

國立交通大學

多媒體工程研究所
碩士論文

利用擴增實境與環場電腦視覺技術作園區行車導覽之
研究

A Study on Tour Guidance by Car Driving in Park Areas Using
Augmented Reality and Omni-vision Techniques

研 究 生：陳柏丞

指導教授：蔡文祥 教授

中 華 民 國 一 〇 一 年 六 月

利用擴增實境與環場電腦視覺技術作園區行車導覽之研究
A Study on Tour Guidance by Car Driving in Park Areas Using
Augmented Reality and Omni-vision Techniques

研究生：陳柏丞

Student : Bo-Cheng Chen

指導教授：蔡文祥

Advisor : Wen-Hsiang Tsai



Hsinchu, Taiwan, Republic of China

中華民國一〇一年六月

利用擴增實境與環場電腦視覺作室外園區行車導覽系統

研究生：陳柏丞

指導教授：蔡文祥 博士

國立交通大學資訊科學與工程研究所

摘要

本研究利用架設在監控車頂上的全方位環場攝影機裝置以及投射於擋風玻璃上的平板影像，以及偵測特殊地標得到的資訊來達成直覺性且獨立的擴增實境戶外園區導覽功能。駕駛者可以利用投射在擋風玻璃的擴增影像看到目前經過的左右兩邊的建築物及其名字。

為了讓每個地標都有導覽的資訊，首先要建立園區導覽地圖，在這裡提出了環場攝影機與 PTZ 攝影機的座標轉換方法，輔助建立單一地標的區間地圖。之後，我們也提出把區間地圖合成完整的園區導覽地圖的方法，獲得每個地標在導覽地圖中的關係。

在本研究中，此環場攝影機成像系統能夠廣範圍監看任何角度的影像畫面。在這裡我們提出圓形地標在環場影像中會得到橢圓的形狀以及此橢圓面積在影像中的關係。此外利用影像處理的技術，偵測出地標在影像中的中心位置。之後計算出地標的真實世界座標，得出監控車和地標之間的方向與距離。

同時，我們利用平板投射在擋風玻璃上的影像達成擴增實境的應用。發展出顯示目前車子所在的建築物資訊的方法。本研究也提出得到平板放置的位置與投射的影像位置的方法，讓車內的人了解成像的位置。最後我們利用圖表遍歷方法以及光流法，偵測出監控車轉彎的方向以及在導覽俯視圖中如何得知目前的位置。

實驗結果顯示所提出戶外園區導覽系統確實可行。

A Study on Tour Guidance by Car Driving in Park Areas Using Augmented Reality and Omni-vision Techniques

Student: Bo-Cheng Chen Advisor: Prof. Wen-Hsiang Tsai

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

ABSTRACT

In this study, an augmented-reality based tour guidance system for use in park areas using a video surveillance vehicle and computer vision techniques has been proposed. When a user drives the vehicle in a park, he/she can get from the system tour guidance information about the names of the nearby buildings appearing along the way on the two roadsides. The building names are displayed on the iPad, which then are projected onto the windshield for the user to observe on the driver's seat, in a sense of augmenting the scene seen through the car windshield by the projected building names.

To implement a system of this augmented reality function, at first, a PTZ camera and a two-camera omni-imaging device equipped on the vehicle roof are used for acquiring PTZ-images and omni-images, respectively, for the purposes of guidance map construction and vehicle localization. Guidance map construction is carried out by use of the PTZ camera to measure feature points on nearby buildings. Vehicle localization during vehicle driving is accomplished through 3D image analysis using omni-images acquired of a series of red circular-shaped landmarks attached beforehand on along-way flat roadside objects (sidewalks, pedestrian chairs, etc.).

To detect red circular-shaped landmarks around the video surveillance vehicle during park navigation, image analysis techniques, including YC_bC_r color modeling, region growing, approximation of a circle in an omni-image by an ellipse, elliptical shape fitting, etc., are applied to segment out the corresponding elliptical-shaped landmark in the omni-image. Also proposed is a method for obtaining the center point of the ellipse in the omni-image to compute the 3D data of the detected landmark for vehicle localization.

Furthermore, methods for calculating the accurate position of the iPad image projected on the windshield, computing the accurate position of the vehicle in the guidance map, and showing the names of the nearby buildings are proposed. Finally, also proposed are methods for analyzing the vehicle turning direction by motion vectors produced by optical flow analysis, and for generating a guidance map graph for use in path planning for each tour in the park.

Good experimental results are also shown, which prove the feasibility of the proposed methods for augmented reality based tour guidance in park areas.

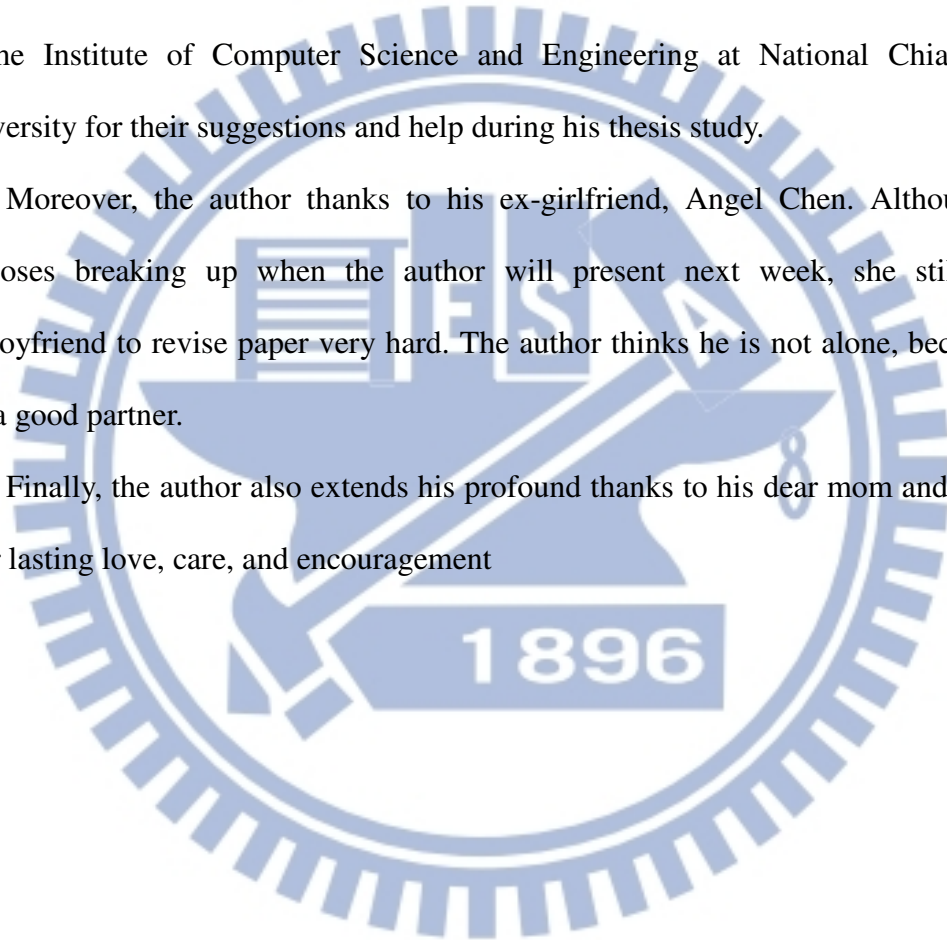
ACKNOWLEDGEMENTS

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

Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during his thesis study.

Moreover, the author thanks to his ex-girlfriend, Angel Chen. Although she proposes breaking up when the author will present next week, she still helps ex-boyfriend to revise paper very hard. The author thinks he is not alone, because he has a good partner.

Finally, the author also extends his profound thanks to his dear mom and dad for their lasting love, care, and encouragement



CONTENTS

ABSTRACT (in English)	i
ACKNOWLEDGEMENTS.....	iii
CONTENTS.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLES.....	xi

CHAPTER 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Survey of Related Studies.....	4
1.3 Overview of Proposed Methods.....	5
1.3.1 Terminologies	5
1.4 Contributions.....	8
1.5 Thesis Organization	9
CHAPTER 2 Ideas Of Proposed Methods And System Design.....	11
2.1 Ideas of System Design.....	11
2.2 System Configuration	14
2.2.1 Hardware configuration	15
2.2.2 Software configuration	17
2.2.3 Network Configuration	17
2.3 Review of Adopted PTZ-camera System	18
2.4 Review of Adopted Omni-camera System	19
2.5 System Processes	22
2.5.1 Learning Process.....	22
2.5.2 Navigation Process	24
CHAPTER 3 Creation Of Guidance Map.....	26
3.1 Introduction.....	26
3.2 Integration of Information Acquired by PTZ-camera and Omni-cameras	27
3.2.1 Review of Adopted Calibration Method for PTZ-camera	27
3.2.2 Conversion of PTZ-camera Coordinates into Omni-camera Coordinates	30
3.3 Creation of Guidance Map.....	35
3.3.1 Creation of Database for Guidance Map	35
3.3.2 Creation of Guidance Map with PTZ-camera and Omni-camer.....	36
3.3.3 Conversion of Local Maps into a Global Map.....	38
3.3.4 Algorithm for generating the guidance map	40
CHAPTER 4 Automatic Detection Of Circular-Shaped Landmark With A	

Two-Camera Omni-Directional Imaging Device	43
4.1 Introduction.....	43
4.2 Analysis of Circular-shaped landmark in Omni-directional Image	44
4.2.1 Approximation of Circles in Omni-directional Images by Ellipses.....	45
4.2.2 Property of Red Circular-shaped Landmarks in Omni-directional Images Approximated by Ellipses.....	49
4.3 Red Circular-shaped Landmark Detection.....	54
4.3.1 YCbCr Color Model	54
4.3.2 Region Growing.....	56
4.3.3 Elliptical Shape Fitting in Omni-directional Images	59
4.4 Distance Estimation between a Surveillance Vehicle and a Circular-shaped Landmark.....	60
4.4.1 Review of Adopted Pano-mapping Method for Omni-image and 3D Data Acquisition Process	61
4.4.2 Calculation of Landmark Distance	67
CHAPTER 5 Using Augmented Reality And A Two-Camera Omni-Directional Imaging Device For Park Guidance.....	68
5.1 Introduction.....	68
5.2 Implementation of Augmented Reality Tour Guidance in a Surveillance Vehicle with an iPad	69
5.2.1 Idea of Projecting Images Displayed on iPad onto Car Windshield.....	69
5.2.2 Coordinate Estimation for Projecting iPad onto the Windshield	70
5.3 Showing Names of Buildings to the Left and Right of Surveillance Vehicle on the Windshield.....	77
5.3.1 Computing Accurate Position of Surveillance Vehicle on Guidance Map.....	77
5.3.2 Decision of Left and Right Buildings	82
CHAPTER 6 Navigation By Guidance Map Traversals Using Vehicle Turning Information	86
6.1 Introduction.....	86
6.2 Analysis of Vehicle Turning by Motion Vectors in Omni-images.....	87
6.2.1 Review of Vehicle Direction Analysis by Optical Flows.....	87
6.2.2 Vehicle Turning Decision	87
6.3 Organization of Guidance Map for Park Navigation by Graph Traversals	93
6.3.1 Creation of Guidance Map Graph.....	93
6.3.2 Updating of Landmark Labels on Guidance Map Graph during	

