of the person because the illumination over the vehicle changes while the vehicle moves in the environment.

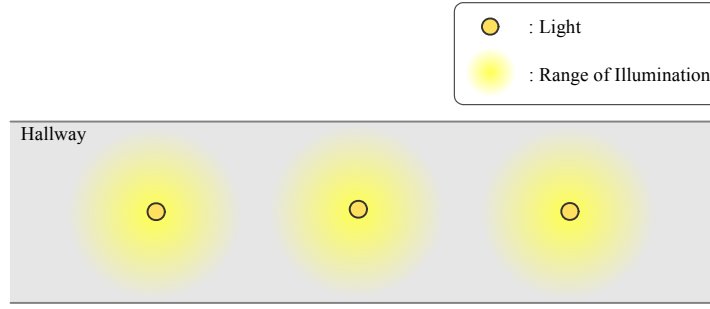


Figure 4.3 The top-view of light and illumination in the hallway.

To deal with the problem, we propose an improvement of an adopted process which dynamically adjusts the values of C_b and C_r for detecting the color of the clothes. First, when we detect the clothes of a person, we collect the pixels p_1, p_2, \dots, p_n of the clothes in the image. Second, we get the values of C_b and C_r of all the pixels p_1, p_2, \dots, p_n , named $C_{b1}, C_{b2}, \dots, C_{bn}$ and $C_{r1}, C_{r2}, \dots, C_{rn}$, respectively. Third, we calculate the average values $C_{b,average}$ and $C_{r,average}$ of all the values of C_{bi} and C_{ri} by Eqs. (4.1) and (4.2) below:

$$C_{b,average} = \frac{1}{n} \sum_{i=1}^n C_{bi}, \quad (4.1)$$

$$C_{r,average} = \frac{1}{n} \sum_{i=1}^n C_{ri}. \quad (4.2)$$

Fourth, we detect persons by using the new values $C_{b,average}$ and $C_{r,average}$. Instead of using pre-selected fixed values, the vehicle uses *dynamic* values of $C_{b,average}$ and $C_{r,average}$ in each cycle to detect and follow the person and overcomes the problem of luminance changes under different lighting conditions. The major steps are shown in Figure 4.4.

4.2.3 Person Tracking Strategy

When the person being followed turns fast in front of the vehicle, it is difficult for the vehicle to keep track of the person. Chen and Tsai [2] proposed a method of recording the disappearing direction of a fast-moving person by an arm equipped on the vehicle. Our system records the direction of disappearance by using the *capability of panning* of a PTZ camera.

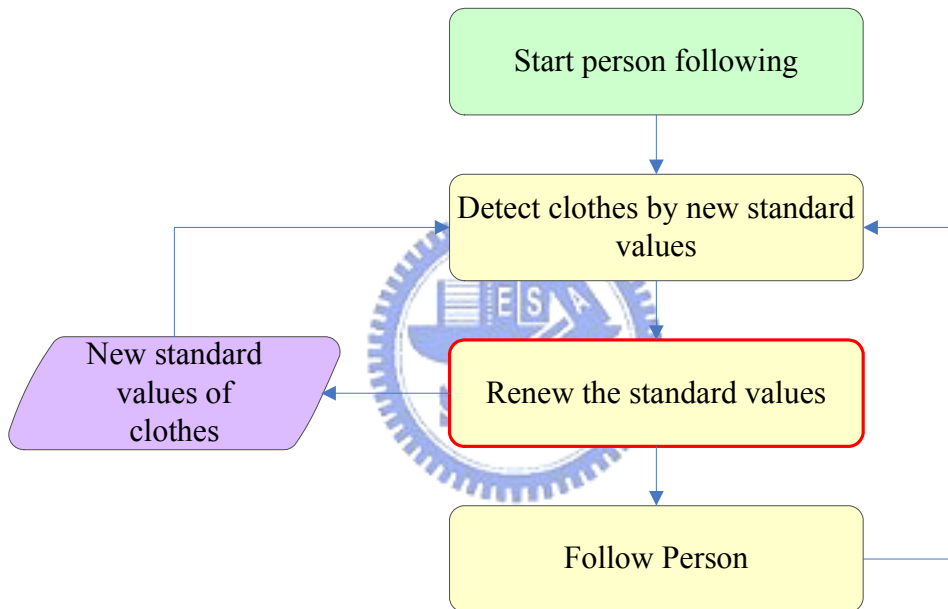


Figure 4.4 An illustration of improvement of adopted process.

When the person turns fast in front of the vehicle, we can find out his/her turning direction, *left* or *right*, and guide the vehicle to turn for a large angle, θ_{find} , to the direction to find the disappearing person. If the disappearing person appears in the view of the camera after the vehicle turning, then the vehicle will follow the person again and continually. But, if the vehicle cannot find the disappearing person after turning, then it turns again until the sum of the multiple turning angles is equal to

180°. When the vehicle turns 180° (half of a circle), we can be sure that the person disappears in the direction which the system recorded. Then the system will end the process of learning. We save the positions of this kind of *loss-of-tracking* point, where the vehicle turns the angle of θ_{find} , as the turning point data in this study and use them in the patrolling process. An illustration of the previously-described person tracking strategy is shown in Figure 4.5, and an example of results is shown in Figure 4.6.

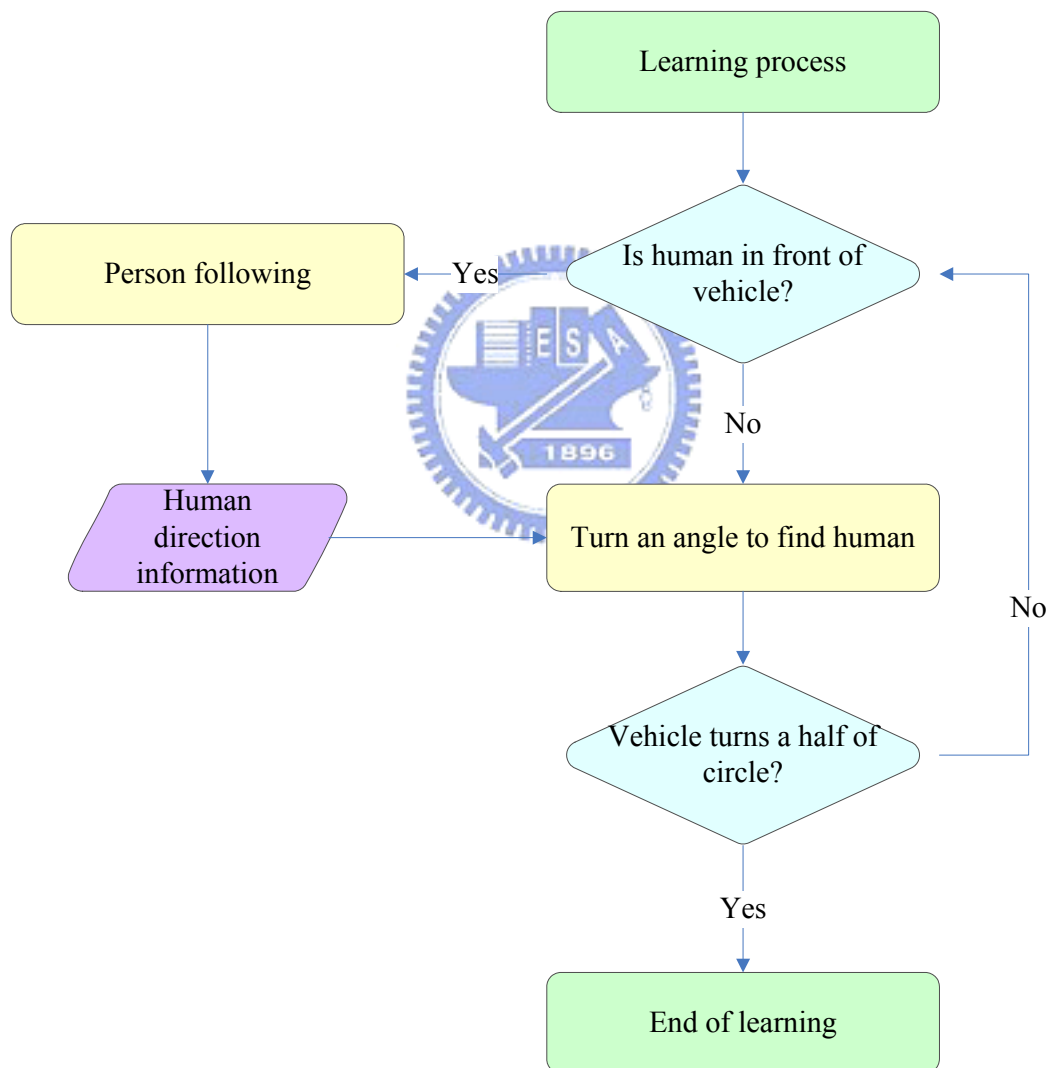


Figure 4.5 An illustration of person tracking strategy.



(a)



(b)



(c)



(d)



(e)

Figure 4.6 An example of person tracking results. (a) A vehicle is following a person. (b) The person turns fast to right. (c) The vehicle turns an angle θ_{find} to find the person. (d) The vehicle finds the person. (e) The vehicle follows the person again.

4.3 Learning of Path Data

When the vehicle is following a person, the odometer provides continuously the current position data of the vehicle in the global space with respect to the start point of the current navigation session, and the ultrasonic sensors provide continuously the current distance data with respect to the surrounding objects or walls. The position data, which are provided by the odometer, consist of the vehicle coordinates (x, y) in the vehicle coordinates system. And the distance data, which are provided by the ultrasonic sensors, consist of the left and right side distances (d_l, d_r) from the vehicle to the upholstery of concern in the environment. We record both the position and the distance data as the path data in this study.

We save once every second the coordinates (x, y) and the distances (d_l, d_r) integrally as a node N_i . Each node is labeled with a serial number. These nodes form a graph of the learned path. After learning, we have a set of nodes, denoted as N_{path} . We will use the path data in the process of path planning. Additionally, we can use the data to draw a draft of the map of the environment. Two drafts of maps of two different environments are shown in Figure 4.7.



Figure 4.7 Drafts of maps. (a) Map of an open indoor environment. (b) Map of a hallway environment

4.4 Detection of Human Facing Direction

4.4.1 Review of Adopted Process

We can know the direction of the monitored object by detecting the human facing direction. Chen and Tsai [2] proposed a method for human facing direction detection, which can decide what direction a person is facing to, based on the use of the change of the ratio of the width of the person's clothes to the height. When the person turns, this ratio becomes smaller because the shape of the clothes extracted in the image becomes thinner. Then the distribution of the person's hair and skin is used to judge whether the person turns to the right or to the left. For this, the center of the face is found out first and the colors of the pixels in a horizontal line which passes this center point are collected to compute the color distribution of the hair and skin in the person's face.

If the person is facing to the vehicle, then most pixels on the horizontal scan line are of the skin color; if the vehicle is at the back of the person, then most pixels on the line are of the hair color. When the person turns to the right, the hair is at the left side of the face and the skin is at the right side, so most pixels at the left side of the scan line are of the hair color and most pixels at the right side of the scan line are of the face color, and vice versa. An illustration of human facing direction detection is shown in Figure 4.8. An illustration of the facing direction of the person is shown in Figure 4.9.

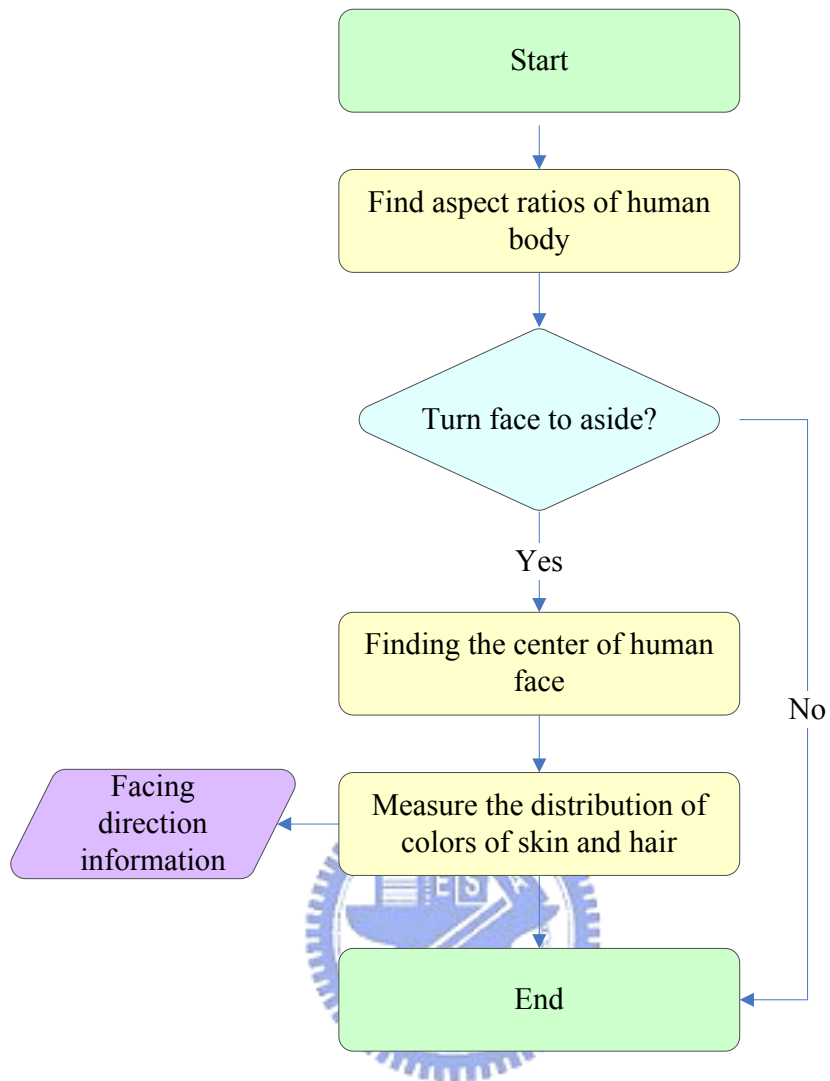


Figure 4.8 An illustration of detection of a person's facing direction.

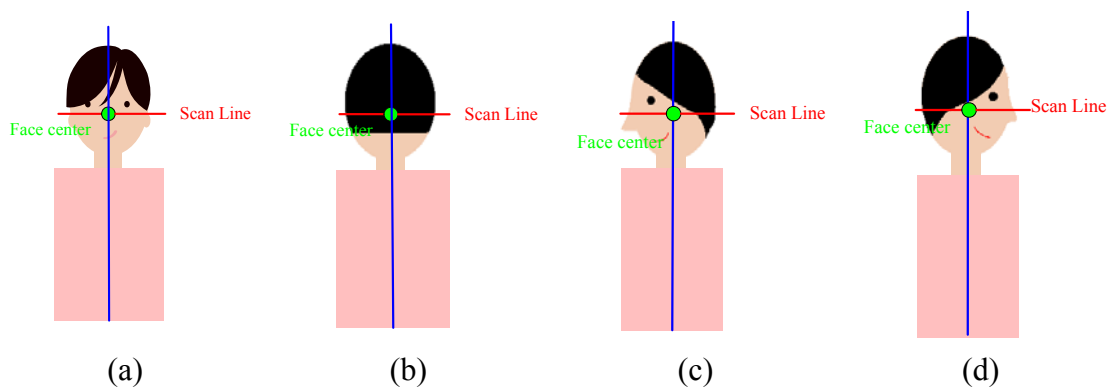


Figure 4.9 An illustration of the facing direction of the person. (a) Front. (b) Back. (c) Left. (d) Right.