Driving	94
CHAPTER 7 Experimental Results And Discussions	98
7.1 Experimental Results	98
7.2 Discussions	107
CHAPTER 8 Conclusions And Suggestions For Future Works.....	108
8.1 Conclusions.....	108
8.2 Suggestions for Future Works	109
REFERENCES	111



LIST OF FIGURES

Figure 1.1 Illustration of proposed guidance system using augmented reality techniques..	3
Figure 1.2 A flowchart of proposed system process..	9
Figure 2.1 The video surveillance vehicle used in this study with a two-camera omni-imaging device and a PTZ camera affixed on the car roof. (a) A front view of the vehicle. (b) A side view of the vehicle..	11
Figure 2.2 Positions of cameras affixed to the video surveillance vehicle roof and the corresponding FOV. (a) The omni-camera is affixed at the rear-middle of the car roof. (b) The omni-camera is affixed at the right-front of the car roof. (c) The PTZ-camera is affixed at the middle of the car roof. (b) The PTZ-camera is affixed at the border of the car roof.....	13
Figure 2.3 An example of landmark detection and guidance map. (a) An omni-image of a landmark detected at a sidewalk. (b) A generated guidance map showing the relative position of the car. (c) A projected image on the windshield.	14
Figure 2.4 Structure of the proposed surveillance system.	16
Figure 2.5 The architecture of the local network used in this study	18
Figure 2.6 Relationship of the mirror and the CMOS sensor in camera.....	19
Figure 2.7 (a) Relation between the world coordinates system and the image coordinate system. (b) Simple geometry between the mirror and the CMOS sensor in the camera	20
Figure 2.8 Flowchart of calibration of omni-cameras and PTZ-camera.....	22
Figure 2.9 Flowchart of learning guidance map.	23
Figure 2.10 Flowchart of tour guidance.....	24
Figure 2.11 Flowchart of tour guidance.....	25
Figure 3.1 An illustration of transformation between image coordinate system (ICS) and spherical coordinate system (SCS) and Cartesian coordinate system.	28
Figure 3.2 Camera calibrations 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.	29
Figure 3.3 Camera views of Figure 3.2. (a) View of Figure 3.2(a). (b) View of Figure 3.2(b). (c) The bmp image with camera calibration information.....	30
Figure 3.4 An illustration of the layout of the video surveillance car roof.....	30
Figure 3.5 The WCS used in this study which includes the SCS..	31
Figure 3.6 The angle between the detected point and the PTZ-camera (a) the	

PTZ-camera has a tilt angle of $\varphi_c < 0$ (b) the PTZ-camera has a tilt angle of $\varphi_c > 0$ (c) the PTZ-camera has a pan angle of $\theta_c < 0$ (d). the PTZ-camera has a pan angle of $\theta_c > 0$	32
Figure 3.7 Illustration of proposed guidance map.	33
Figure 3.8 Structure of the communication between two laptops used in this study...	36
Figure 3.9 Between two local maps there is an overlapping point.	40
Figure 3.10 The GUI designed in this study for creating local maps (a) The interface for using the omni-cameras. (b) The interface for using the PTZ-camera.	41
Figure 3.11 Creating the guidance map. (a) Various feature points on the global map. (b) The guidance map after organizing these feature points	42
Figure 4.1 A flowchart of circular-shaped landmark detection.....	44
Figure 4.2 An illustration of the omni-camera coordinate system involved in this study	46
Figure 4.3 Top view from the z direction showing the relationship between new and original coordinate system with the new image coordinate system (u', v') obtained by rotating the u -axis through an angle of θ_w with respect to the center of the circular-shaped landmark W	47
Figure 4.4 A photo of a used landmark.	50
Figure 4.5 Top view from the z direction showing the area of the ellipse A is equal to the area of the ellipse B when the distance xw' is the same.	50
Figure 4.6 The results of curve fitting of the areas of the ellipses and the uwi'	54
Figure 4.7 An illustration of the 3D YC_bC_r color model in [15].	55
Figure 4.8 An illustration of red-colored region detection using YC_bC_r color model (a) The original omni-image. (b) The binary image of the detection result... ..	56
Figure 4.9 The binary images of the landmark detection. (a) The image before region growing. (b) The image after region growing.....	57
Figure 4.10 An illustration of the region growing process — the blue region represents the elliptical-shaped region and the white region represents the non-elliptical-shaped region. Once the scan point is found to be in the elliptical-shaped region, the region growing process starts	58
Figure 4.11 Detection of the landmark by ellipse shape fitting. (a) The image before region growing. (b) The image after region growing (c) Generating an ellipse for every region. (d) Deciding the best-fit ellipse shape, where the blue shape is the best-fit ellipse, and the green shape is an erroneous ellipse	61
Figure 4.12 Interface to for user to select the landmark points.	62
Figure 4.13 Mapping between a radius distance r and elevation angle ρ	63
Figure 4.14 Illustration of mapping between the azimuth-elevation angle pair of the omni-image and the horizontal and vertical axes of the pano-mapping table,	

respectively..	64
Figure 4.15 Computation of 3D information using the two-camera omni-directional imaging device. (a) The ray tracing of a scene point P in the imaging device with a hyperbolic-shaped mirror. (b) A triangle in detail (part of (a)).....	65
Figure 4.16 System configuration of upper omni-camera with a hyperbolic-shaped mirror	66
Figure 4.17 The relative position of the landmark and the video surveillance car. (a) The omni-image. (b) The position of the video surveillance car on the guidance map	67
Figure 5.1 An illustration of the augmented reality display of the image on the iPad. (a) The projected image on the windshield (b) The relationship between the eyes of a visual angle and the positions of the seats.....	71
Figure 5.2 An illustration of the projection of iPad image onto the windshield.....	72
Figure 5.3 An illustration of the projected image.	74
Figure 5.4 The four corner points of the iPad.....	75
Figure 5.5 Illustration of the coverage of the projected image on the windshield.....	76
Figure 5.6 Illustration of selected detection regions. (a) The left-front region. (b) The left-back region.	78
Figure 5.7 Illustration of the car direction on the guidance map.....	80
Figure 5.8 (a) The detected landmark of the left-front region. (b) The position of the video surveillance vehicle by (a) (c) The detected landmark of left-back region. (d) the position of the video surveillance vehicle by (c) (e) The detected landmark of left-front region again (f) the position of the video surveillance vehicle by (e)	81
Figure 5.9 Illustration of displaying the names of buildings.	82
Figure 5.10 An experimental result of displaying the names of the buildings. (a) The name of the current-visited buildings. (b) The position of the video surveillance vehicle by (a) (c) The name of the next-visited buildings. (d) The position of the video surveillance vehicle by (c).....	85
Figure 6.1 Transformation of a motion vector from the ICS into the WCS. (a) An illustration of the camera system and the motion vector. (b) The ray tracing of a scene point P on the ground projected on the hyperbolic-shaped mirror.....	90
Figure 6.2 A distribution chart of the direction angles of motion vectors..	92
Figure 6.3 A graph of finite state machine proposed to determine the moving direction [17].....	93
Figure 6.4 The graph of the guidance map. (a) An illustration of organizing guidance map. (b) The guidance map graph of our experimental environment.....	94

Figure 6.5 A flowchart of deciding the label of the next landmark on the guidance map graph for graph traversals.	96
Figure 6.6 An experimental result of turning direction determination. (a) The position of the video surveillance vehicle and the detected forward direction. (b) The position of the video surveillance vehicle and the decided left direction..	97
Figure 7.1 Illustration of the experimental guidance area..	98
Figure 7.2 A learning interface of the experiment. (a) An interface for using the omni-camera. (b) An interface for using the PTZ-camera.....	99
Figure 7.3 A guidance map of the experiment environment	99
Figure 7.4 An experimental result of landmark detection. (a) An elliptical-shaped landmark in the omni-image. (b) Image of landmark detection by region growing	100
Figure 7.5 A guidance map of the experimental environment.....	100
Figure 7.6 An experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	101
Figure 7.7 Another experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	102
Figure 7.8 A third experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	103
Figure 7.9 An experimental result of vehicle turning determination. (a) The omni-image, the image of detection result, and the decided vehicle direction of positive x-axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	104
Figure 7.10 Another experimental result of vehicle turning determination. (a) The omni-image, the image of detection result, and the decided vehicle direction of positive x-axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	105
Figure 7.11 A third experimental result of vehicle turning determination. (a) The omni-image, the image of detection result, and the decided vehicle direction of positive x-axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.....	106

LIST OF TABLES

Table 2.1 Specifications of the laptop computers and the pad used in this study.	17
Table 2.2 Specifications of used COMS cameras	21
Table 4.1 The computed values of the area of the elliptical shape and u_w	56
Table 4.2 An example of the pano-mapping table.....	.69
Table 6.1 The range of the angles of the three vehicle moving directions.....	97



Chapter 1

Introduction

1.1 Motivation

Nowadays, with the progress of computer technology, video cameras have become more and more popular in various applications, bringing convenience and improving welfare in our daily life. For example, a video surveillance vehicle with its roof equipped with video cameras can help a driver to monitor surrounding environments and to be aware of dangerous conditions so that upcoming accidents can be avoided in time. In other cases, people often equip cars with video cameras like digital driving recorders, and once a traffic accident occurs, the driver can clarify the responsibility for the event by inspecting the video record.

In addition to video cameras, the augmented reality technique can help the driver of a vehicle as well. For example, there exists commercially a type of head-up display on the windshield, which can show the car speed and the fuel consumption condition in real time. Through this type of equipment, the driver can look up at the information displayed on the windshield conveniently rather than look down at the dashboard. The latter action will cause accidents sometimes. Besides, by projecting the information of the road in front and the surrounding buildings on the windshield (their names, numbers, etc.), the augmented reality technique can help implementing a car navigation system which provides the driver the surround condition at any time for safe driving on roads.