4.4.2 Proposed Improvement of Adopted Process

The proposed method mentioned in Section 4.4.1 detects the human facing direction by using a scan line. However, it gets a false result easily, because the scan line is thin and the numbers of pixels on the scan line are little. For this situation, instead of using the scan line, our system uses a scan rectangle to increase pixels for detection and enhance the accuracy of detection. The details of the improved method are described in the following algorithm and an illustration is shown in Figure 4.10.

Algorithm 4.2. *Finding the facing direction of a person.*

Input: The current image I_c , the clothes center $C_a (i_c, j_c)$ and the four corner points

$P_{TopLeft} (i_{tl}, j_{tl})$, $P_{TopRight} (i_{tr}, j_{tr})$, $P_{BottomLeft} (i_{bl}, j_{bl})$, and $P_{BottomRight} (i_{br}, j_{br})$ of the clothes region, the length of the person's face L_{face} , a region of the color of blackness, $Black$, the skin color region $Skin$, a minimum threshold T_2 and a maximum T_3 threshold, a width value L_{wide} of the scan rectangle.

Output: The facing direction of the person, $Direction$.

Steps:

Step 1. Scan the column of the image I_c , which contains the pixel $C_a (i_c, j_c)$ to find the first pixel $Hair (i_{hair}, j_{hair})$ with the color of the hair, $Black$.

Step 2. Find the center of the face $C_{face} (i_{face}, j_{face})$ by the following way:

$$i_{face} = i_c; \quad (4.3)$$

$$j_{face} = j_{hair} + \frac{L_{face}}{2}. \quad (4.4)$$

Step 3. Because the range of the width of the face will not exceed the clothes, limit

the horizontal search region to be from $P_{TopLeft}(i)$ to $P_{TopRight}(i)$.

Step 4. Find the value R_{Top} and R_{Bottom} by the following way, and limit the vertical search region to be from R_{Top} to R_{Bottom} :

$$R_{Top} = j_{face} - \frac{L_{wide}}{2}; \quad (4.5)$$

$$R_{Bottom} = j_{face} + \frac{L_{wide}}{2}. \quad (4.6)$$

Step 5. Measure the values of C_b and C_r of the pixels inside the left half of the scan rectangle position of the horizontal direction from i_{tl} to i_{face} and the vertical direction from R_{Top} to R_{Bottom} . If the values fall into the region *Black*, set the number NL_{black} of black color as

$$NL_{Black} = NL_{Black} + 1. \quad (4.7)$$



If the values fall into the region *Skin*, set the number NL_{skin} of skin color as

$$NL_{skin} = NL_{skin} + 1. \quad (4.8)$$

Step 6. Measure the values of C_b and C_r of the pixel inside the right half of the scan rectangle from the horizontal position i_{face} to i_{tr} and from the vertical position R_{Top} to R_{Bottom} . If the values fall into the region *Black*, set the number NR_{black} of black color as

$$NR_{Black} = NL_{Black} + 1. \quad (4.9)$$

If the value falls into the region *Skin*, set the number NL_{skin} of skin color as

$$NR_{skin} = NR_{skink} + 1. \quad (4.10)$$

Step 7. Check the sizes of the distributions of the colors of the skin and the hair as follows:

$$NL_{black} + NR_{black} < T_2; \quad (4.11)$$

$$NL_{skin} + NR_{skin} > T_3; \quad (4.12)$$

$$NL_{black} > NR_{black}; \quad (4.13)$$

$$NL_{skin} < NR_{Skin}. \quad (4.14)$$

If Inequalities (4.11) and (4.12) are satisfied, set *Direction* as “Front”;
 else,
 if Inequalities (4.13) and (4.14) are satisfied, set *Direction* as “Right”;
 else, as “Left”.

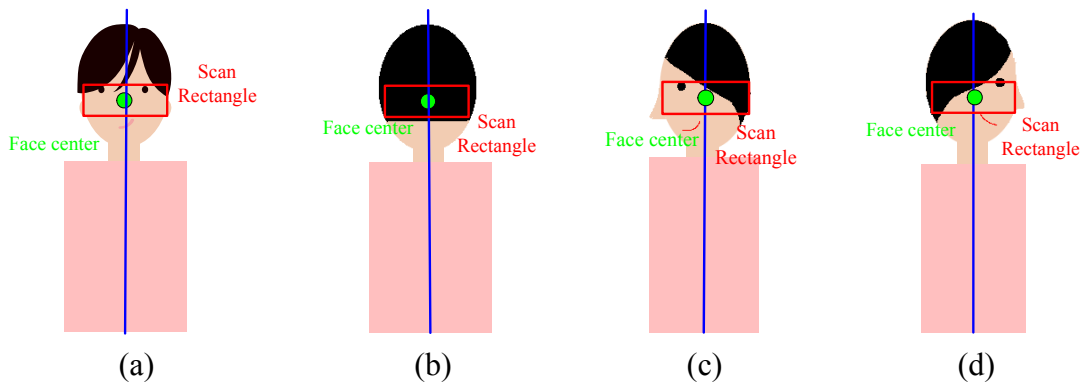


Figure 4.10 An illustration of the scan rectangle for human facing direction detection.
 (a) Front. (b) Back. (c) Left. (d) Right.

4.5 Learning of 2D Objects

4.5.1 Review of adopted algorithm for learning of objects

In order to learn concerned objects, Chen and Tsai [9] designed a user interface to help users specify the object which they want to monitor. While the user controls the vehicle to the front of the object to be monitored, they can move the PTZ camera toward the object. Then, they can select the object in the image by the use of the mouse connected to the computer to drag a rectangle as an interesting region to cover the object which appears in the image, as shown in Figure 4.11(a). After that, the system applies the simplified SIFT algorithm [9] to obtain the feature set of the interesting region.

During the learning phase, if the user gives a horizontal line, which is parallel to the floor plane in the 3D global coordinate system in the image as shown in Figure 4.11(b), the system can acquire the same line found in the image taken in the navigation phase by applying this affine transformation. Then, by some analytic mathematics analysis on this horizontal line found in the image taken in the navigation phase, the proposed system could obtain the relative position $C(x_r, y_r)$ and the relative angle θ_r of the vehicle in the world coordinate system with respect to the monitored object, as shown in Figure 4.12.

Then, the user saves the vehicle location, the PTZ position, the feature set, and the interesting region into the storage of the computer. The last thing is to save the calibration information data including the start point (u_l, v_l) of the horizontal line, the coefficients b and c in the equation $u + bv + c = 0$ of the horizontal line, the relative

position (x_r, y_r) , and the relative angle θ_r for use in adjusting the vehicle location in the navigation phase. The detailed learning process is described in the following.



Figure 4.11 (a) A red rectangle including the monitored object as an interesting region. (b) A user interface to specify the horizontal line.

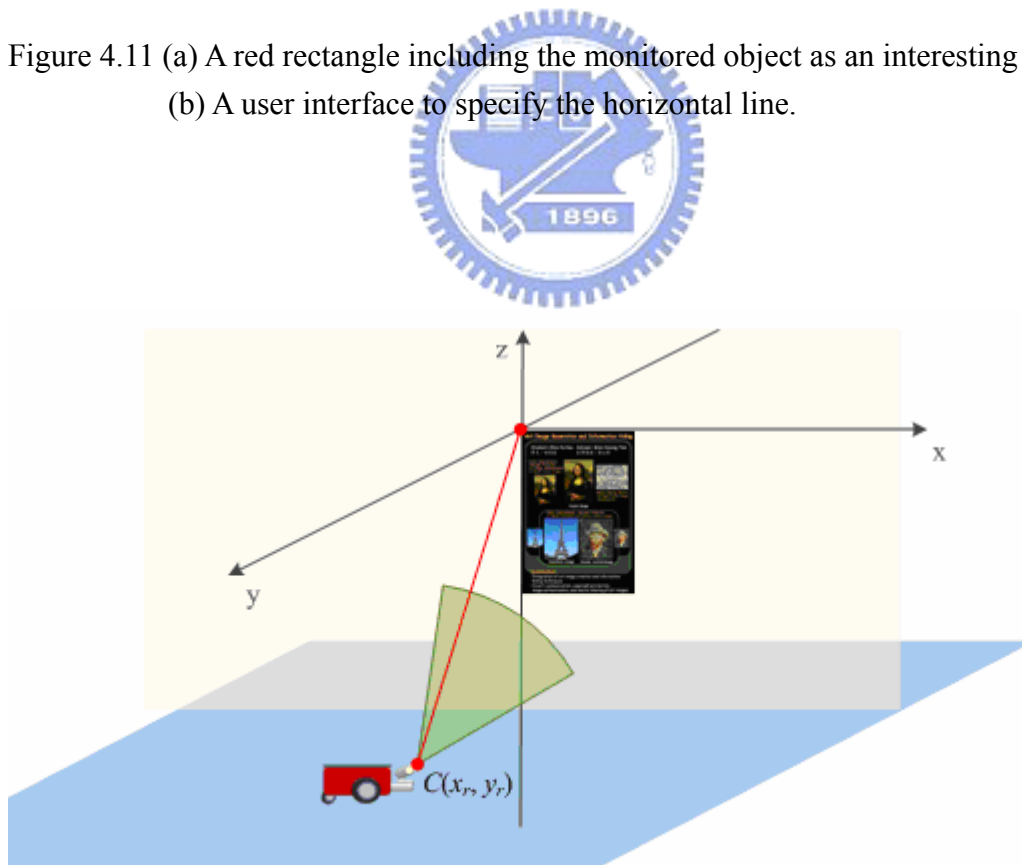


Figure 4.12 The relative position $C(x_r, y_r)$ of the vehicle with respect to the start point in the world coordinate system.

Algorithm 4.3. *Learning of a monitored object.*

Input: The position P of a monitored object.

Output: A calibration information data.

Steps:

- Step 1. Drive the vehicle to the monitored object position P .
- Step 2. Move the PTZ camera toward the object and take an image I .
- Step 3. Drag a rectangle on the image I as an interesting region.
- Step 4. Apply the simplified SIFT algorithm [9] on the interesting region to extract the feature set.
- Step 5. Select the start point (u_1, v_1) and the end point (u_2, v_2) of the horizontal line in the image I which is parallel to the floor plane in the 3D global coordinate system.
- Step 6. Compute the coefficients b and c by solving the equation $u + bv + c = 0$ with (u_1, v_1) and (u_2, v_2) .
- Step 7. Apply the location estimation algorithm proposed in [9] to find the relative position $C(x_r, y_r)$ and the relative angle θ_r with respect to the start point in the 3D global coordinate system as an origin.
- Step 8. Save the vehicle location, the PTZ position, and the feature set of the interesting region as monitored object information data.
- Step 9. Save the start point (u_1, v_1) , the coefficients b and c , the relative position (x_r, y_r) , and the relative angle θ_r as calibration information data for a monitored object.
- Step 10. Save the interesting region in the image.

4.5.2 Learning 2D Objects Automatically

The proposed method of learning 2D objects described in Section 4.5.1 need to be performed manually. In our system, the vehicle learns 2D objects and follows a person in the mean time. It is unreasonable to let the user controls the computer and leads the vehicle at the same time. Hence, we propose a method which can learn 2D objects automatically.

Because of the vehicle always detects the facing direction of the user, if there is a picture on a wall and we hope that the vehicle could remember the picture, the user just need to stand in front of the picture and face to it for a while, and then the vehicle will go to the position where the user stands and pan the angle of camera to the same direction as the user. The vehicle remembers the view of the camera frame as image I . The system finds the interesting region of the 2D object and a horizontal line L , which is parallel to the floor plane in the 3D global coordinate system in image I . We save the positions of this kind of nodes where the vehicle learns 2D objects, as monitoring point data in this study and it will be used in the patrolling process.

The system uses the method of the simplified SIFT algorithm [9] to obtain the feature set of the picture and applies the location estimation algorithm proposed in [9] to find the relative position $C(x_r, y_r)$ and the relative angle θ_r with respect to the start point of the horizontal line L in the 3D global coordinate system as an origin. Finally, we save the information into the storage of the computer. The detail process is described in the following algorithm, and a flowchart is illustrated in Figure 4.13.

Algorithm 4.4. *Learning of a monitored object automatically.*

Input: A vehicle and a user.

Output: Monitored object information data.

Step 1. Follow the user.

Step 2. Detect the human facing direction, if the user stops and turns to left or right for a while, go to Step 3, else go back to Step 1.

Step 3. Go to the position of the user and pan the camera to the picture.

Step 4. Remember the view of camera frame as image I .

Step 5. Find the interesting region of the 2D object and a horizontal line L , which is parallel to the floor in the image I .

Step 6. Calculate the position of the vehicle with respect to the horizontal line L .

Step 7. Calculate the feature set of the 2D object.

Step 8. Save information.



4.5.3 Finding Regions of 2D Objects

There are two crucial techniques in the method of learning 2D objects automatically. The first is finding the interesting region of the 2D object in the image I and the other is finding the horizontal line L , which is parallel to the floor plane in the 3D global coordinate system in image I . For finding the interesting region, we deal with image I by using thresholding and region growing methods. The detail process is described in the following algorithm. Additionally, we will describe the method of finding the horizontal line L automatically in Section 4.5.4.

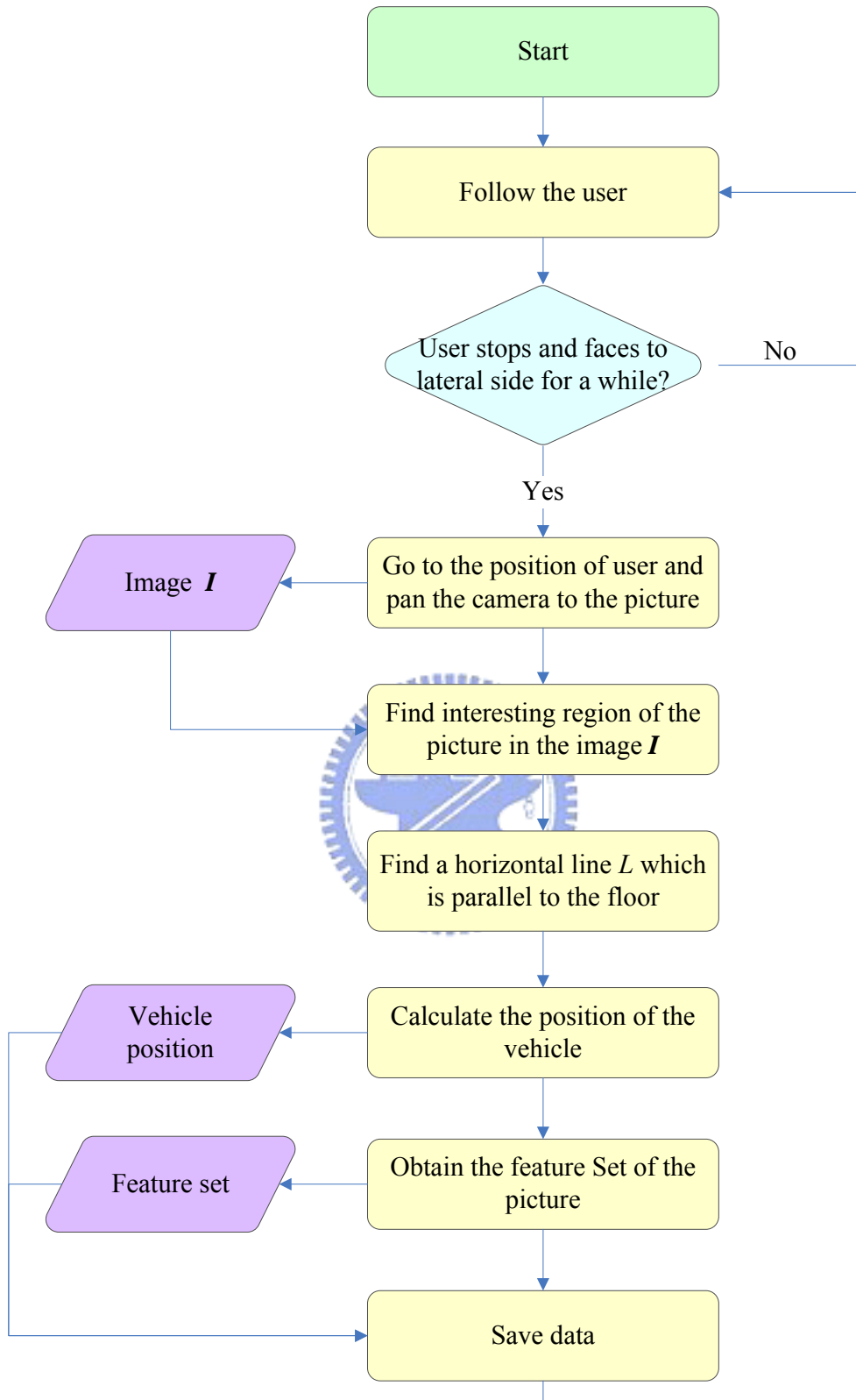


Figure 4.13 An illustration of learning of a monitored object automatically.

Algorithm 4.5. *Finding the interesting region in the image.*

Input: An image I with a monitored object, and the gray version I_g of I .

Output: Four corner points $P_{TopLeft}(i_l, j_t)$, $P_{TopRight}(i_r, j_t)$, $P_{BottomLeft}(i_l, j_b)$, and $P_{BottomRight}(i_r, j_b)$ of an interesting region, and the horizontal line L .

Step 1. Calculate the threshold value T of the gray value which can differentiate background and foreground of I .

Step 2. Reset the gray value g_{pi} of every pixel p_i in the image I_g in the following way:

If $g_{pi} > T$ is satisfied, set g_{pi} as a foreground point;

else,

if $g_{pi} \leq T$ is satisfied, set g_{pi} as a background point.

Step 3. Conduct region growing from the center of I_g and get the top and bottom vertical coordinate values G_T and G_B , the leftmost and rightmost horizontal coordinate values G_L and G_R .

Step 4. Set the coordinate values i_l , i_r , j_t and j_b for corner points $P_{TopLeft}(i_l, j_t)$, $P_{TopRight}(i_r, j_t)$, $P_{BottomLeft}(i_l, j_b)$, and $P_{BottomRight}(i_r, j_b)$ of the interesting region by the following way

$$i_l = G_L ; \quad (4.15)$$

$$i_r = G_R ; \quad (4.16)$$

$$j_t = G_T ; \quad (4.17)$$

$$j_b = G_B . \quad (4.18)$$

4.5.4 Finding the Horizontal Line Automatically

If we want to find the horizontal line, then we need to find the set of edge points by *edge detection* in the image first. There are many different operators which can find edges in an image, for example, the *Laplacian operator*, *Marr-Hildreth operator* and *Canny operator*, etc. In our system, we detect edges by using the *Sobel operator* as shown in Figure 4.14. Let T be the threshold value of the Sobel operator.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)



Figure 4.14 Sobel Operator (a) x direction (b) y direction.

We apply the Sobel operator on a pixel p_i , and let r_{p_i} be the resulting value. If Inequalities (4.19) below is satisfied, then we mark the pixel p_i as an edge point.

$$r_{p_i} > T \quad (4.19)$$

After checking every pixel p_i in the interesting region which was obtained from Section 4.5.3, we can get a set of edge points, EP .

Now we can find the line by using the *Hough Transform*. It is a method of *line detection* by a voting technique and the relationship between the parameter space and the normal distance-normal angle space is as shown in Figure 4.15.

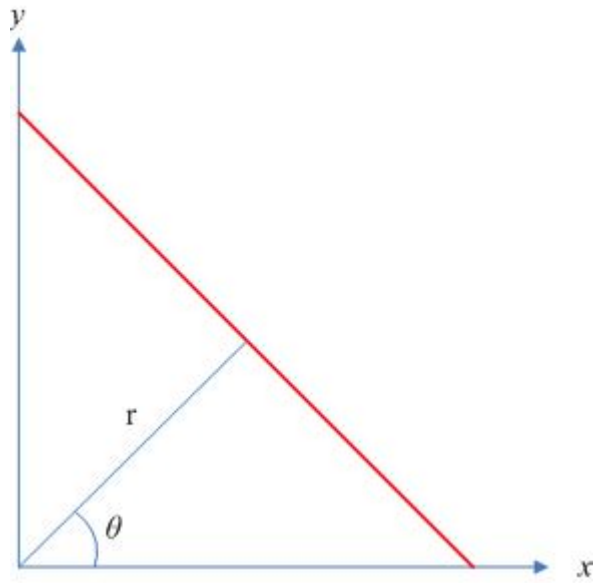


Figure 4.15 The relationship with x - y space and r - θ space.

At first, we divide the range of angles, $[0, \pi]$, in to n part, $\theta_0, \theta_1, \dots, \theta_n$, and calculate the length r_{region} of a diagonal of the interesting region. We prepare next an accumulation array A , whose size is $n \times r_{region}$, and set the value of every cell to zero, as shown in Figure 4.16.

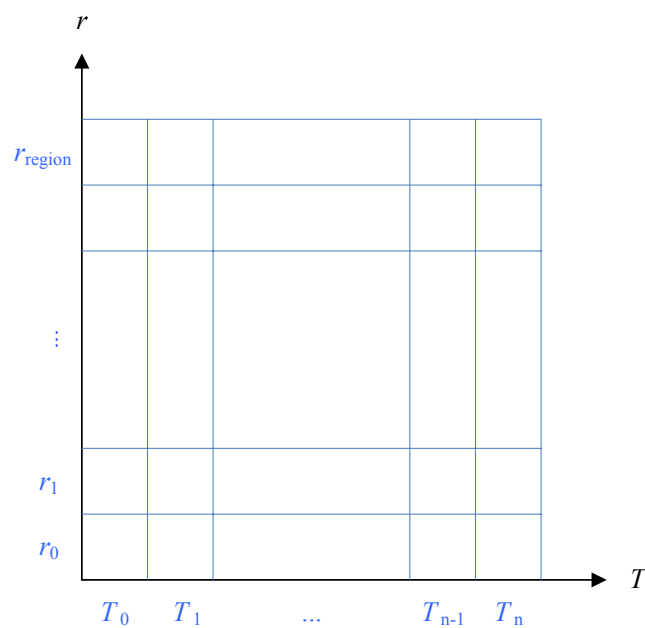


Figure 4.16 The accumulation array A of the *Hough Transform*.

We substitute a point (x, y) in the edge point set EP and θ_i into Eq. (4.20) below to get a value r and put a vote into the cell (r, θ_i) :

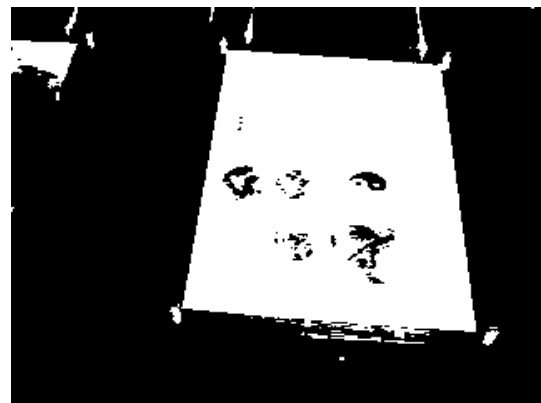
$$r = x \cos \theta_i + y \sin \theta_i. \quad (4.20)$$

We repeat this step of voting in the cells for every edge point and every θ_i . After voting, we find the cell (γ, θ) with the maximum number of votes to obtain a line which has the intercept γ and the angle θ in the normal distance-normal angle space.

There is a restriction in our method, that is, the heights of monitored objects must be of fixed values known in advance. By the progressive method as described above, we can find the interesting region of the concerned 2D object in the image I and the horizontal line L , which is parallel to the floor plane in the 3D global coordinate system in image I , as shown by the example in Figure 4.17 and Figure 4.18.



(a)



(b)

Figure 4.17 An example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line.



(c)

(d)

Figure 4.17 An example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line. (continued)



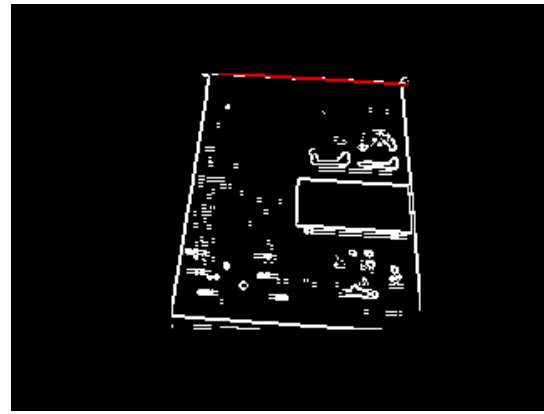
(a)

(b)

Figure 4.18 Another example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line.



(c)



(d)

Figure 4.18 Another example of detection of information of monitored object by the progressive method. (a) An image with a monitored object. (b) The thresholding result. (c) The interesting region. (d) The horizontal line. (continued)



Chapter 5

Path Planning by Minimizing Mean Square Errors Using Ultrasonic Signals

5.1 Introduction

After learning the path and the monitored object data, the system will refine the path data before the vehicle navigates to the start point. The path data are composed of many path nodes. Two extreme examples of path planning results are shown in Figure 5.1(a) and Figure 5.1(b). The First example is a path for backing to the start point directly which is very simple and crude, and there might have some obstacles on the path. The second example is a path stepping on every node precisely. It may not seem smart because the path could be rough or tortuous. Hence, we desire the planned path could be smooth but not lose the original trend, as the illustration shown in Figure 5.1(c). We describe the details of the proposed path planning process in Section 5.2.

The system performs the path planning process by using the MSE criterion. In the MSE criterion, the system needs to set a threshold value which is used to control the precision of the path. Hence, the choice of the threshold value is very important. Our system uses a technique which can choose the threshold value by adapting to different environments. We describe the detail of the proposed process in Section 5.3. In Section 5.4, we will describe the details of the patrolling technique of our system.

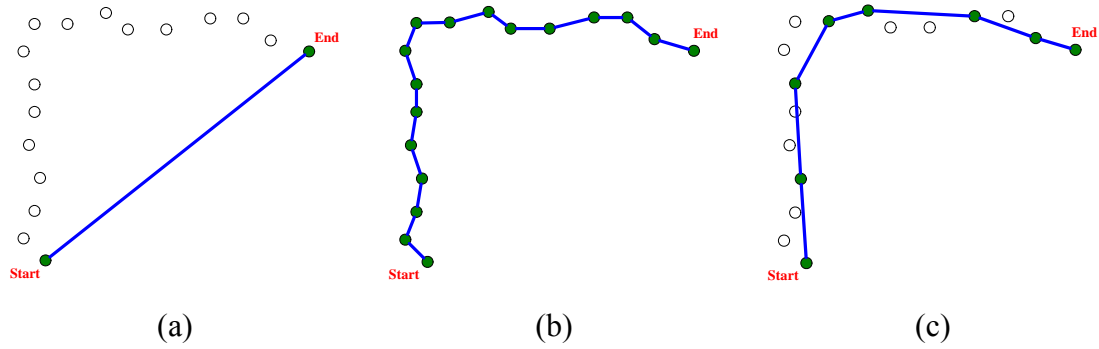


Figure 5.1 Illustrations of path planning. (a) Backing to the start point directly. (b) Stepping on every path node precisely. (c) The planned path we desire.

5.2 Path Planning by MSE criterion

The path data are composed of path points. As a start, the system finds out n path points where we learned monitored objects, and separates the path into $n+1$ segments by the n path points. Then we construct an empty queue Q , and push the $n+1$ segments into Q .

The system pops out a segment of the path, and uses the *line fitting* technique which is described in Section 3.3.3, to find a linear equation of a line L in two variables, x and y , like the formula of Eq. (5.1) below, where A , B and C are constants:

$$L: Ax + By + C = 0. \quad (5.1)$$

We calculate the distance values d_i from every path point $P_i(x_i, y_i)$ in this segment to the line L by

$$d_i = \frac{|Ax_i + By_i + C|}{\sqrt{A^2 + B^2}}. \quad (5.2)$$

The value of the induced square error E of the line L can be calculated by

$$E = \frac{1}{m} \sum_{i=1}^m (d_i)^2 . \quad (5.3)$$

The system uses a threshold value T_{mse} as the upper bound of the square error value. If the error E is smaller than T_{mse} , then the system considers the currently-processed segment of the path smooth enough. But if the error E is larger than T_{mse} , then the system considers the segment not smooth enough, and we need to do more processing for this path.

It is proposed in this study to choose a cut point from the path points P_1, P_2, \dots, P_m of the segment of the path which is not smooth enough. If we cut on the point P_j , then we need to find the line L_{1j} and L_{jm} by using the line fitting method again and calculate the values of the two mean square errors E_{1j} and E_{jm} separately. Then, the square error of cutting on the point P_j is

$$E_j = E_{1j} + E_{jm} . \quad (5.4)$$

After cutting the path $m-2$ times on P_2, \dots, P_{m-1} , we can get $m-2$ mean square errors E_2, E_3, \dots, E_{m-1} . We then find the minimum value E_k from the $m-2$ error values, with E_k meaning that if we cut the path on P_k , then we can get the least error value. We finally push the two shorter paths P_1, P_2, \dots, P_k and $P_{k+1}, P_{k+2}, \dots, P_m$ into the queue Q .

The system executes the above process repeatedly, until the queue is empty, and at that time the original path has been cut into many segments which are smooth enough. Finally, we save every start point of these segments of the path as the navigation path data. An illustration of the path planning process is shown in Figure 5.2.