Most researches of vision-based techniques for the mentioned applications are based on the use of traditional projective video cameras. Generally speaking, the

field of view (FOV) of a projective video camera is limited. When we want to see panoramic images of a spot from the roof of a video surveillance vehicle, four or more of such cameras are required. If we implement a video surveillance car system using such cameras, it will consume a larger cost and need more overhead in computing. Therefore, we choose the omni-camera to be our imaging device in this study. Panoramic images of the environment around the video surveillance vehicle can be covered by such an omni-camera system.

Although current augmented reality techniques seem useful to assist drivers, they are difficult to use for the purpose of acquiring the current information of the environment around the car. Specifically, a GPS is often used to get outdoor location information, but using the GPS for this purpose has some problems. Firstly, the GPS is not very accurate, and this will yield possibly erroneous information to the augmented reality system. Secondly, the GPS does not work in tunnels or inside big buildings. Therefore, we propose in this study is to integrate omni-cameras and augmented reality techniques with a video surveillance vehicle to implement an effective augmented reality system for outdoor uses. Through omni-image analysis and omni-camera calibration, we can get accurate information of the outdoor environment for the proposed augmented reality system. Furthermore, the omni-cameras equipped on the video surveillance vehicle and the augmented reality system inside the vehicle can be used together to develop functions for various applications.

To sum up, the research goal in this study is to develop a tour guidance system for use by car driving. To accomplish the goal, we will use a video surveillance car with a two-camera omni-imaging device as an experimental platform. Also, we will use an iPad to simulate the head-up display device for use inside the video surveillance car. Moreover, in order to get the information of a guidance map, we

will place landmarks in the guidance area and use computer vision techniques to obtain the positions of nearby buildings via analysis of the omni-images of these landmarks acquired by the omni-camera on top of the vehicle. An illustration of the proposed system is shown in Fig. 1.1.

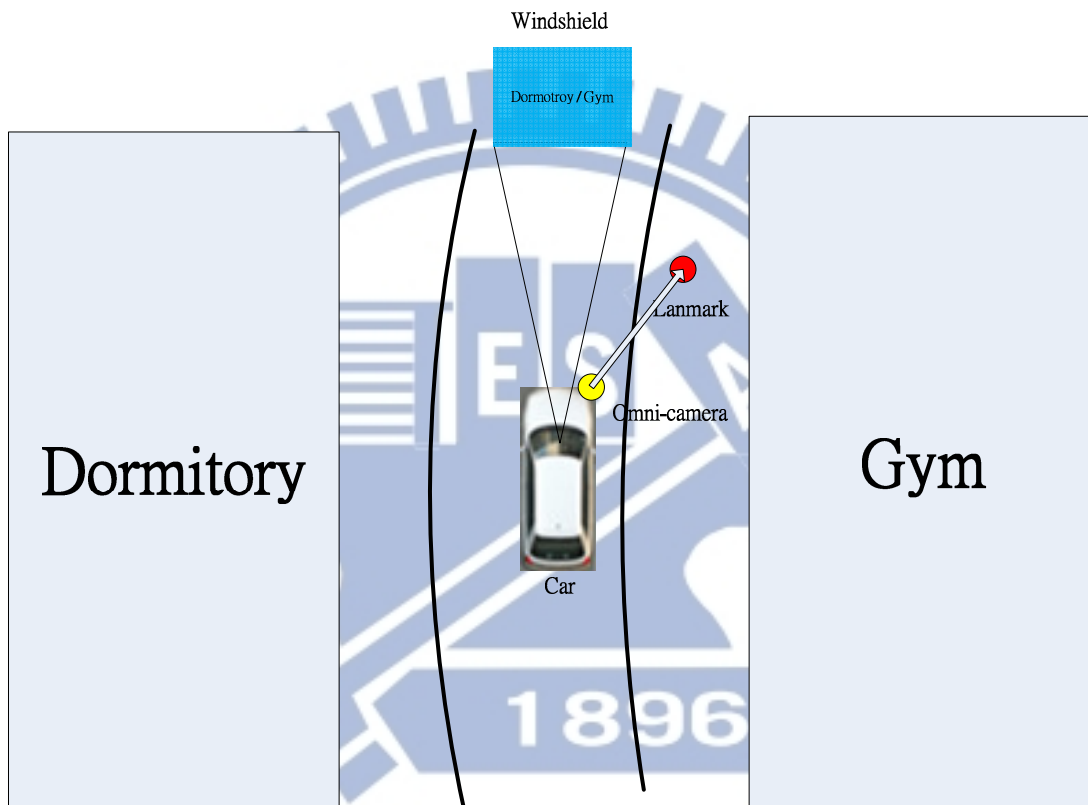


Figure 1.1 Illustration of proposed guidance system using augmented reality techniques

Listed below are the more detailed descriptions of the desired capabilities of the proposed system.

- (1) The proposed video camera system can help creating a guidance map. For this, a local map is created first by the use of the two-camera omni-image system and a PTZ camera. Moreover, we propose a method to convert local maps into a global map.
- (2) The system can detect a circular-shaped red-colored landmark automatically and

measure its position and height. It then marks the relative location of the video surveillance car on the guidance map.

- (3) The system is capable of computing the accurate position of the information displayed on the iPad for projection onto the windshield. Moreover, in the projected image onto the windshield, the name of nearby buildings on the two roadsides can be displayed for tour guidance.
- (4) The system can analyze omni-images to decide the turning direction of the vehicle when necessary, and choose a correct path on the guidance map.

1.2 Survey of Related Studies

In recent years, many techniques have been developed for use in the applications of video surveillance. For example, Trivedi et al. [1] proposed methods to enhance vehicle safety by video surveillance systems with omni-cameras; and Kim and Suga [2] proposed a method to detect motion vectors using optical flows with an omni-camera.

Moreover, new video devices can be designed by combining projective cameras and mirrors, resulting in new types of omni-vision systems. Also, two omni-images together taken by these omni-vision systems can provide stereo information. In this aspect, a method to obtain stereo information for mobile robot navigation with an omni-vision system which consists of two mirrors and one camera was proposed by He et al. [3]; and a method to detect a suspicious passer-by automatically by a stereo vision system which consists of a pair of two-camera omni-imaging devices was proposed by Yuan and Tsai [4]. Jeng and Tsai [5] proposed a method based on the concept of pano-mapping table to calibrate omni-cameras without knowing the extrinsic parameters of the omni-cameras.

In addition, many methods for land vehicle navigation by the use of landmarks

have been developed in the past decade. Betke and Gurrivits [6] proposed a standard localization method to identify surrounding landmarks in the environment and find their corresponding locations in an environment map built in advance. Wu and Tsai [7] proposed a vision-based method for location estimation for use in autonomous land vehicle (ALV) navigation in indoor environments using circular landmark information in an omni-directional image. To detect various landmarks in omni-images, Ho and Chen [8] proposed an algorithm to detect ellipses, and Wang and Tsai [9] proposed a method which uses an elliptic skin model to detect human faces by color and shape features in images.

In this study, an integration of augmented reality and a video surveillance system for various applications is developed. In a similar work, Grosch [10] proposed a method to use panoramic images for navigation in a real environment. Also, augmented reality for outdoor applications has been widely investigated in recent years. In this aspect, Lee et al. [11] conducted a study on using omni-vision to track, in large areas, the camera pose which simulates the user's view in the augmented reality environment; and Reitmayr and Drummond [12] proposed a model-based hybrid tracking system for outdoor augmented reality in urban environments enabling accurate, realtime overlays for a handheld device. In addition, techniques of augmented reality can assist a driver in driving a car. For example, one can use the augmented reality technique to create a head-up display device in a car. Sandor et al. et al. [13] has proposed a method for doing this type of task in their study.

1.3 Overview of Proposed Methods

1.3.1 Terminologies

The definitions of some related terms used in this study are described as follows.

1. *Omni-camera*: a camera system with a traditional projective camera and a reflective mirror which can be used to capture images of 360° fields of view.
2. *Omni-image*: an image captured with an omni-camera.
3. *Two-camera omni-imaging device*: a camera system constructed with two omni-cameras connected back to back and coaxially.
4. *Video surveillance vehicle*: a car with a two-camera omni-imaging device equipped on the car roof as well as a laptop for use as a control unit and a pad inside the car used to develop a tour guidance system.
5. *Optical flow*: a method to estimate the motions of shapes, surfaces, and edges of concerned objects between two sequential images.
6. *Guidance map*: a map to show the relative position of a car with respect to the entire guidance area.
7. *Landmark*: a circular-shaped red-colored flat object which we place on roadsides of the guidance area to get the information of the guidance map.
8. *PTZ camera*: a projective camera with a controllable configuration which can be used to pan, tile, and zoom.
9. *Perspective-view image*: an image acquired with the PTZ camera.

1.3.2 Brief Descriptions of Proposed System

There are four goals in developing the proposed system as described in the following.

1. The system is able to create local maps using omni-images acquired with the two-camera omni-imaging device and perspective-view images acquire by the PTZ camera, and convert the local maps into a global one.
2. Regardless of high or low terrains, the system is able to detect a landmark and obtain its 3D information, and compute accordingly the location of the video

surveillance car on the guidance map.

3. The proposed system is able to compute the position of the information to be displayed on the iPad which, together with the names of nearby buildings at the current position, is then projected onto the windshield of the video surveillance vehicle.
4. The system is able to analyze the acquired omni-images to decide the car turning direction (left or right) when encountering a road cross or branch.

In order to achieve the above goals, the following are the major steps of the system process of the proposed guidance system.

1. Set up the previously-mentioned two-camera omni-image device on the roof of the video surveillance vehicle on the front-right corner and an iPad inside the car.
2. Calibrate the omni-cameras for six outward view directions and use the pano-mapping technique to generate six corresponding pano-mapping tables.
3. Integrate the PTZ-camera coordinate system with the omni-camera coordinate system to create local maps in the guidance area.
4. Convert the local maps into a global one.
5. Detect any landmark in the surround of the video surveillance car using the two acquired omni-images by eliminating the ground regions in the two omni-images, using the property of a circular shape in each omni-image, and getting the center point of the landmark as its location in each ;omni-image.
6. Use the landmark location detected in each of two omni-images to compute the 3D information of the vehicle.
7. Organize the graphs in the guidance map and use a graph traversal method to create a navigation path, including road branches as well as turnings.
8. Calculate the relative positions of the image to be projected onto the windshield and display the names of the currently-visited buildings on two roadsides.

9. While the surveillance vehicle is going to turn, analyze its turning direction by an optical flow method, transform the extracted motion vectors into the world coordinate system (WCS), and keep track of the nodes of the graph.
10. Repeat the above steps until the vehicle reaches a pre-selected destination.

A brief illustration of the above system process is shown in Fig. 1.2.

1.4 Contributions

The following is a list of the major contributions made in this study.

1. A method is proposed to convert the PTZ-camera coordinates into the omni-camera coordinates, to help creating the global guidance map.
2. A local network is constructed, which integrates a two-camera omni-imaging device, a PTZ-camera, and two laptop computers for use in guidance map creations.
3. A method for detecting an ellipse-shaped landmark and computing accordingly the accurate position of the car in the guidance map is proposed.
4. A method of projecting the screen of the iPad onto the windshield of the video surveillance vehicle to implement an augmented reality-based head-up display device.
5. A method for computing the accurate position of the iPad image projected on the windshield is proposed.
6. A method for computing the accurate position of the car in the guidance map and showing the names of the nearby buildings on the windshield is proposed.
7. A method for analyzing car turning directions is proposed.
8. A method for creating graphs from the guidance map and keeping track of the node being visited by graph traversal is proposed.

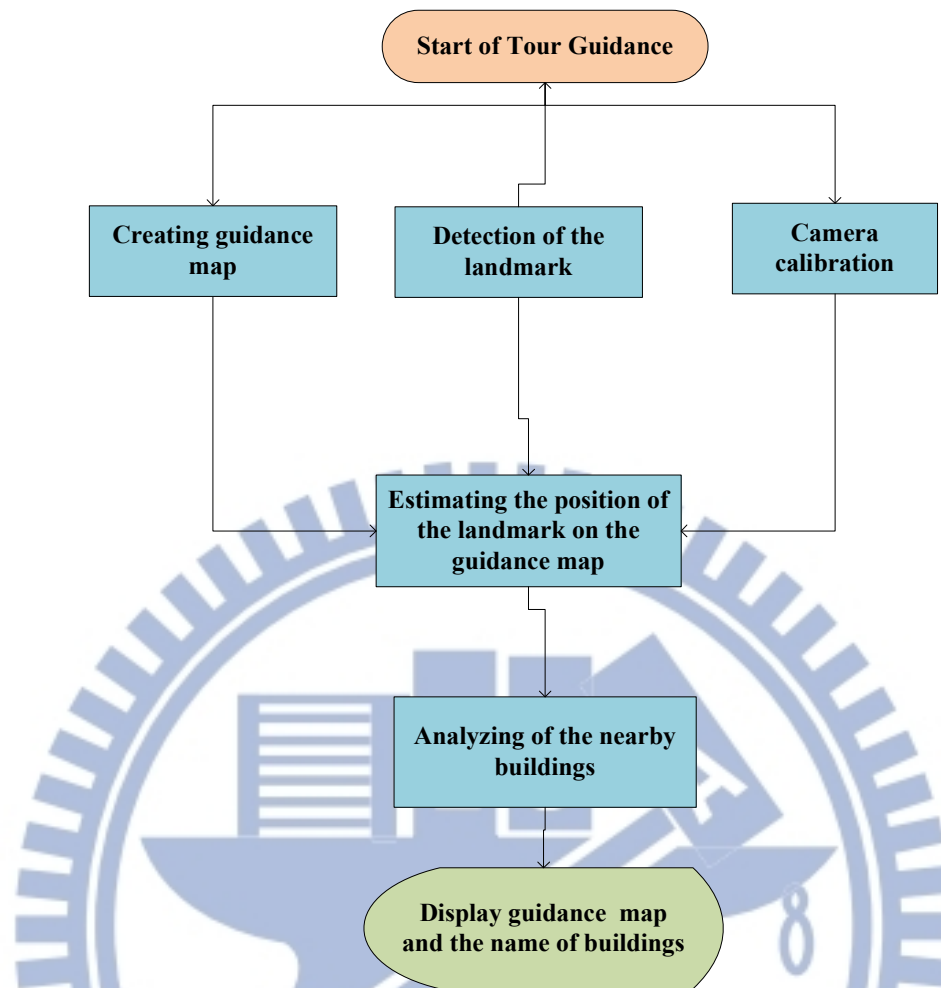


Figure 1.2 A flowchart of proposed system process.

1.5 Thesis Organization

In the remainder of this thesis, we introduce the system configuration and the idea behind the proposed system in Chapter 2. The structure of the two-camera omni-imaging device is also described. In Chapter 3, the proposed method for using the two-camera omni-imaging device and the PTZ camera to help creating a guidance map is described. In Chapter 4, the proposed method for detecting a landmark to obtain the stereo data is described. In Chapter 5, the proposed method for projecting tour guidance information on an iPad onto the windshield is described. In Chapter 6, the proposed method for organizing graphs in the guidance map to conduct graph traversal

for proceed car tours in the guidance area is described. In Chapter 7, experimental results and discussions are included. Finally, conclusions and some suggestions for future works are given in Chapter 8.



Chapter 2

Ideas of Proposed Methods and System Design

2.1 Ideas of System Design

In order to monitor the surrounding environment of the video surveillance vehicle, we affix a two-camera omni-imaging device and a PTZ camera, instead of traditional projective cameras, on the roof of the surveillance vehicle in this study. The omni-camera can be used to monitor 360 degrees of the car surround and enhance acquisition of necessary scene information outside the car. The PTZ camera can pan, tilt, and zoom by computer control. The aforementioned structure of the surveillance vehicle used in this study is shown in Figure 2.1. Note that the two-camera omni-imaging device includes two omni-cameras aligned coaxially and back to back, as mentioned previously.



(a)



(b)

Figure 2.1 The video surveillance vehicle used in this study with a two-camera omni-imaging device and a PTZ camera affixed on the car roof. (a) A front view of the vehicle. (b) A side view of the vehicle.

The video surveillance vehicle embraces high mobility so that we can move the

onboard camera system to everywhere, but we have to determine the best locations on the car roof where the omni-imaging device and the PTZ camera should be affixed, respectively. We discuss where to affix the omni-imaging device at first. As illustrated in Figures 2.2(a) and 2.2(b), if we affix the device at the front middle of the car roof; a half of the omni-image acquired with the device is undesirably the car body. But if we affix it at the right-front position of the surveillance vehicle roof, only a quarter of the omni-image taken with the same imaging device is the car body. Therefore, in this study we decide to affix an omni-imaging device at the right-front of the surveillance vehicle roof. Second, as illustrated in Figures 2.2(a) and 2.2(b), the PTZ-camera is affixed at the border position of the surveillance vehicle roof. If the PTZ-camera were affixed instead at the middle position of the surveillance car roof, it would cover undesirably a half of the car body.

Furthermore, we analyze the images acquired with these imaging devices to create a guidance map and extract information for tour guidance by detecting landmarks appearing in the acquired images. The landmarks are detected by algorithms proposed in this study, such as region growing, YCbCr color modeling, ellipse fitting, etc. The details will be described in Chapter 4. The landmarks can be used to navigate the vehicle in the guidance area, but they do not all lie on planes of the same height. Using the two omni-cameras in the omni-imaging device, we can solve this problem by estimating relevant 3D data of the landmarks.

Moreover, in order to create the guidance map, the PTZ-camera can be used at first to get feature points of nearby buildings (the detail will be described in Chapter 3), then the positions of landmarks are obtained accordingly, and finally the location of the video surveillance vehicle is computed (the detail will be described in Chapter 4). Also, we use an iPad-like mobile device (called a pad hereafter) to simulate a head-up display on the windshield. The position of the projected image appearing on

the pad onto the windshield can be estimated so that the driver does not have to put the pad into an accurate pose under the windshield (the detail will be described in Chapter 5). The projected image is designed to include the names of the currently-visited buildings on the left and right road sides, allowing the driver to understand the current vehicle's location in the guidance area. An example of the guidance map and a displayed image with building names on the pad is shown in Figure 2.3.

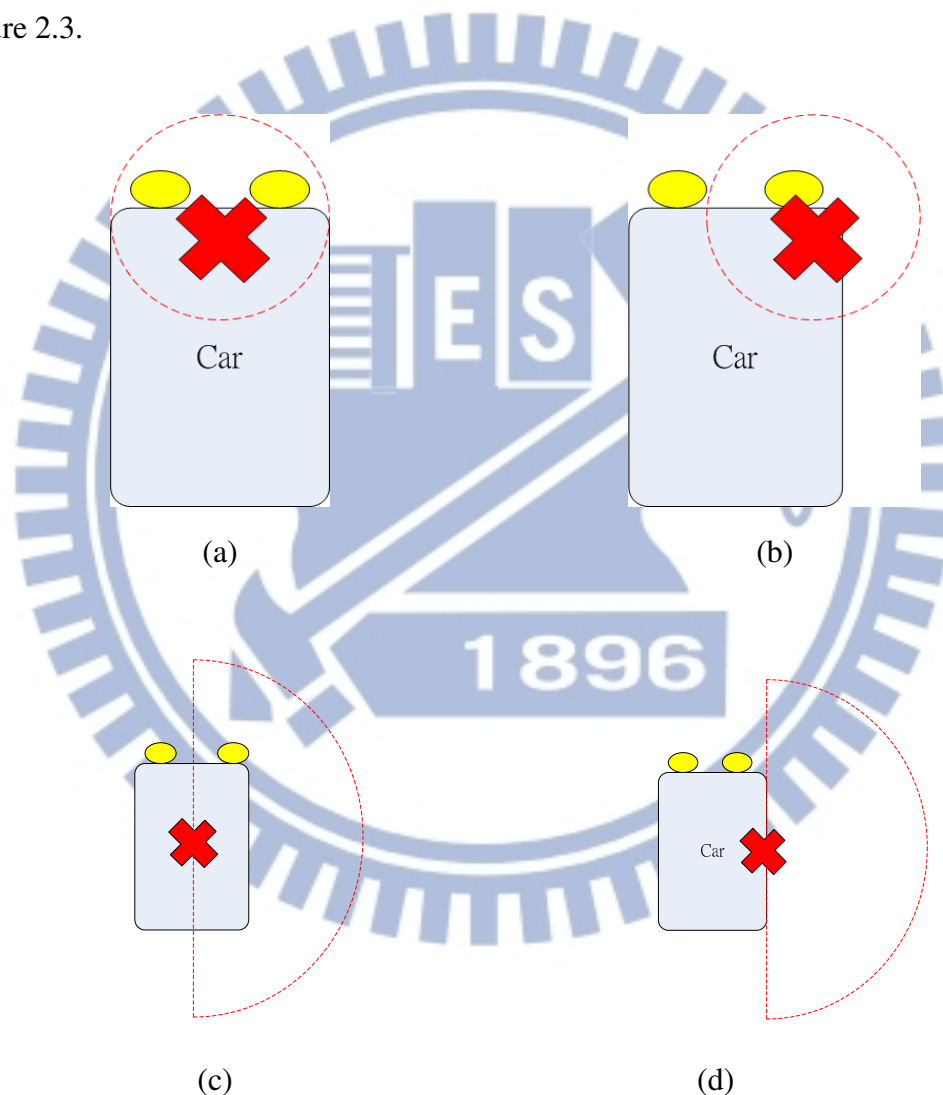


Figure 2.2 Positions of cameras affixed to the video surveillance vehicle roof and the corresponding FOV. (a) The omni-camera is affixed at the rear-middle of the car roof. (b) The omni-camera is affixed at the right-front of the car roof. (c) The PTZ-camera is affixed at the middle of the car roof. (d) The PTZ-camera is affixed at the border of the car roof.