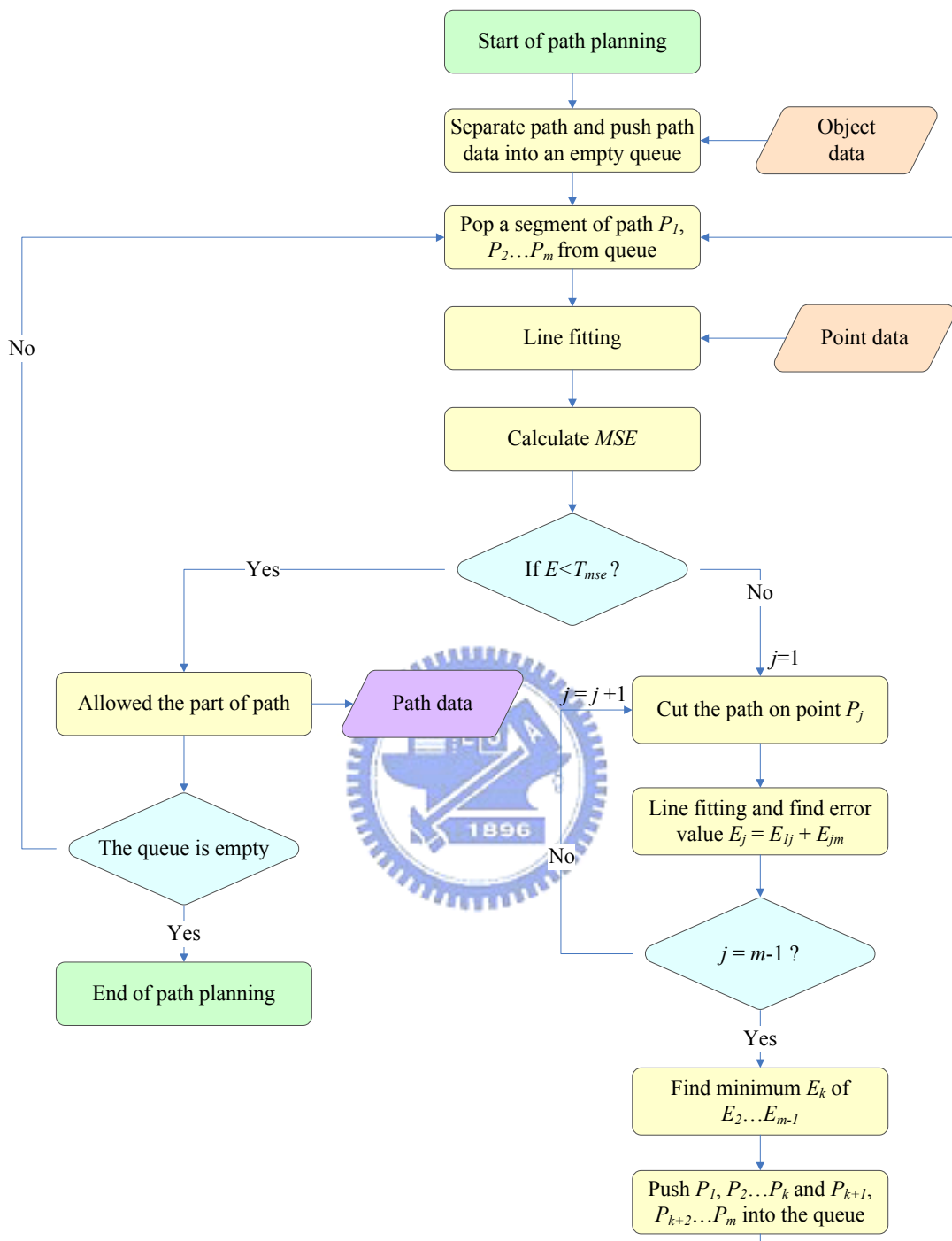


Figure 5.2 Illustration of proposed path planning.

5.3 Dynamic Path Decomposition

The threshold value of the MSE criterion is used to tolerate the value of the error. The larger the threshold, the cruder the path decomposed. There is no advantage if the threshold is very large or very small. The choice of the threshold should adapt to the environment of the path. Moreover, the environment could be different at different parts of a path, and each different part of the path should be decomposed by a different degree of precision. We propose a technique here which chooses the threshold value dynamically.

Except the coordinates of path points, the distances are also part of the information of path data which are learned in the learning process by using ultrasonic sensors. Our system knows the breadth of the environment of a path by analyzing the distance data collected by the ultrasonic sensors. In the learning process, the system learned the left and right distances ($d_{l,i}$, $d_{r,i}$) of every path point P_i with respect to the obstacle or the wall around the point. We simplify such distance data to get a single distance value D_i of every path point by

$$D_i = \min(d_{l,i}, d_{r,i}). \quad (5.5)$$

Considering the correlation of the connected segments of the path, our system calculates the threshold value of one segment by using the distance values of not only the path points in this segment but also five points before and five points behind this segment. Additionally, the composite the mean square error for this segment is taken to be the average of the squares of the error distances, as Eq. (5.3), and the threshold value is related to the mean square error. So we derive the equation of Eq. (5.6) below to calculate the threshold value of a segment with n points;

$$Threshold = \frac{1}{k(5n+10)} \left(\sum_{i=-5}^{-1} D_i^2 + \sum_{i=1}^n D_i^2 + \sum_{i=n+1}^{n+5} D_i^2 \right), \quad (5.6)$$

where k is a constant.

We decompose the path dynamically by recalculating the above threshold value in every cycle of a navigation session. An illustration of the proposed dynamic path decomposition scheme is shown in Figure 5.3.

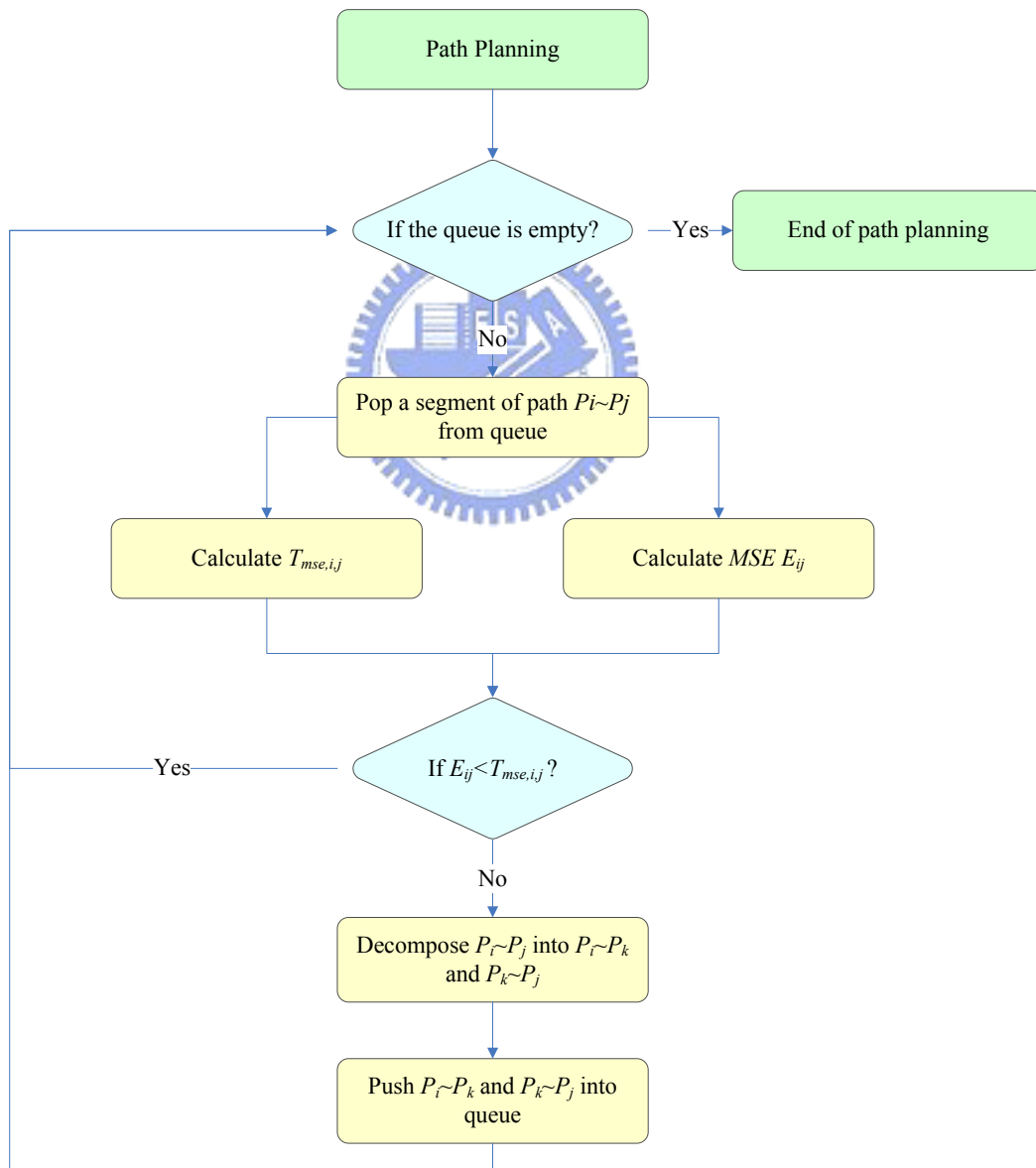


Figure 5.3 An illustration of proposed dynamic path decomposition.

5.4 Vehicle Navigation by 2D Object Image Matching

After planning the path, the last process of the system is vehicle patrolling from the end to the start position of the path obtained from the path planning process. In a patrolling session, the vehicle navigates along a learned path by visiting each path point consecutively. The learned navigation path, which consists of a set N_{path} of points and we specify the last point of N_{path} to be the initial position of the navigation path. Then, the vehicle reads the next node data $N_{i+1}(x_{i+1}, y_{i+1})$ and computes a turning angle and a moving distance by Eqs. (5.7) and (5.8) in the following algorithm, for the vehicle to move to the next position $N_{i+1}(x_{i+1}, y_{i+1})$, as shown in Figure 5.4.

Algorithm 5.1. *Process of vehicle guidance by a learned path.*

Input: A set N_{path} of nodes.

Output: The vehicle navigates to the start point in the learning process.

Steps:

- Step 1. Start vehicle navigation from the starting node N_0 in N_{path} .
- Step 2. Let the current position be $N_i(x_i, y_i)$, and the direction read from the odometer be θ_{odo} .
- Step 3. Scan N_{path} to read the next node $N_{i+1}(x_{i+1}, y_{i+1})$.
- Step 4. Compute the moving direction as a vector \vec{V}_i by using the following equation:

$$\vec{V}_i = \begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix}. \quad (5.7)$$

- Step 5. Compute the direction angle θ_{new} for the vehicle to turn toward the node

N_{i+1} by using the following equation:

$$\theta_{new} = \tan^{-1}\left(\frac{Y_i}{X_i}\right). \quad (5.8)$$

Step 6. Compute the navigation distance for the vehicle to advance as $d = |\vec{V}_i|$.

Step 7. Calculate the odometer calibration angle θ_{cali} using the following equation obtained from Section 3.3.3:

$$\theta_{cali} = 0.00000476 \times d^2 + 0.00592048 \times d + 4.16437951. \quad (5.9)$$

Step 8. Compute the rotation angle for the vehicle as $\theta_{turn} = \theta_{new} - \theta_{odo} + \theta_{cali}$.

Step 9. Turn the vehicle leftward for the angle θ_{turn} if θ_{turn} is larger than zero; otherwise, turn the vehicle rightward for the angle θ_{turn} .

Step 10. Move the vehicle forward for the distance d .

Step 11. Read the next node data. If there exist remaining nodes, repeat Steps 3 though 8; else, finish the navigation.

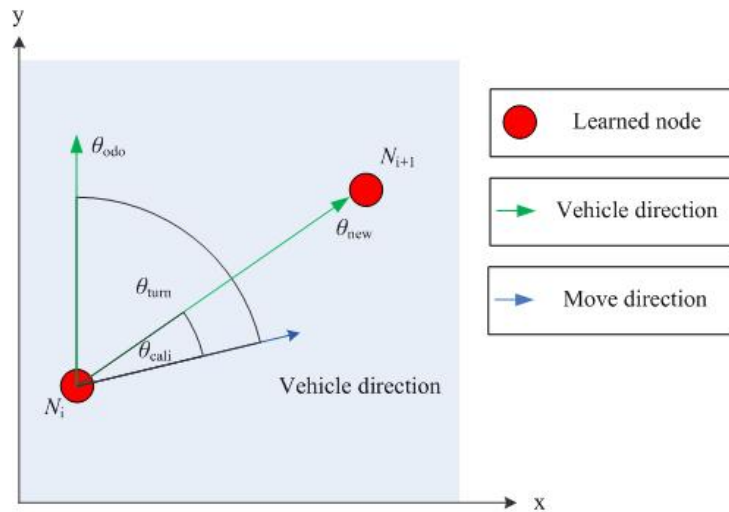


Figure 5.4 Computation of the turning angle and move distance.

Additionally, the navigation path is composed of two kinds of special path points. The first kind is turning point where the vehicle turns a large angle θ_{find} , obtained from Section 4.2.3. Because θ_{find} is a large angle, we hope the vehicle can really turn this angle in the navigation process; otherwise, the resulting vehicle trajectory will be too far away from the desired path learned in the learning stage, according to our experience of experiments conducted in this study. Hence, when the vehicle navigates to this kind of path point, it is commanded to turn to the inverse direction of the angle $\theta_{\text{find}} + \theta_{\text{cali}}$ instead of the angle θ_{turn} calculated in the algorithm above. The second kind of path point is monitoring point where the vehicle learned a 2D object, obtained from Section 4.5.2. When the vehicle navigates to this kind of path point, the vehicle will “see” the 2D object again and the system will monitor this object. After successfully monitoring an object, we can take advantage of the matching result to adjust the vehicle location [9]. An illustration of the patrolling process is shown in Figure 5.6. Some examples of experimental results are shown in Figure 5.5 and Figure 5.7.

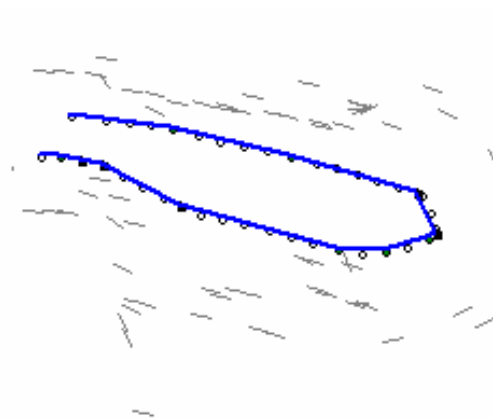


Figure 5.5 An example of a planned path of an ellipse shape.