Furthermore, we analyze the motion vectors yielded by an adopted optical flow

method in acquired omni-images so that we can estimate the vehicle's moving direction when meeting a branching road. In this way, we will not get lost on any path in the guidance map by keeping track of the graph nodes in the guidance map.

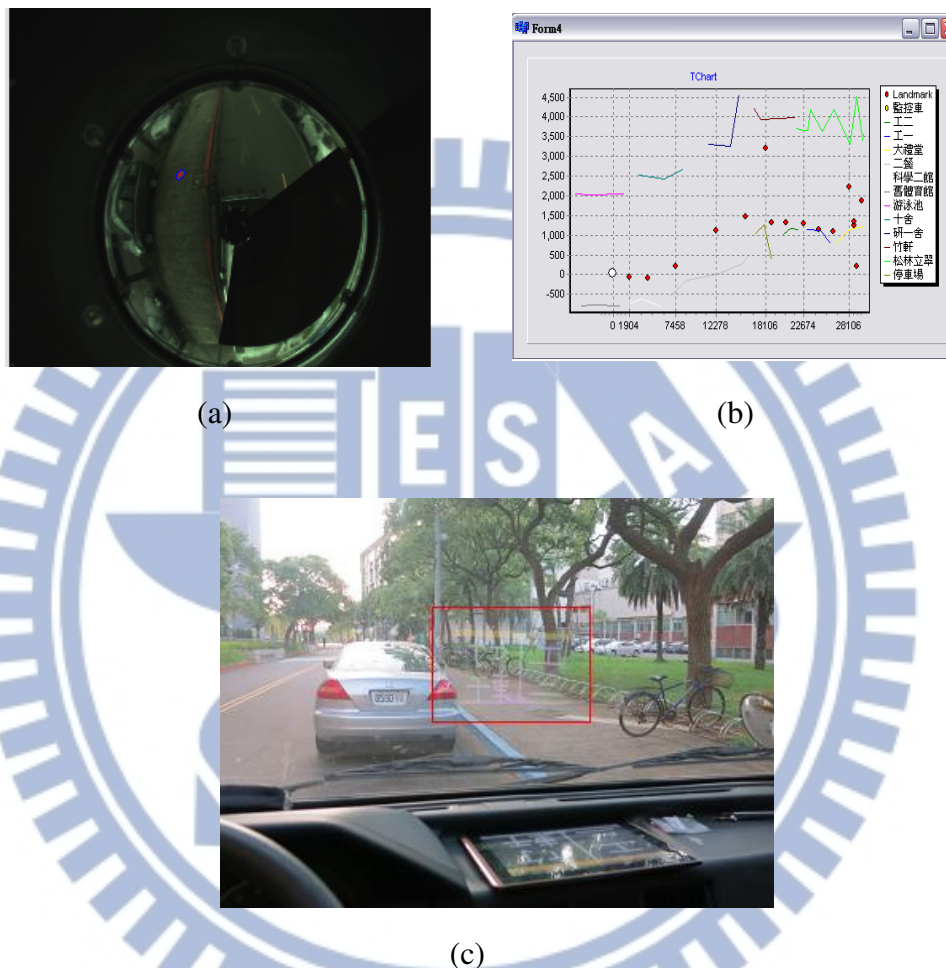


Figure 2.3 An example of landmark detection and guidance map. (a) An omni-image of a landmark detected at a sidewalk. (b) A generated guidance map showing the relative position of the car. (c) A projected image on the windshield.

2.2 System Configuration

The proposed video surveillance system will be described elaborately in this section. The description will be separated into three parts: hardware configuration, software configuration, and network configuration. The hardware includes: 1) a video surveillance vehicle, 2) a two-camera omni-directional imaging device and a

PTZ-camera device, and 3) two laptop computers and a pad. The software includes: 1) a program used to integrate the vision-based system, 2) the drivers of the omni-cameras and the PTZ-camera, and 3) the program developed by the ARTRAY Company which is a provider of CCD cameras. The two-camera omni-directional imaging device and the PTZ-camera are controlled by the laptop computer, and the pad has to receive information sent by a laptop computer, so we construct a local network to handle the task of communication among all the equipments.

2.2.1 Hardware configuration

The surveillance vehicle, named Delica, is made by Mitsubishi Co. It is a 469cm ×169cm×196cm vehicle with a working table and a power supply. System operators may sit inside the surveillance vehicle to operate the laptop computers and monitor the entire surrounding environment. Moreover, a steel frame is affixed to the car roof, on which the omni-image device and the PTZ camera are affixed. And two extension USB cords and a cross-over cable crossing the video surveillance vehicle were added to facilitate transmitting images captured with the omni-imaging device and the PTZ-camera. Detailed descriptions of the functions of the imaging devices will be given in Sections 2.3 and 2.4. The entire video surveillance system is shown in Fig. 2.4.

In order to control the entire guidance system, we use two laptop computers and a pad as control units, with the laptops handling the omni-imaging device and the PTZ-camera. Both laptops are produced by TOSHIBA Computer Inc. The pad, named Eee Pad Transformer, is produced by ASUS Computer Inc. We simulate a head-up display device by projecting images appearing on the pad onto the windshield of the vehicle. Detailed specifications of these devices are listed in Table 2.1.

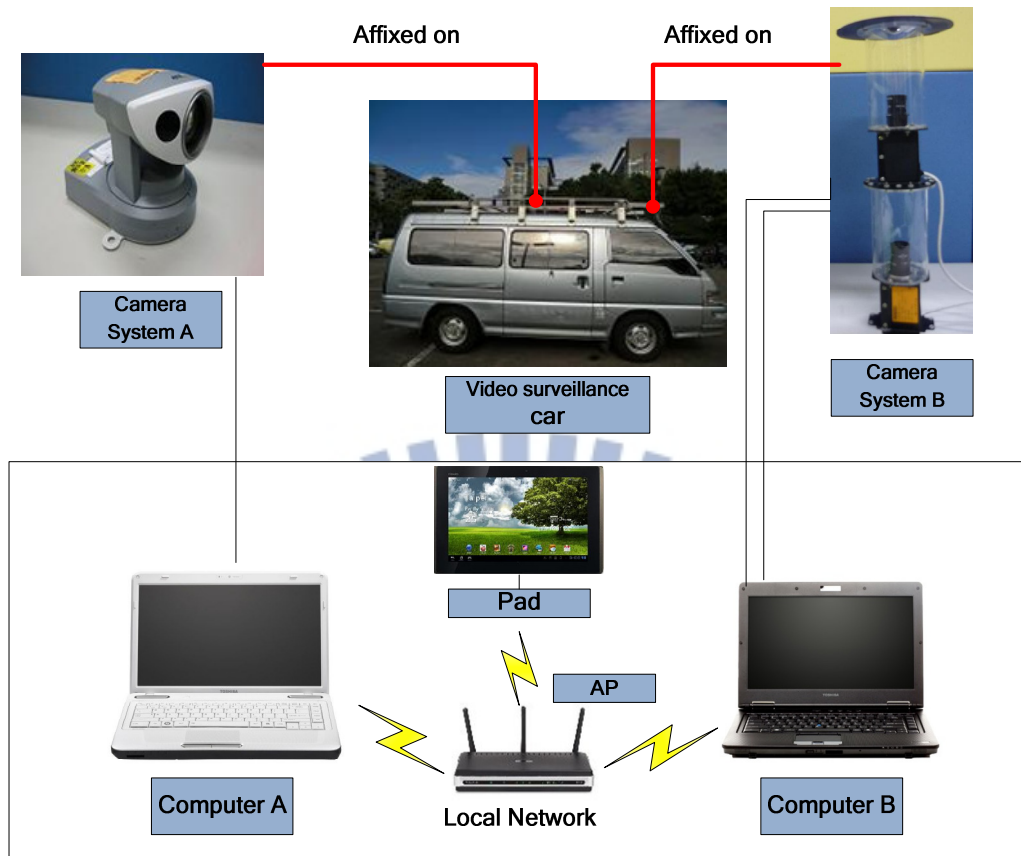


Figure 2.4 Structure of the proposed surveillance system.

Table 2.1 Specifications of the laptop computers and the pad used in this study.

	Tecra M11	Satellite A660	Eee Pad Transformer
CPU	Intel Core i7-620M 2.66/3.33GHz	Intel Core i5-480M 2.66/2.93GHz	NVIDIA Tegra2.1.0 GHz
RAM	4G DDR3 1066MHz	2G DDR3 1066MHz	1GB
GPU	nVidia NVS 2100M	ATI HD5650	none
Network	Gigabit LAN	Fast Ethernet LAN	WLAN 802.11 b/g/n 2.4GHz

To exchange commands and information of guidance between the two laptops and the pad, we use an access point (AP) to connect them and set up a local network for communication.

2.2.2 Software configuration

We use a Borland C++ Builder (BCB) V6 as the development platform to build our guidance system. The BCB is a program development tool for the operating system of Windows by which we can create a graphic user interface (GUI) conveniently and quickly. The programming language we use is C++. It is a widely used language. One of the laptops, the Tecra M11 computer, uses the operating system of Windows 7, and the other, Satellite A660, uses Windows XP.

The operating system of the pad is Android 3.2, and we develop the applications by the use of the Eclipse. However, we need to install the JAVA development tool (JDK) and Android development tool (ADT) in the Eclipse, so that we can develop Android applications in this environment.

In order to use the camera devices, we have to install the drivers of the ARTCAM-200SS cameras and ARIA into the laptops. The camera company also provides corresponding software development kits (SDKs), in addition to, we can use simple source codes to know the purpose of call functions in the program. Accordingly, we can adjust the parameters of each camera, such as the value of exposure or the global color gain, through the SDK. Moreover, the camera company not only provides the BCB but also the C, VB.NET or C#.NET to the programmers.

2.2.3 Network Configuration

A network configuration is needed for communication between the two laptop computers and the pad because two omni-images are acquired from the two-camera omni-directional imaging device and the PTZ-camera, and each imaging device is processed by a laptop, respectively. Moreover, the laptops also send data to the pad. As a result, to communicate between the two laptops and the pad, we set up a local area network.

As shown in Fig. 2.5, the access point (AP) can provide a wireless environment; and the devices can be connected to one another through the AP. The laptop computer COM_A can be used to create the guidance map and guide the navigation. It can be used to acquire not only the data of the two omni-images by itself but also the data of the *PTZ image* (the image taken by the PTZ camera). On the other hand, the laptop computer COM_B needs to receive these images from COM_A through the local network. Afterwards, the pad can be used to display the names of buildings. Using the local network, COM_A can send the information of the current buildings to the pad.

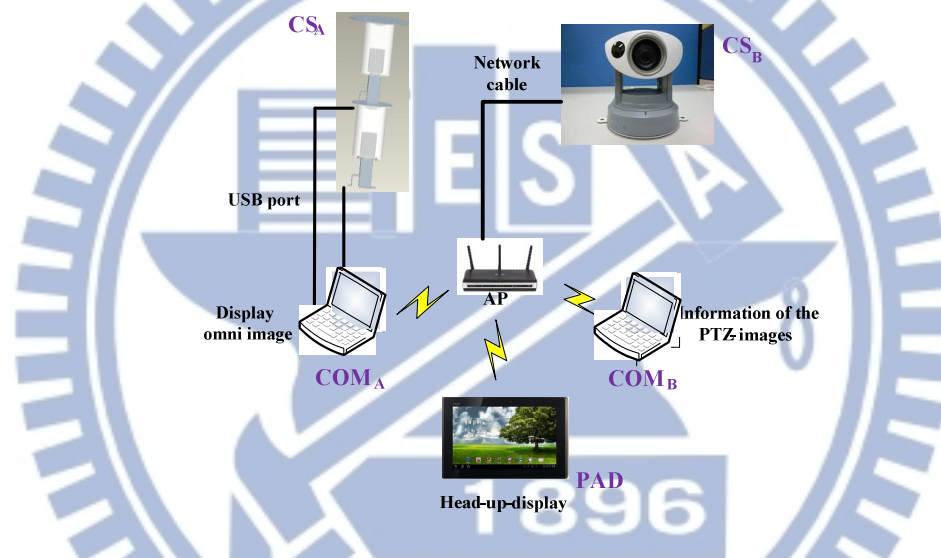


Figure 2.5 The architecture of the local network used in this study.

2.3 Review of Adopted PTZ-camera System

In this section, we review the adopted PTZ-camera system with panning, tilting, and zooming capabilities. In this study we use an AXIS 213 PTZ camera made by AXIS Inc. as shown in Figure 2.6. 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 is of the resolution of 320×240 pixels.



Figure 2.6 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. (d) A back view of the camera.

2.4 Review of Adopted Omni-camera System

In this section, we review the adopted omni-camera system which includes two lenses of model LV0612H, two CMOS cameras of model ARTCAM-200SO, and two

CMOS cameras of model ARTCAM-200MI. Table 2.2 lists the specifications of the COMS cameras.

Table 2.2 Specifications of used COMS cameras.

	ARTCAM-200MI
Resolution	2.0 M pixels(1600*1200)
Dimension	33mm × 33mm × 50mm
CMOS sensor size	1/2" (6.4×4.8mm)
Mount	C-mount
Frame per second	5 fps
Direct show camera	No

To produce an omni-camera, we need to combine a projective CCD camera and a hyperbolic-shaped mirror together. The parameters of each of the hyperbolic-shaped mirrors are described here. The radius r of the hyperbolic-shape mirror is 4cm, the focal length f of the projective camera is 6 mm, and the sensor width S_w of the camera is 2.4mm. Also, the axis of the camera is aligned with the axis of the center point of the hyperbolic-shape mirror.

As shown in Fig. 2.7, by the principle of similar triangles, the distance d between the optical center and the mirror center can be computed by the following equation:

$$\frac{d}{r} = \frac{f}{S_w} \quad (2.1)$$

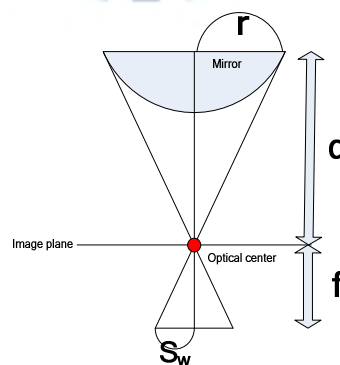


Figure 2.7 Relationship of the mirror and the CMOS sensor in camera.

Also, as shown in Fig. 2.7(a), the hyperbolic-shape of the mirror in the camera coordinate system may be described as:

$$\frac{R^2}{a^2} - \frac{Z^2}{b^2} = -1, \quad R = \sqrt{X^2 + Y^2}. \quad (2.2)$$

The coordinates (X, Y, Z) specify a point P in the world coordinate system (WCS). Let the projection of P into the image plane of the camera be the point p with image coordinates (u, v) in the image coordinate system (ICS). In order to get the parameters a and b of the hyperbolic-shape of the mirror, first we have to acquire the elevation angle α in Figure 2.7(a) from the relation between the camera coordinate system (CCS) and the ICS according to Wu and Tsai [7] as follows:

$$\tan \alpha = \frac{(b^2 + c^2) \sin \beta - 2bc}{(b^2 - c^2) \cos \beta} \quad (2.3)$$

where β is the azimuth angle as shown in Fig. 2.8(a). To compute it, let the distance from the origin O of the camera coordinate system shown in Fig. 2.8 to the mirror center O_m be denoted as c , and let that from the lens center O_c of the camera to O_m be denoted as d which may be measured in advance. Then, we can compute c by the simple formula $d = 2c$ because O is defined to be at the middle point between O_m and O_c . Accordingly, in Eq. (2.3), let the omni-camera have the largest FOV, and the incidence angle α be set 0. Then, the angle θ and thereby β , according to Fig. 2.8(a), can be computed as follows:

$$\begin{aligned} \theta &= \tan^{-1} \frac{r}{2c}, \\ \beta &= \frac{\pi}{2} - \theta \end{aligned} \quad (2.4)$$

where r is the radius of the circular area of the base of the mirror. Using Eq. (2.4), the parameter b can be obtained by solving Eq. (2.3). Finally, the parameter a is derived from the following equation:

Therefore, we propose a learning strategy. As shown in Figure 2.9, the laptop COM_A is used to analyze acquired omni-images of the surrounding environment and compute the position of the landmark in the omni-image through an ellipse fitting method. In addition, the laptop COM_B is used to capture PTZ-images of neighboring buildings, and then we choose feature points of the buildings manually and use compute their distances using the pano-table. Furthermore, through the angular mapping method, we can get the angels of the feature points. Afterwards, the distance and orientation data of the feature points are sent to another laptop COM_A for creating the desired local map.

A local map only includes a landmark. When we generate the local maps for all landmarks, the laptop COM_B does not need to send the information of the feature points to the laptop COM_A . Our local maps are independent; therefore, we propose a method for quickly converting the local maps into a global map, which is described in Chapter 3.

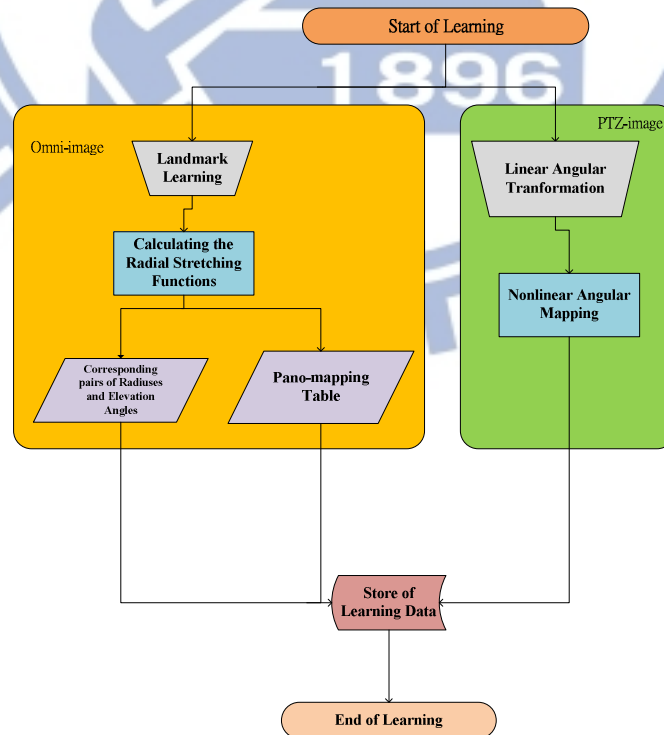


Figure 2.9 Flowchart of calibration of omni-cameras and PTZ-camera.

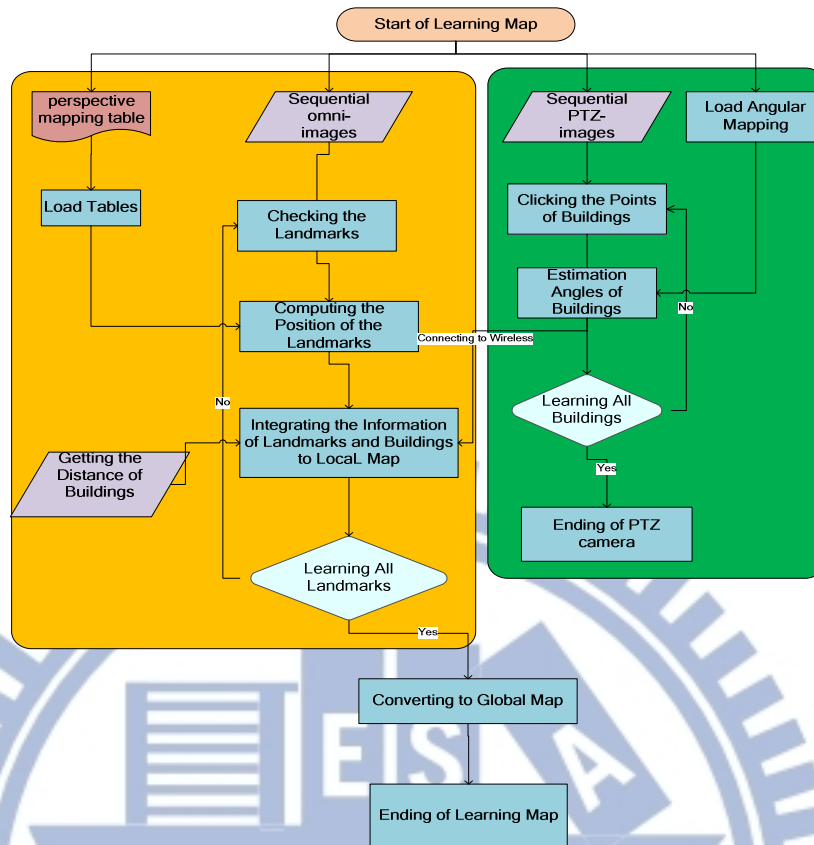


Figure 2.10 Flowchart of learning guidance map.