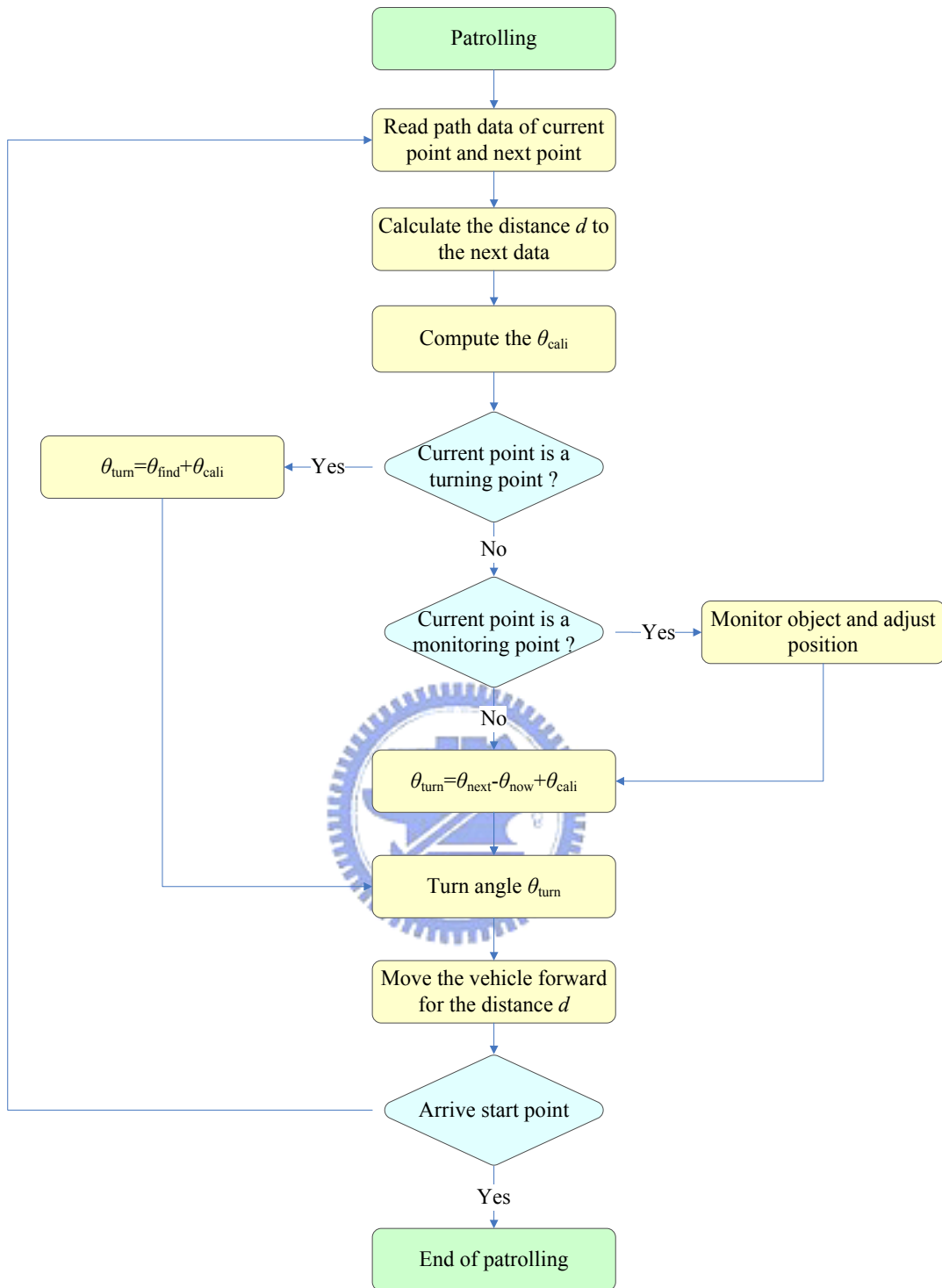


Figure 5.6 An illustration of proposed patrolling process.



(a)



(b)



(c)



(d)



(e)

Figure 5.7 An example of experimental results. (a) The vehicle monitors an object. (b) The vehicle walks to a turning point (c) The vehicle monitors another object. (d) The vehicle goes back to the original start point.

Chapter 6

ALV Navigation by Ultrasonic Signal Sequences

6.1 Ultrasonic sensing in ALV System

The vehicle of our system has eight ultrasonic sensors that facilitate implementations of object detection and range finding functions for collision avoidance, recognition, localization, and navigation. The ultrasonic sensors are equipped in the vehicle at fixed locations around the vehicle: one on each side and six facing outward at twenty degree intervals, as shown in Figure 6.1.

The ultrasonic sensors may be used to obtain the information of distances by calculating the time spent from the start of firing an ultrasonic sound to the end of receiving a reflected sound. The distance data could have errors if the plane, which receives the incident, is not perpendicular to the fired sound ray. If the angle of incidence grows, then the erroneous distance values will grow, too. This phenomenon needs to be noticed when using this kind of sensor.

In our system, we set the left distance D_L as the value detected from ultrasonic sensor No. 0 plus half of the width of the vehicle, set the right distance D_R as the value detected from ultrasonic sensor No. 7 plus half of the width of the vehicle, and set the front distance D_F as the average value of those detected from ultrasonic sensors No. 3 and No. 4. The system catches the reflected values F_{emit} times every second as a sequence of distance data.

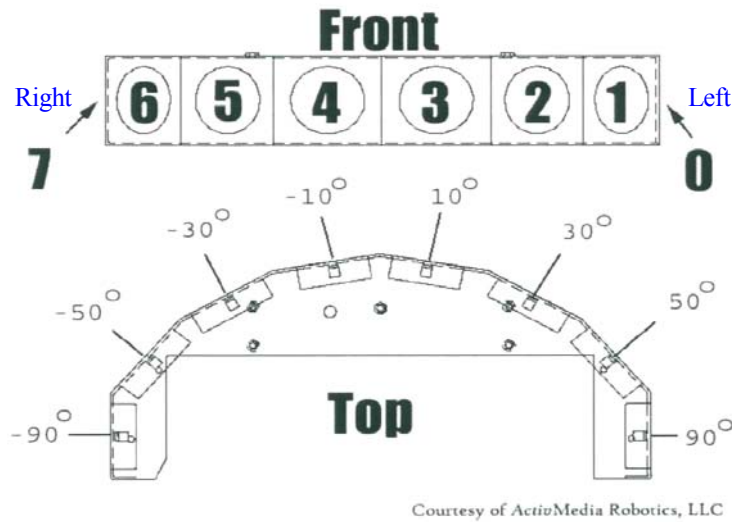


Figure 6.1 The positions of ultrasonic sensors around the vehicle.

6.2 Principle of Navigation

6.2.1 Learning Strategy

The navigation paths in an indoor environment are usually composed of some straight lines which are connected by turning points. Before navigation, the vehicle has to learn the parameters of $Width$, $Direction_{turn}$, $Direction_{detect}$, and $Distance_{detect}$ of the environment. For a straight line of a path in the indoor environment, if it is a hallway with two walls at both the left and the right sides, then the value $Width$ is set equal to the width of the hallway. But if the straight path only has one wall at the left or the right side, then $Width$ is set equal to double the distance between the vehicle and the wall. An illustration is shown in Figure 6.2. For a turning point of the path, if the vehicle needs to turn to the right at the turning point, then we set $Direction_{turn} = right$, and if the vehicle needs to turn to the left, we set $Direction_{turn} = left$.

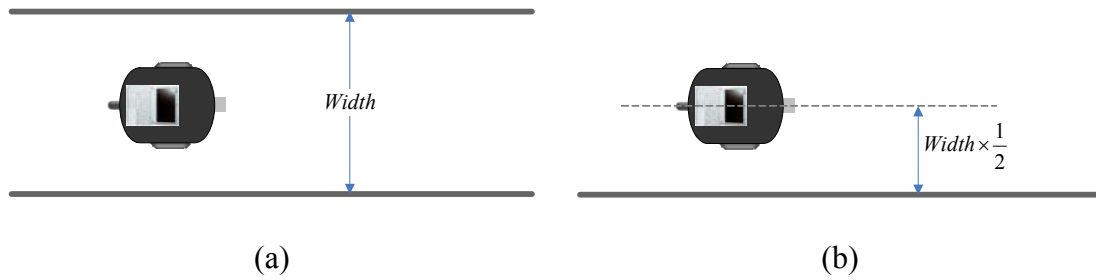


Figure 6.2 An illustration of the parameter of *Width*. (a) The hallway with two walls. (b) The hallway only has one wall.

Additionally, the vehicle has to know the conditions of turning points. Turning points are usually located at the end of a wall or a hallway. The distance values caught from the ultrasonic sensors will vary to large values at the position of the end of a wall. The parameter $Direction_{detect}$ is used to save the direction where the environment has caused such a kind of large-value variation and the vehicle can detect such variations to find out turning points. If a turning point is located at the end of the left wall, then we set $Direction_{detect} = left$; otherwise, we set $Direction_{detect} = right$. Moreover, we set $Direction_{detect} = left$ or $right$ to mean that the turning point is located at the end of the left or the right wall, respectively. If the turning point is located at the end of the hallway, it means that there is something stopping the hallway. In that case, we set $Direction_{detect} = front$. The last parameter needs to learn is $Distance_{detect}$ which is the distance between the turning point and the start position of the end of the wall in the environment. An illustration is shown in Figure 6.3.

6.2.2 Navigation Strategy

After learning the navigation path in the environment, the vehicle can navigate along the learned path by analyzing the sequential signals acquired by the ultrasonic

sensors. The vehicle has two missions, one to navigate along the middle line of the hallway, and the other to keep the direction of navigation parallel to the hallway.

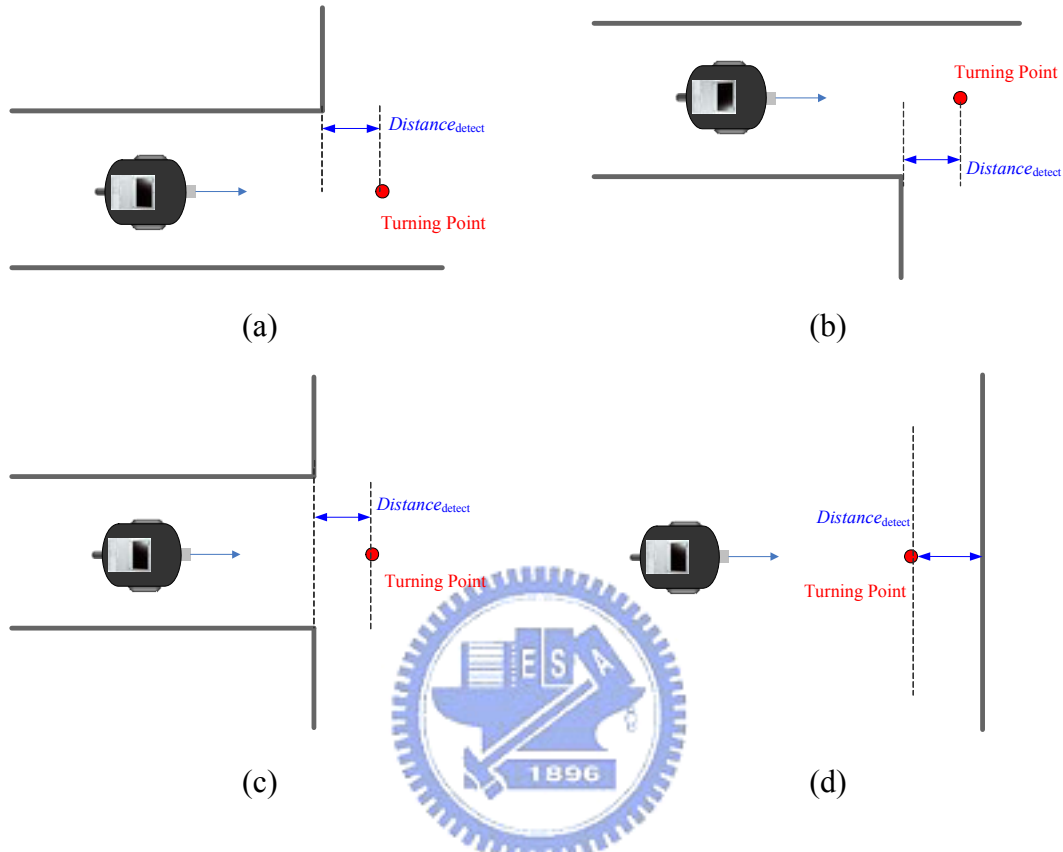


Figure 6.3 An illustration of the parameter of $Direction_{detect}$ (a) $Direction_{detect} = left$.
 (b) $Direction_{detect} = right$. (c) $Direction_{detect} = left + right$. (d) $Direction_{detect} = front$.

We begin a navigation session at a start position in the hallway, and let it go forward by a fixed speed $Speed$. From the ultrasonic sensors, the system retrieves distance information D_i which includes the left and the right distances ($D_{i,L}$, $D_{i,R}$) at time i , and analyzes n sets of data, D_{i-n} , D_{i-n+1} , \dots , D_i , before time i . The vehicle needs to adjust its direction in four situations. When the vehicle moves close to the wall at the left side gradually, the system could find out the fact that the left distance of the vehicle is reducing or that the right distance of the vehicle is growing. Then, if the vehicle is close enough to the left wall, that is, if the right distance of the vehicle

exceeds half of the width of the hallway, then the vehicle turns an angle θ_{little} to the right, that is, the vehicle turns when the following conditions are met:

$$D_{k,L} > D_{k-1,L}, \quad (6.1)$$

$$\text{or } D_{k,R} < D_{k-1,R}, \quad (6.2)$$

$$\text{with } D_{k,R} < \frac{1}{2} \times \text{Width}, \quad (6.3)$$

for every k between $i - n$ and i .

On the other hand, when the vehicle moves close to the wall at the right side gradually, the system could find out the fact that the right distance of the vehicle is reducing or that the left distance of the vehicle is growing. Then, if the vehicle is close enough to the right wall, that is, if the left distance of vehicle exceeds half of the width of the hallway, the vehicle turns an angle θ_{little} to the left, that is, the vehicle does so if

$$D_{k,L} < D_{k-1,L}, \quad (6.4)$$

$$\text{or } D_{k,R} > D_{k-1,R}, \quad (6.5)$$

$$\text{with } D_{k,L} < \frac{1}{2} \times \text{Width} \quad (6.6)$$

for every k between $i - n$ and i . An illustration of the situation of adjusting the direction of the vehicle is shown in Figure 6.4. An illustration of navigation in a hallway is shown in Figure 6.5.

It seems superfluously if we both consider Eq. (6.1) and (6.2), but it is necessary. The walls in environments are not completely planar and the sensor could get some noise, so we need to consider the data from both sides of the wall. Eq. (6.4)

and (6.5) are for use to handle the same situation as above.