2.5.2 Navigation Process

The second part of our guidance system is the navigation process. In Section 2.5.1, we mentioned how we create a guidance map through the learning process. Accordingly, we can estimate the position of the video surveillance car on the guidance map and implement our tour guidance system in the navigation process. First of all, we use the captured omni-images to detect the landmarks at the sidewalk. Then, by using the pano-mapping table, the 3D information of the landmarks can be estimated. The process will be introduced elaborately in Chapter 4. Next, the turning direction of the surveillance vehicle is checked by an optical flow method. We use the turning direction to keep track the current node corresponding to the vehicle on the graph of the guidance map. The

detailed process will be introduced in Chapter 6. After the laptop COM_A receives the position of the currently-visited landmark and the vehicle turning direction, we integrate these data to obtain the position of the video surveillance vehicle and get the information of the neighboring buildings. Finally, we need to send the names of buildings to the pad. For this, at first through the AP, we can use the local wireless network to connect the laptop COM_A and the pad. The laptop COM_A then sends the names of the nearby buildings to the pad. Then, we display the names of the buildings as an image shown on the pad and project the image onto the car windshield. Moreover, we propose a method to estimate the position of the projected images on the windshield. The user can use the method to place the pad conveniently. The detailed process will be introduced in Chapter 5. As shown in Figure 2.11, we can know the flowchart of navigation process.

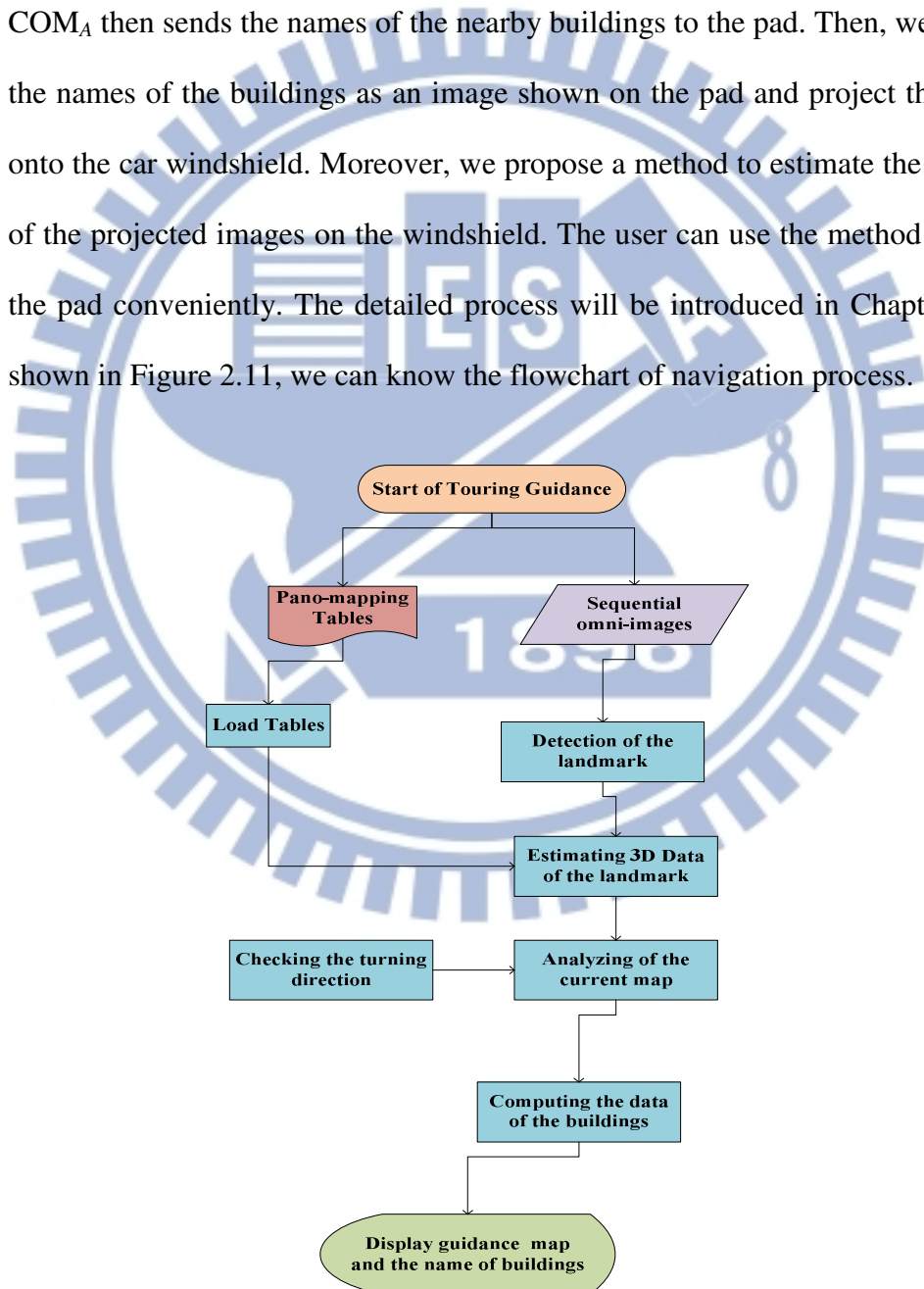


Figure 2.11 Flowchart of tour guidance

Chapter 3

Creation of Guidance Map

3.1 Introduction

In this chapter, we describe the details of the method we propose to generate the guidance map for use in the proposed augmented reality based tour guidance. Using the omni-camera device and the PTZ-camera device affixed to the roof of the video surveillance vehicle, we can create the guidance map quickly.

The proposed guidance map includes the center points of the landmarks and some feature points of objects of interests in the guidance area. We can use prominent points on any objects, like buildings, lamps, chairs, etc., as feature points. However, in this study, we just choose the corner points of buildings as the feature points because buildings are the most obvious objects in the guidance area and we can get prominent feature points of buildings conveniently in images.

In general, we use a ruler and a protractor manually the distance between the PTZ-camera and every feature point, as well as the orientation of the camera direction with respect to the line from the PTZ-camera to the feature point. On the other hand, in this study we use the omni-camera to detect and estimate the location of each landmark automatically, which is defined to the position of the center of the landmark. The detail will be described in Chapter 4.

In addition, Wang and Tsai [9] proposed a method for angular mapping for camera calibration to compute the angular information of points in the PTZ-image. The PTZ-camera can be used to estimate the directions of feature points by the method. However, we have to combine the information of the landmark with that of the feature points. Therefore, we propose a method to convert PTZ-camera

coordinates into omni-camera coordinates. The detail will be described in Sections 3.2 and 3.3.2.

By the aforementioned techniques, local maps of landmarks can be created. In Section 3.3.2, we describe a graphical user interface (GUI) which we design in this study for a user to create the local map of every landmark. Moreover, a method for combining multiple local maps into a single global map is also proposed in this study, which we describe in Section 3.3.3.

3.2 Integration of Information Acquired by PTZ-camera and Omni-cameras

3.2.1 Review of Adopted Calibration Method for PTZ-camera

In order to get the orientation of every feature point with respect to the PTZ-camera direction, Wang and Tsai [9] proposed a nonlinear angular mapping method to conduct angular transformations between real-world space points and acquired PTZ image points. The method is adopted for use in this study. It is reviewed here at first.

First of all, we introduce the involved coordinate systems, as shown in Figure 3.1. The image coordinate system (ICS) defined for PTZ-images is described by image coordinates (u, v) , and the spherical coordinate system (SCS) defined for the real-world space is described by the space coordinates (ρ, θ, φ) . The SCS is a 3D polar coordinate system whose coordinates can be computed in terms of those of a Cartesian coordinate system (CCS) described by (i, j, k) . The origin S of the SCS is

also the origin of the CCS, and the ij -plane of the CCS is parallel to the uv -plane of the ICS. Reversely, the coordinates (i, j, k) of the CCS can be represented by the coordinates (ρ, θ, φ) of the SCS as well. Specifically, as illustrated in Fig. 3.1, a space point P with coordinates (i, j, k) in the CCS may be transformed into a corresponding one in the SCS with coordinates (ρ, θ, φ) , where ρ is the distance between point P and the origin S of the SCS; θ is the longitude angle between the positive k -axis and the line from the origin S to point P projected onto the ik -plane, and φ is the latitude angle between the ik -plane and the line from the origin S to point P .

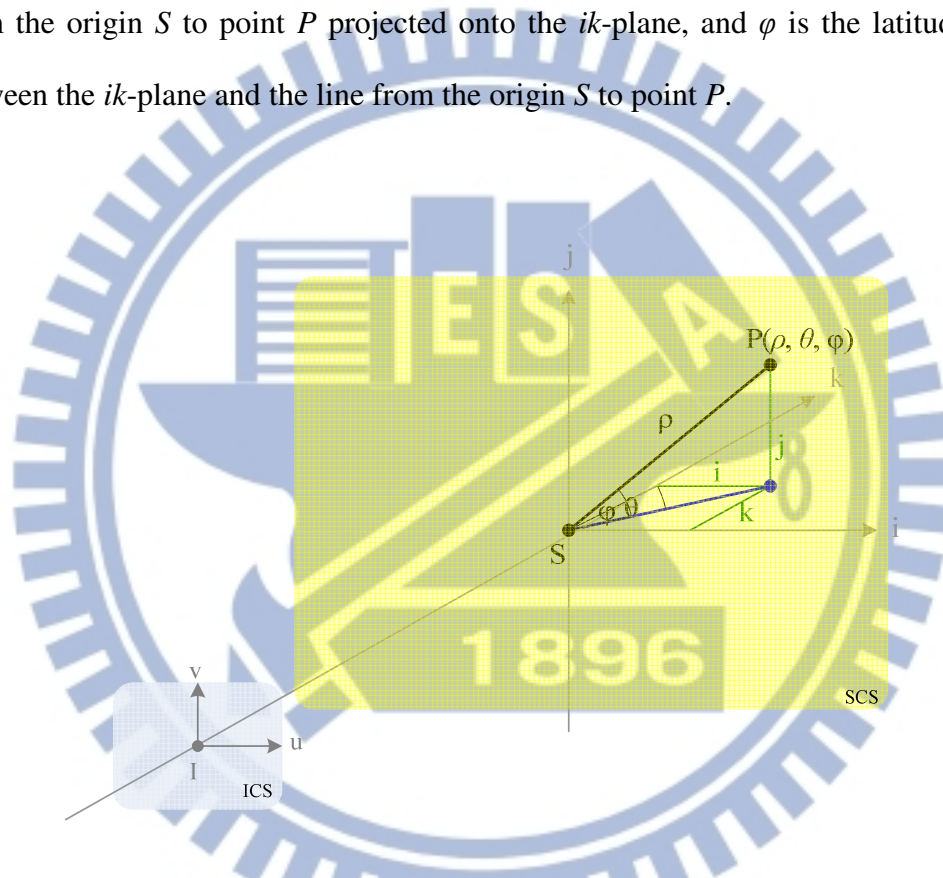


Figure 3.1 An illustration of transformation between image coordinate system (ICS) and spherical coordinate system (SCS) and Cartesian coordinate system.