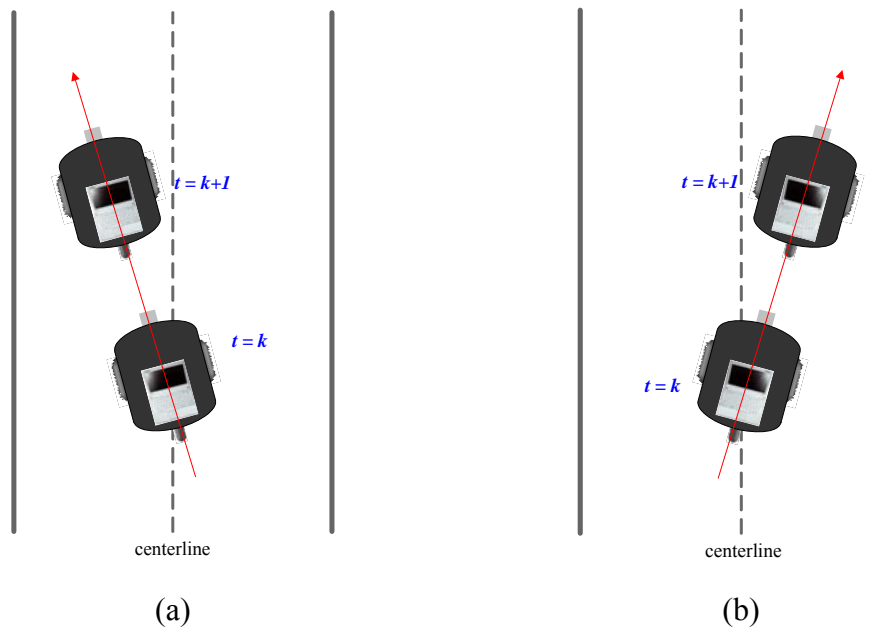


Figure 6.4 An illustration of the situation of adjusting the direction of the vehicle. (a) The vehicle should turn an angle to the right (b) The vehicle should turn an angle to the left

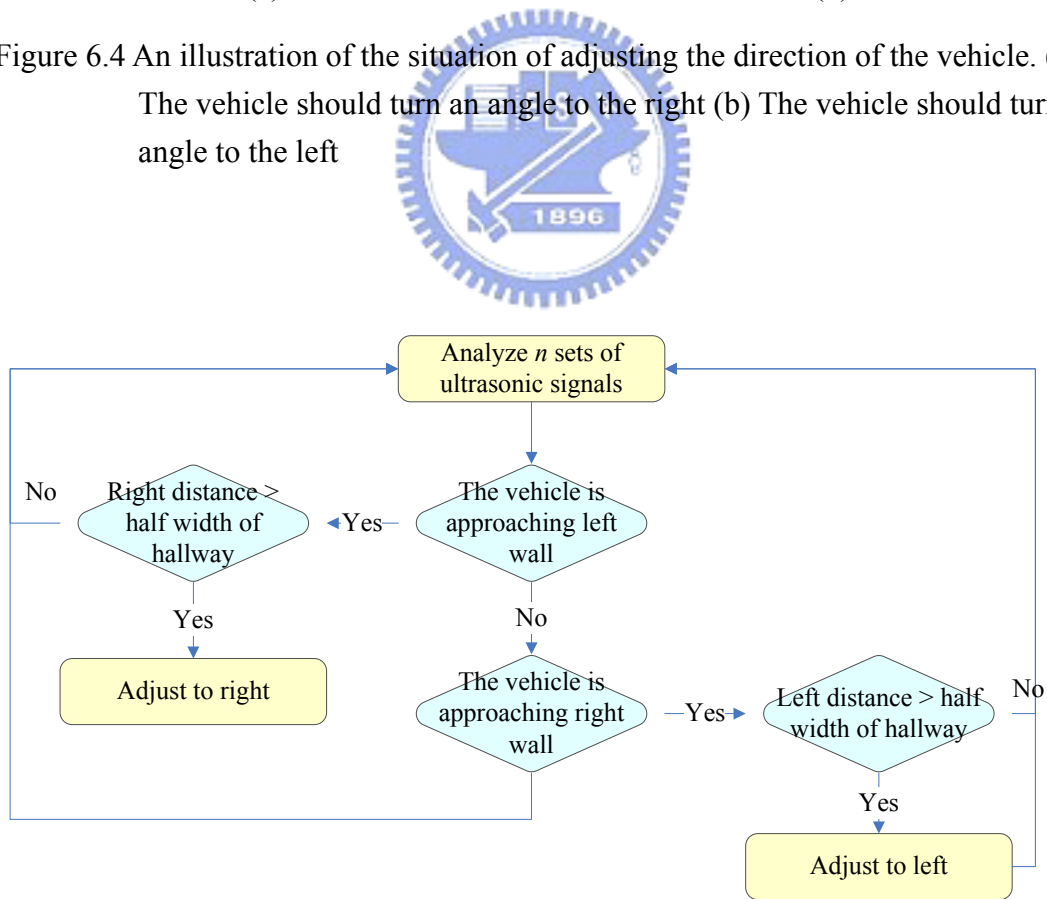


Figure 6.5 An illustration of navigation in a hallway.

For detection of a turning point, we have learned the parameters of the direction to detect $Direction_{detect}$ and the distance to detect $Distance_{detect}$ in the learning stage. If the parameter $Direction_{detect}$ is *left* or *right*, the system calculates the times N to detect the changes of the environment by:

$$N = \left(\frac{Distance_{detect}}{Speed} \right) \times F_{emit}. \quad (6.7)$$

And it also detects the changes of $Direction_{detect}$, and if the times reach N , then the vehicle arrives the turning point and turns to $Direction_{turn}$. If the parameter $Direction_{detect}$ is *front*, then the vehicle will keep detecting the front distance until the distance is smaller than $Distance_{detect}$. Then we consider that the vehicle has arrived at the turning point and command the vehicle to turn to the direction $Direction_{turn}$.



6.3 Applications in Tour Navigation

A set of path information P_i of a straight line and a turning point is composed by four parameters $Width_i$, $Direction_{turn,i}$, $Direction_{detect,i}$, and, $Distance_{detect,i}$. We can construct a complete navigation path by learning many sets of such path information, P_1, P_2, \dots, P_n . While navigation, every time the vehicle goes through the turning point, the system reads the next set of path information and detects the next turning point. The navigation path can be considered as a simple automaton as shown in Figure 6.6.

6.4 Experimental Results

In this section, we show some experimental results of the proposed navigation

system. In Figure 6.7, the vehicle navigates on a path.

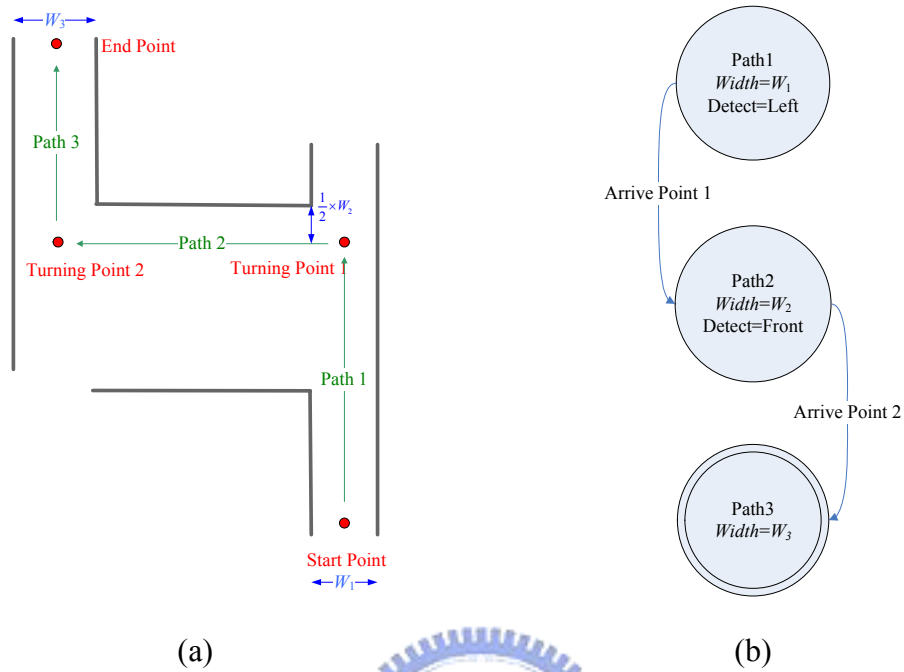


Figure 6.6 The navigation path. (a) An example of paths. (b) The path is transformed into an automation

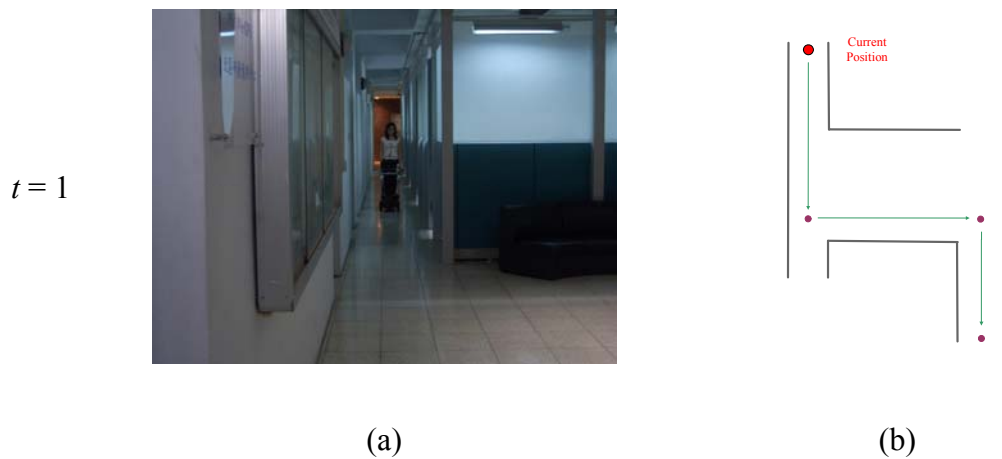
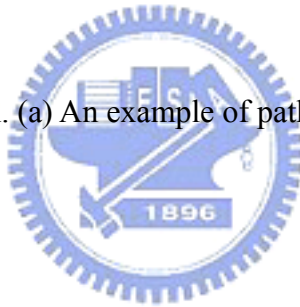
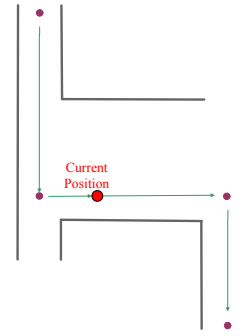


Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map.

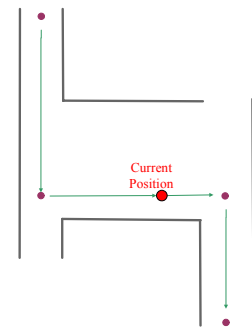


Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map. (continued)

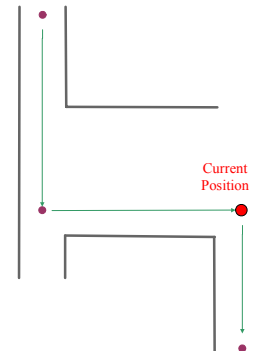
$t = 6$



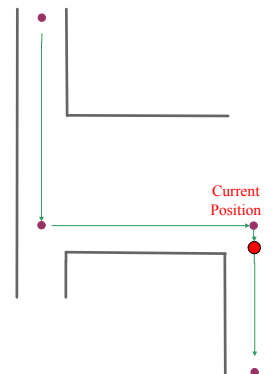
$t = 7$



$t = 8$



$t = 9$

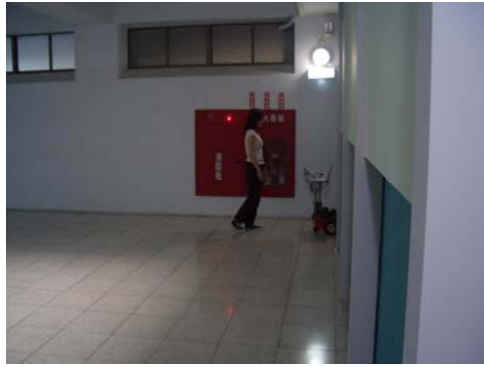


(a)

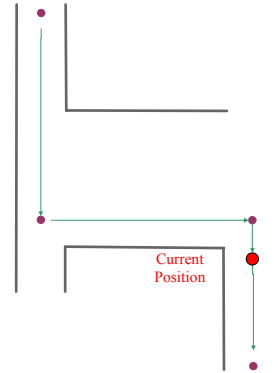
(b)

Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map. (continued)

$t = 10$



(a)



(b)

Figure 6.7 An experimental result of navigation in an indoor environment. (a) The vehicle navigates on the path. (b) The navigation map. (continued)



Chapter 7

Experimental Results and Discussions

7.1 Experimental Results

In this section, we will show some experimental results of the proposed person following and patrolling system and the navigation system in indoor environments. Experiments for this study were performed at the hallway out of the Computer Vision Laboratory at the Department of Computer Science in Engineering 3 Building, and the hallway of the sixth floor of the Microelectronics and Information Systems Building, all in National Chiao Tung University.

The user interface of the person following and patrolling system is shown in Figure 7.1. The proposed system has four modes: detection, following, learning and turning. The system will detect humans in acquired images in the detection mode. In the following mode, the system will follow a target person by the use of the extracted clothes features. After a user presses the start button, the system will start the detection mode and detects any person's clothes. Then the system will extract the clothes region; and if it decides a person to be existent, it will change the detection mode to the following mode, and then finish the current cycle, as shown in Figure 7.2. After changing to following mode, the vehicle will follow the person and calculate a new value of the color of clothes of the target person at the same time, the reference value of clothes will renew continually.

When the person turns to a direction to face a picture, then the vehicle will enter

the learning mode. If the person turns to the right or left for a while, the vehicle will go to the position where the person stands and turn to the same direction to see and remember the view. Then, the system will analyze the view and find the region and the horizontal line of the 2D object, as shown in Figure 7.3 and Figure 7.4.

Besides, when the vehicle follows the person, if the person turns fast in front of the vehicle, the system will change to the turning mode. Then the vehicle will turn a big angle to the right direction to search the disappearing person, as shown in Figure 7.5.

A record map of navigation session is shown in Figure 7.6. After path planning, the vehicle will go back to the start point in the learning process, and monitor the learned objects, as shown in Figure 7.7. An illustration of the experimental process is shown in Figure 7.8.

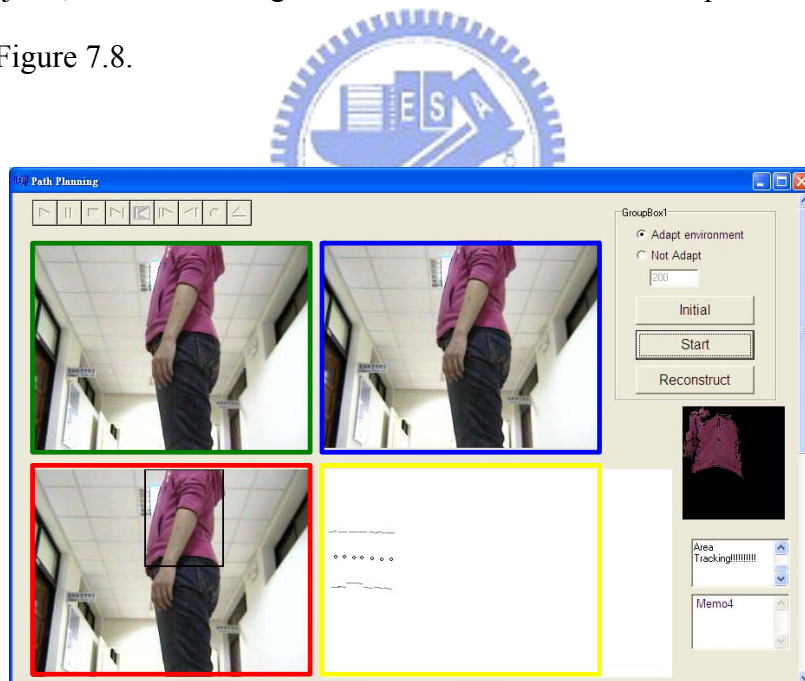


Figure 7.1 An interface of the experiment. The green box shows the image stream and the blue box shows the input image at this moment. The yellow box shows the difference image and the red box shows the draft of the map.



Figure 7.2 An experimental result of extraction of the clothes. (a) The input image. (b) The image of the extracted clothes.

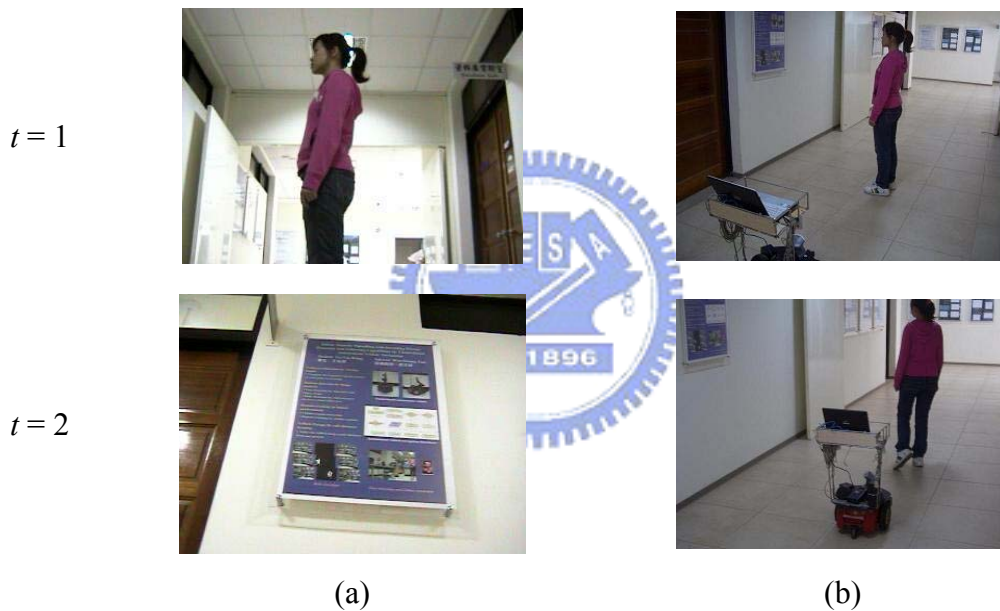


Figure 7.3 An experimental result of the learning mode. (a) The input image. (b) The position of the vehicle.

For the second type of vehicle system using ultrasonic sensors, the user interface of the system is shown in Figure 7.9. We simulate the vehicle as a guiding vehicle for guiding visitors from the door area of an elevator into an office. An illustration of the learned data, the navigation map, and the actual navigation path created in the experiment is shown in Figure 7.10 and the record map of navigation session is shown in Figure 7.11.

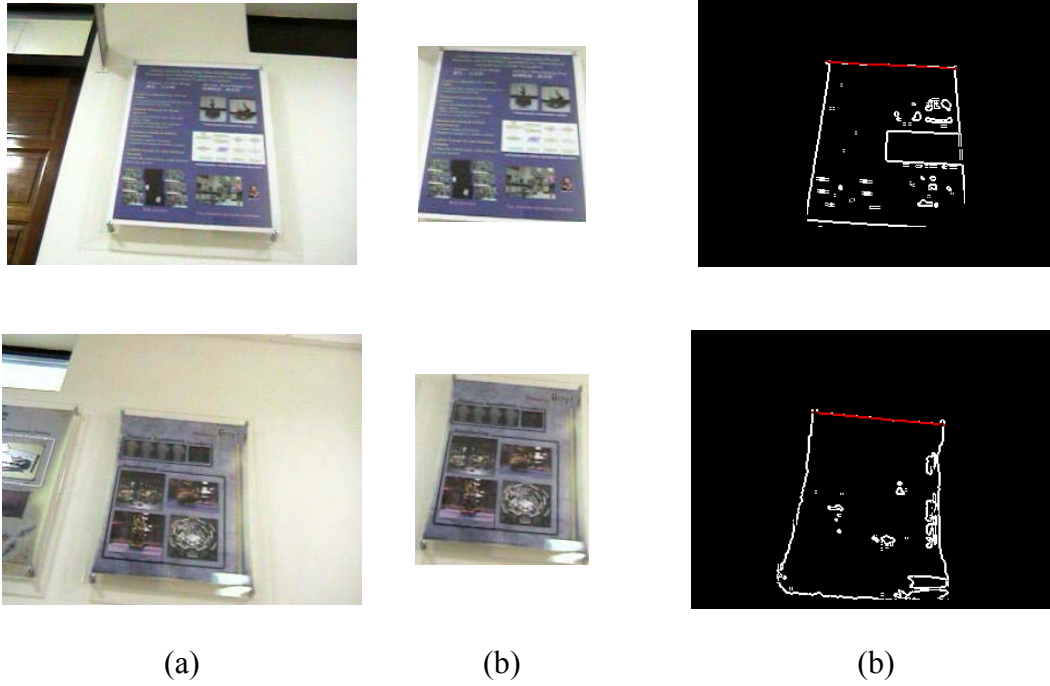


Figure 7.4 An example result of learning a 2D object. (a) The input image. (b) The region of the 2D object. (c) The horizontal line of the 2D object.

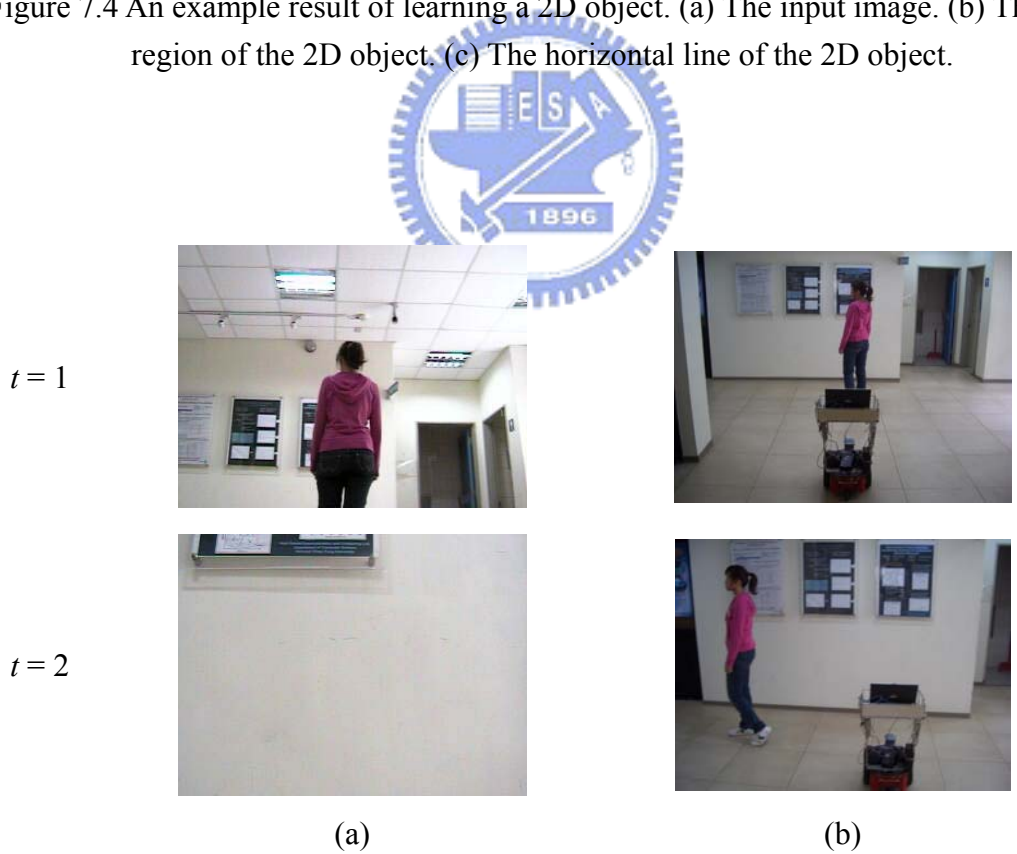


Figure 7.5 An experimental result of following a person turning fast. (a) The input image. (b) The position of the vehicle.

$t = 3$



(a)



(b)

Figure 7.5 An experimental result of following a person turning fast. (a) The input image. (b) The position of the vehicle. (continued)

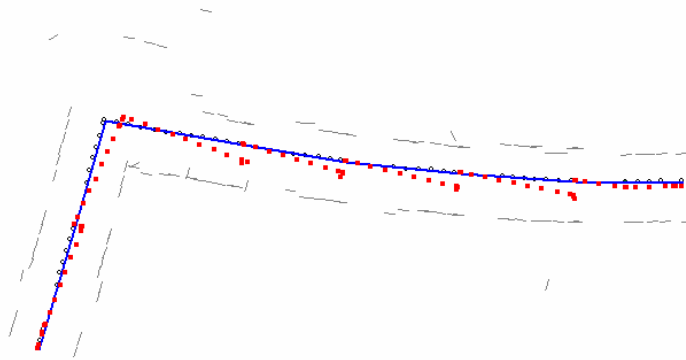


Figure 7.6 An experimental result of navigation. The blue line segments show the planning result. And the red points show the real patrolling path of the vehicle.

After a user presses the start button, the system will read the path data and start the navigation. In the hallway environment, the vehicle will adjust the direction by analyzing sequence of signals detected from ultrasonic sensors and navigation on the middle of the hallway. When the vehicle arrive the detection node, it will start detecting the position of turning node by the learned information of the distance and the direction, and take a turn. The experimental results are as shown in Figure 7.12.

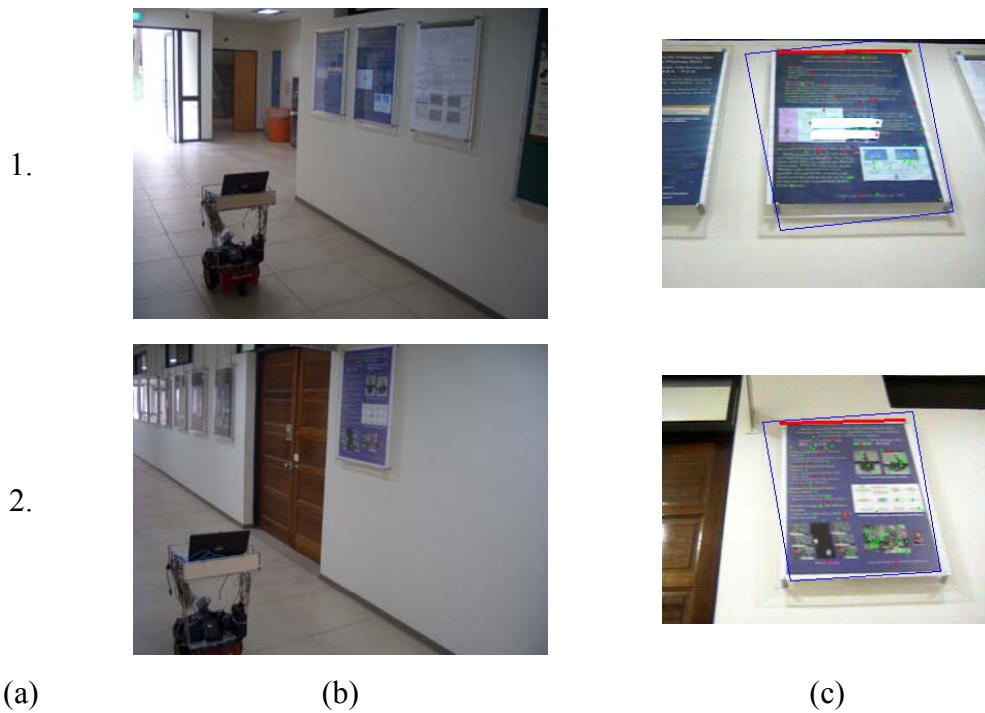


Figure 7.7 The experimental result of object monitoring and navigation path correction. (a) The numbers of monitored objects. (b) The vehicle monitors the objects. (c) The matching result and the horizontal line used for path correction.

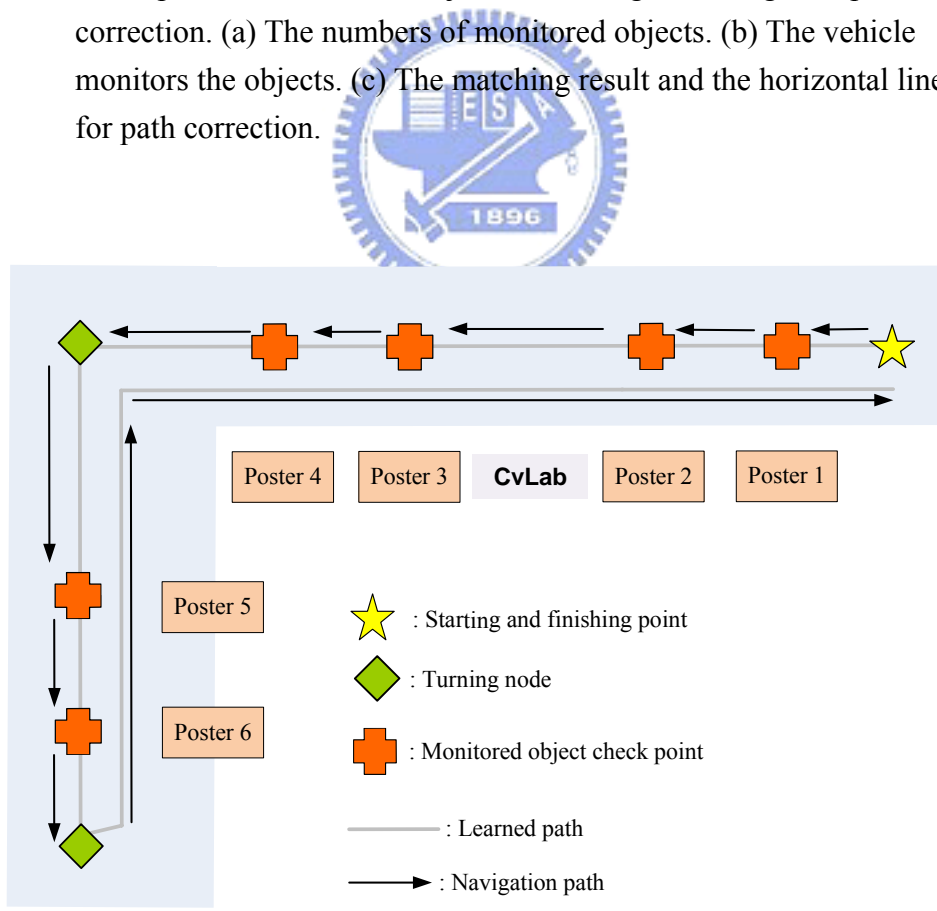


Figure 7.8 An illustration of the path of following a person and monitoring objects in an experiment.

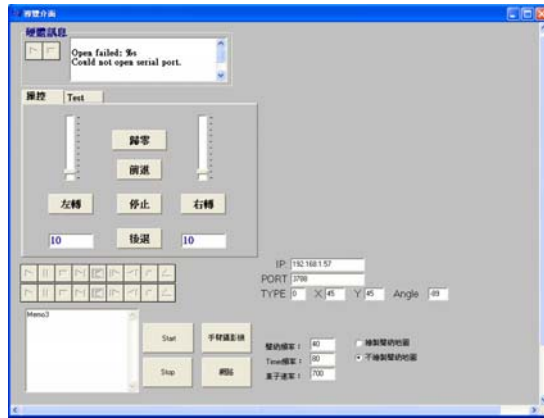


Figure 7.9 Interfaces of the experiment.

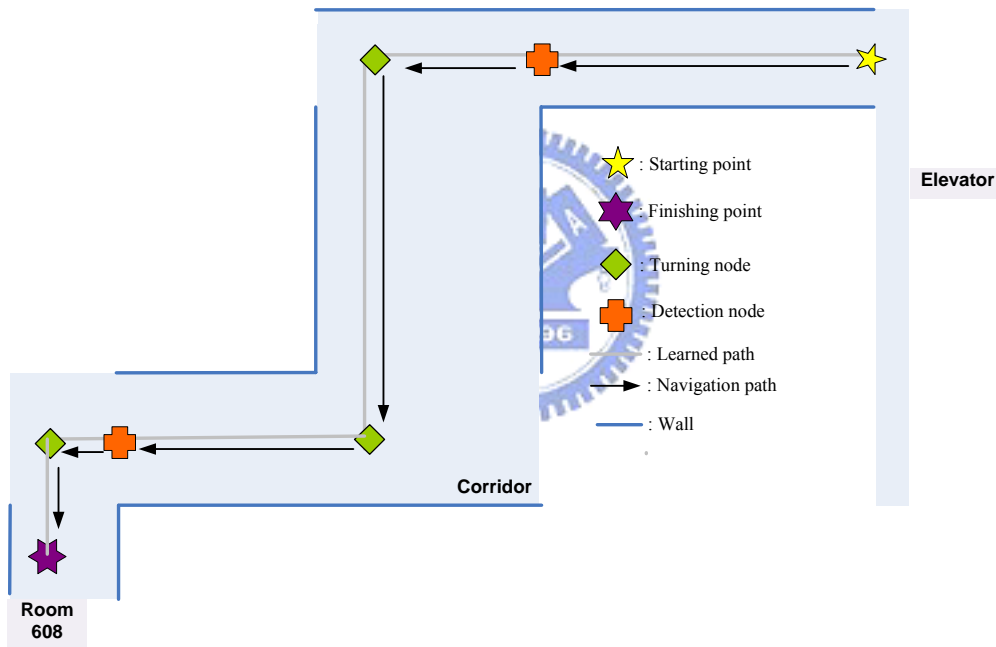


Figure 7.10 An illustration of learned data and navigation path.

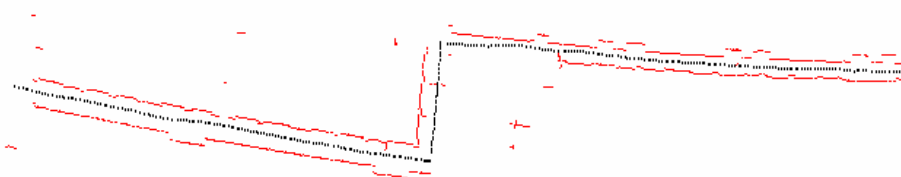


Figure 7.11 An experimental result of navigation.

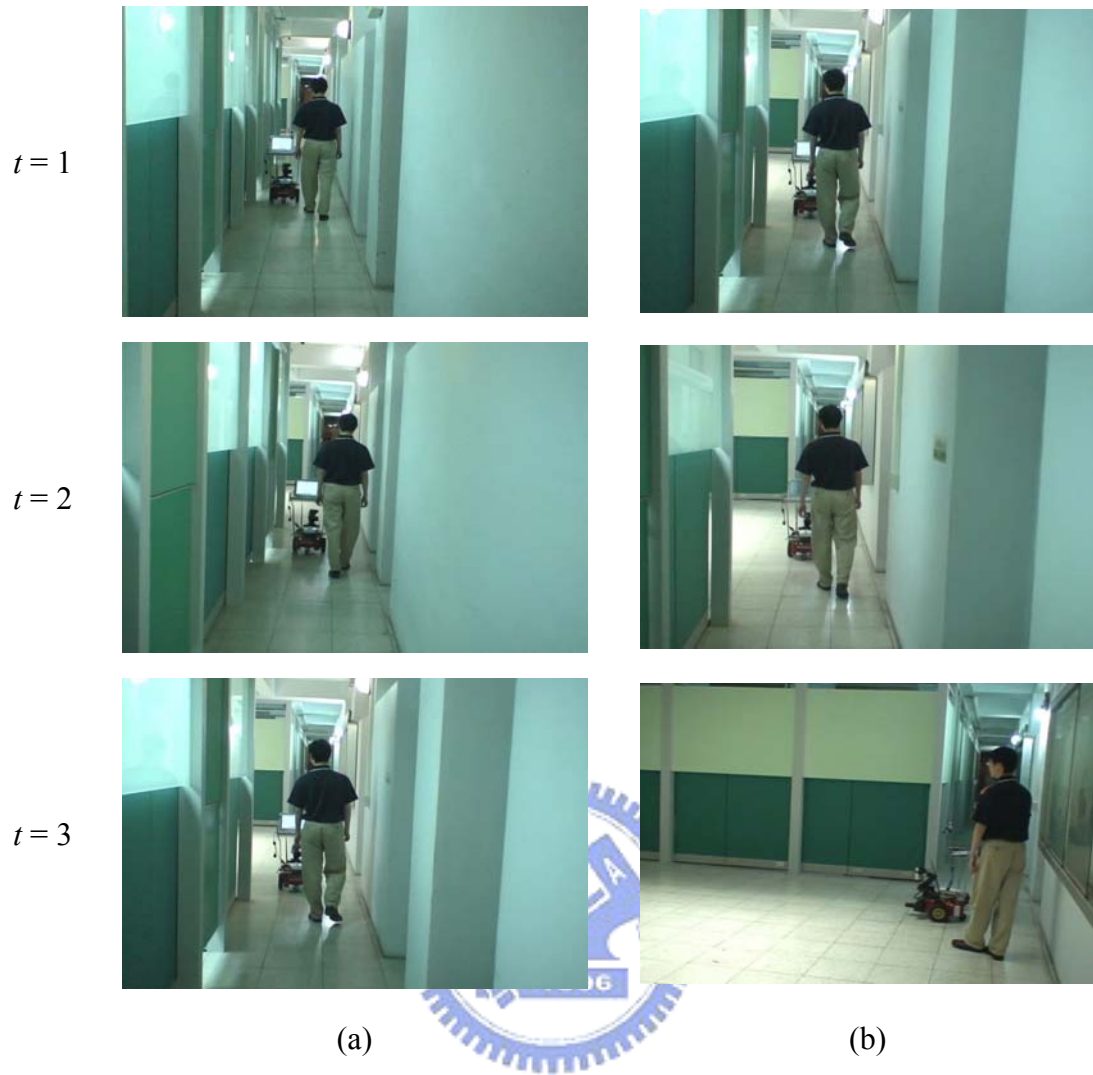


Figure 7.12 The experimental result of the vehicle guide a person. (a) Navigation in a hallway. (b) Navigation at a turning point.

7.2 Discussions

By analyzing the experimental results of person following and guidance, some problems are identified as follows.

- (1) We built an odometer calibration model to calibrate the deviation angle which is made by the action of moving forward of the vehicle. But there are still some deviations made by the action of turning of the vehicle which should be

calibrated.

- (2) The human following process by the use of the clothes color might incur errors due to other people wearing clothes of the same color. When different people wear clothes of the same color, the system will be confused and cannot decide which person is the target to follow.
- (3) In the detection of the facing direction, we use the color of the hair, i.e., the black color, for judging the facing direction of the person. If the hair of the person is dyed or if the person has no hair, the system cannot work.
- (4) In our system, there are some restrictions to finding 2D objects in images, for example, the height of 2D objects must be fixed and the objects need be pasted on a wall of light color. But pictures in common environment are not always pasted at the same height, and the paint of the wall is not always light. This problem should be solved in the future.
- (5) The maps of paths are drawn by the coordinate data which are provided by the odometer in our systems. But the path in the map is 'curved' and different from the traversed path, because of the imprecise odometer data resulting from the incremental mechanical errors.

Chapter 8

Conclusions and Suggestions for Future Works

8.1 Conclusions

Several techniques and strategies have been proposed in this study and integrated into two autonomous vehicle systems for navigation in indoor environments: one with human following and object monitoring capabilities, and the other with person guidance capability.

At first, a method of improving the practicability of a camera calibration method has been proposed. Some steps in a past method for finding the intersection points on a grid board repeat incurring errors easily. We use the property of the cross shape, which has eight points with color changes around the center point, to detect the cross shape. Besides, a method of calibrating the odometer equipped on the vehicle has been proposed. For the purpose of reducing the incremental mechanical errors suffered by the vehicle, we built an odometer calibration model by using a curve fitting technique to calculate a smooth curve of the values of the angle of deviation in different distance moves taken by the vehicle.

The proposed system for patrolling in the indoor environment has three processes: the following process, the path planning process, and the patrolling process. The system is in the following process at the beginning. When the vehicle is following a person, the illumination in common indoor environments is not actually

uniform. If the vehicle keeps using the value detected at the initial time to follow a person, then it may lose the tracking of the person because the illumination over the vehicle changes while the vehicle moves in the environment. We have proposed an improvement of an adopted process to dynamically adjust the values of C_b and C_r for detecting the changing color of the clothes of a followed person.

Then, a method for detecting a disappearing person who turns fast in front of the vehicle has been proposed for use in the person following process. When the person makes a fast turn at a corner, the system will use the recorded information to command the vehicle to turn to the right direction for searching the disappearing person in the acquired image.

When the vehicle follows a person, it will also detect the facing direction of a person by using a scan line. It gets a false result easily, because the scan line is thin and the numbers of pixels on the scan line are little. So, we have proposed a method by using a scan rectangle to increase the number of pixels for detection and enhance the accuracy of detection.

After detecting the facing direction of a person, the vehicle goes to the position where the person just stands and turns to the same direction of the person, and then analyzes the view to find a 2D monitored object. We have proposed a method by using region growing and the Hough transform to find the region and a horizontal line of the object. And we save these data of objects and use them in the patrolling process.

The vehicle collects the path data of positions and distances from the odometer and the ultrasonic sensors, and uses these data to refine the path into a smooth path. A method of path planning by using the MSE criterion has been proposed. Also proposed is a method to adjust the MES threshold automatically, by which the planned path can adapt to different environments.

In another system for navigation in indoor environment, we have proposed a method of analyzing ultrasonic signals for the vehicle to navigate in a corridor environment. The vehicle can navigate in the middle of paths and detect the learned turning point. The system can be applied to be a guiding vehicle in the office environment, which can guide visitors and make some introduction to them.

The experimental results shown in the previous chapters have revealed the feasibility and practicality of the proposed systems.

8.2 Suggestions for Future Works

The proposed strategies and methods, as mentioned previously, have been implemented on a vehicle system. Several suggestions and related interesting issues worth further investigation in the future are stated in the following.

- (1) Improving the extraction of clothes colors, such as a pattern on the cloth.
- (2) Conducting human following by different features, such as texture and shape, to eliminate errors caused by the case that the different people may wear clothes of the same color.
- (3) Following a person by use not only the feature of color, like using the ultrared rays to detect the heat of a person, for overcoming huge changes of the luminance.
- (4) Improving the detection of a person's facing direction by learning other hair and skin colors or using the position of the mouth to know the direction which a person faces to.
- (5) Adding a judgment of the position of a person's body. When a person slips and falls over, the person's body may fall down on the floor. If this situation

continues for a long time, the vehicle can issue an emergence signal to call someone for help.

- (6) Detecting the heights of monitored objects by using two pictures captured at different positions.
- (7) Using a top view camera to make sure the computed position of the vehicle.



References

- [1] Y. T. Wang and W. H. Tsai, "Indoor security patrolling with intruding person detection and following capabilities by vision-based autonomous vehicle navigation," *Proceedings of 2006 International Computer Symposium (ICS 2006), International Workshop on Image Processing, Computer Graphics, and Multimedia Technologies, Taipei, Taiwan, Dec. 2006.*
- [2] K. T. Chen and W. H. Tsai, "A study on autonomous vehicle guidance for person following by 2D human image analysis and 3D computer vision techniques," *Proceedings of 2007 Conference on Computer Vision, Graphics and Image Processing, Miaoli, Taiwan, 2007.*
- [3] C. H. Ku and W. H. Tsai, "Smooth autonomous land vehicle navigation in indoor environments by person following using sequential pattern recognition," *Journal of Robotic Systems*, Vol. 16, No. 5, pp. 249-262, 1999.
- [4] B. Kwolek, "Face Tracking System Based on Color, Stereovision and Elliptical Shape Features," *Proceedings of the IEEE conference on Advanced Video and Signal Based Surveillance, Miami, Florida, U. S. A., pp. 265-270, 2004.*
- [5] K. Morioka, J. H. Lee, H. Hashimoto, "Physical Agent for Human Following in Intelligent Sensor Network," *Proceedings of 2002 IEEE International Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, pp. 1234-1239 vol.2, Oct. 2002.*
- [6] A. Treptow, G. Cielniak and T. Duckett, "Active people recognition using thermal and grey images on a mobile security robot," *Proceedings of 2005 Intelligent Robots and Systems (IROS 2005), Alberta, Canada, pp. 2103-2108, 2005.*
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints,"

International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

- [8] H. Xiao, L. Liao and F. Zhou, "Mobile robot path planning based on Q-ANN," *Proceedings of 2007 Automation and Logistics, IEEE International Conference, Jinan, China*, pp. 2650-2654, 2007.
- [9] K. C. Chen and W. H. Tsai, "A study on autonomous vehicle navigation by 2D object image matching and 3D computer vision analysis for indoor security patrolling applications," *Proceedings of 2007 Conference on Computer Vision, Graphics and Image Processing, Miaoli, Taiwan*, 2007.
- [10] A. J. Davison, "Real-time simultaneous localization and mapping with a single camera," *Proceedings of International Conference on Computer Vision, Nice, France*, Oct. 2003.