In order to calibrate the PTZ camera for the purpose of computing the longitude and latitude angles of space points which appear in the acquired PTZ-image, a grid board is used in this study. It has m vertical lines and n horizontal lines, and is attached on a wall which is perpendicular to the ground. During the calibration process, we measure the angles between the PTZ-camera direction and the lines from

the camera to the intersection points in the grid board, so the longitude and latitude angle values of all the intersection points can be known in advance. Then, we use an interpolation method to compute the orientation θ of each non-intersection point p in the PTZ-image with respect to the PTZ camera direction. This orientation θ is just the angle between the PTZ-camera direction and the line from the position of the PTZ-camera to p in the PTZ-image. In this way, the longitude and latitude values of each point p in the PTZ-image can be obtained. In Figure 3.2, we can see the respective positions of the grid board and the camera used in this method. And two image views of the board from the camera are shown in Figure 3.3. The intersections of the lines are marked by yellow points.

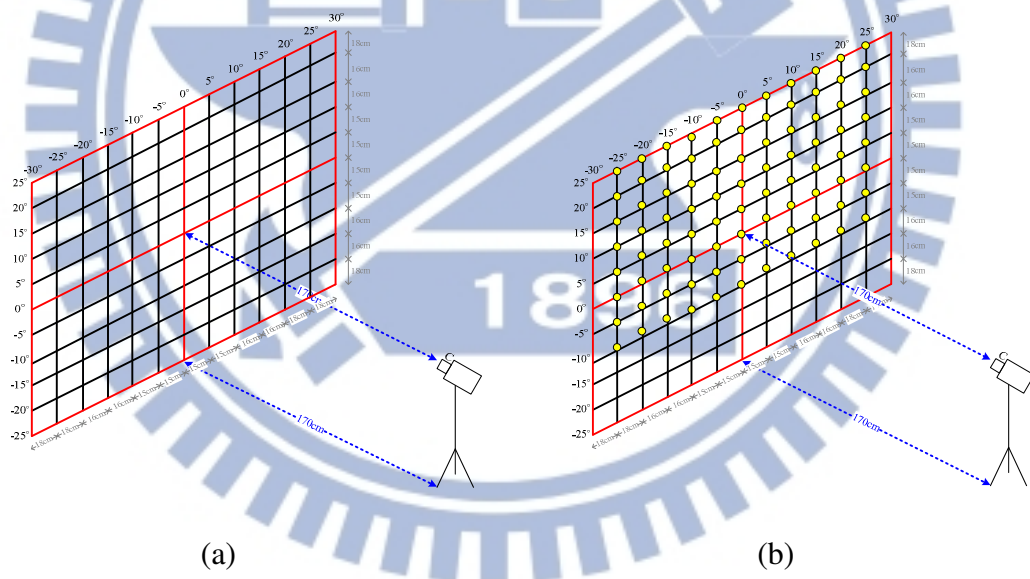


Figure 3.2 Camera calibrations by a vertical grid board. (a) An illustration of attaching the lines on the wall. (b) The intersections seen by camera are marked by yellow points.

After knowing the longitude and latitude values of the non-intersection points in the PTZ-image by nonlinear angular mapping, the PTZ-camera calibration work is completed. These longitude and the latitude values of every point in the image is

finally saved into a bmp image, like that shown in Figure 3.3(c).

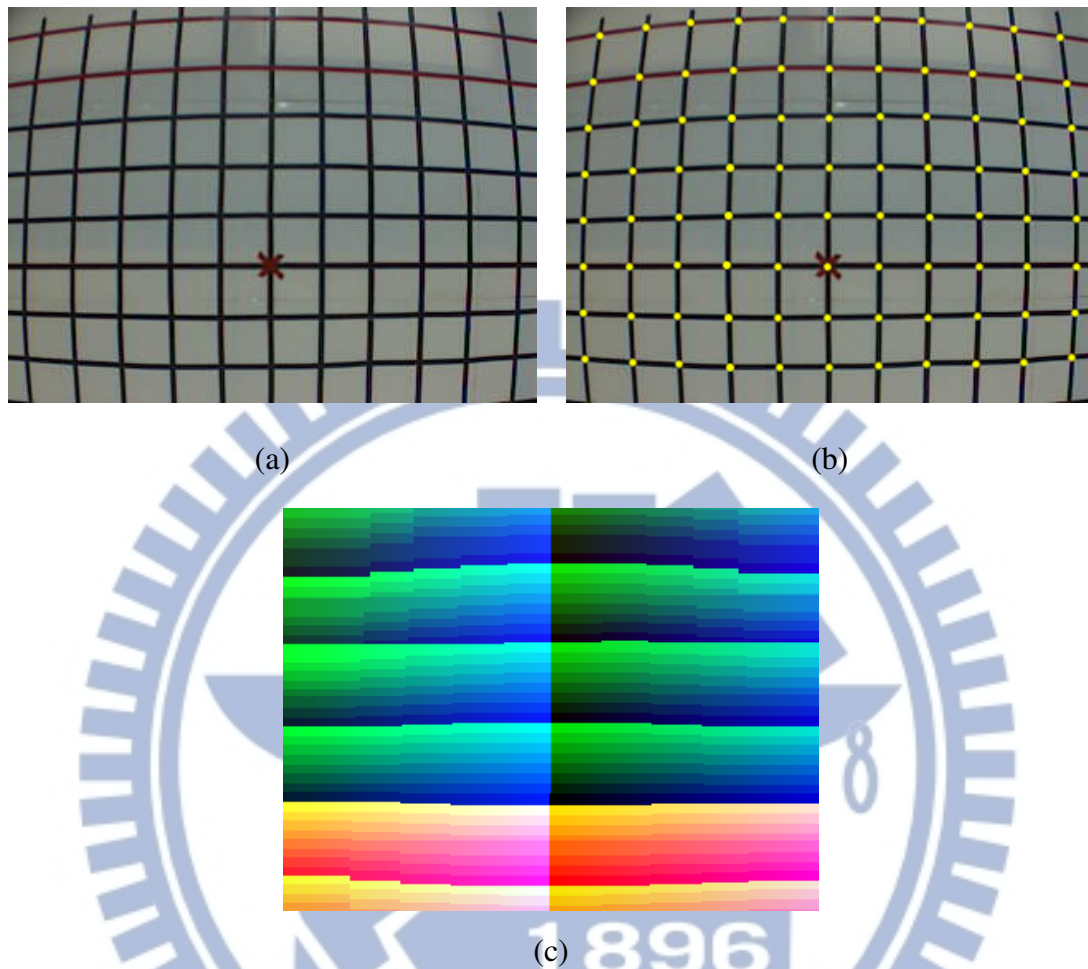


Figure 3.3 Camera views of Figure 3.2. (a) View of Figure 3.2(a). (b) View of Figure 3.2(b). (c) The bmp image with camera calibration information.

3.2.2 Conversion of PTZ-camera Coordinates into Omni-camera Coordinates

The omni-camera coordinate system (OCCS) is regarded as the world coordinate system (WCS) with coordinates (x, y, z) , and the PTZ-camera coordinate system (PCCS) as the SCS, in this study. However, we also have the PTZ-camera in the WCS; therefore, in order to use the information of the PTZ-image coordinates integrally, we have to transform these coordinates into the OCCS. In this section, we propose a

scheme to convert the PTZ-camera coordinates into coordinates in the OCCS. In this way, we can get the relative position between the omni-directional imaging device and the PTZ-camera device easily.

In more detail, we use Figure 3.4 to show an illustration of the layout of the video surveillance car roof where the distance between the omni-camera and the PTZ-camera is obtained by manual measurement. The red circles in the layout are the positions of the camera devices. Accordingly, if the front omni-camera device is assumed to be located at the origin of the WCS, then the position of the PTZ-camera device is $(223, 0, 0)$. That is, the offset between the two devices is $(223, 0, 0)$.

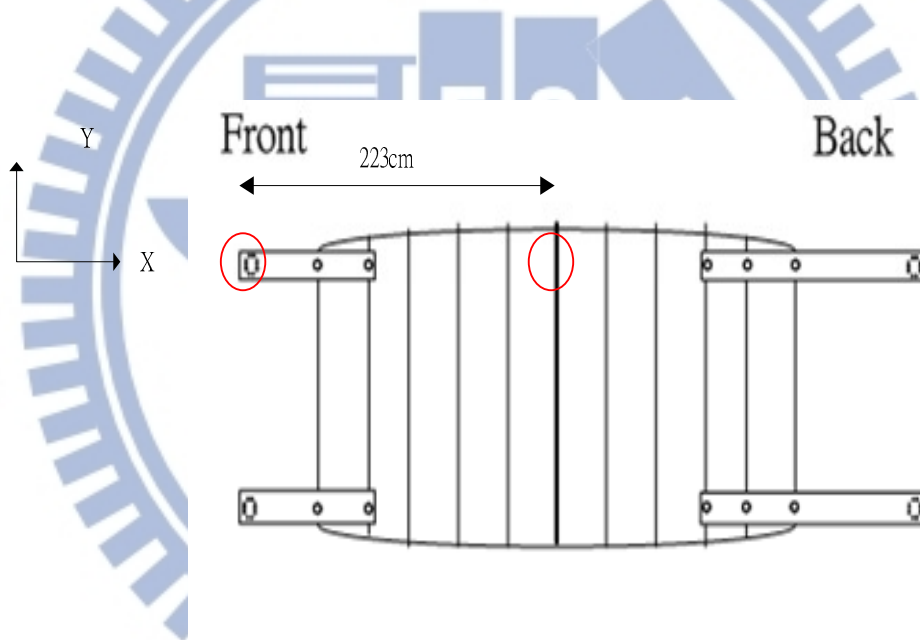


Figure 3.4 an illustration of the layout of the video surveillance car roof.

Furthermore, as shown in Figure 3.5, the *practical* field of view of the omni-camera is limited. Nevertheless, even if a point P is too far from the practical field of view of the omni-camera, we can still get it by the PTZ-camera. Therefore, we have to convert the coordinates of the PCCS into corresponding ones of the OCCS. The detail is described as follows.

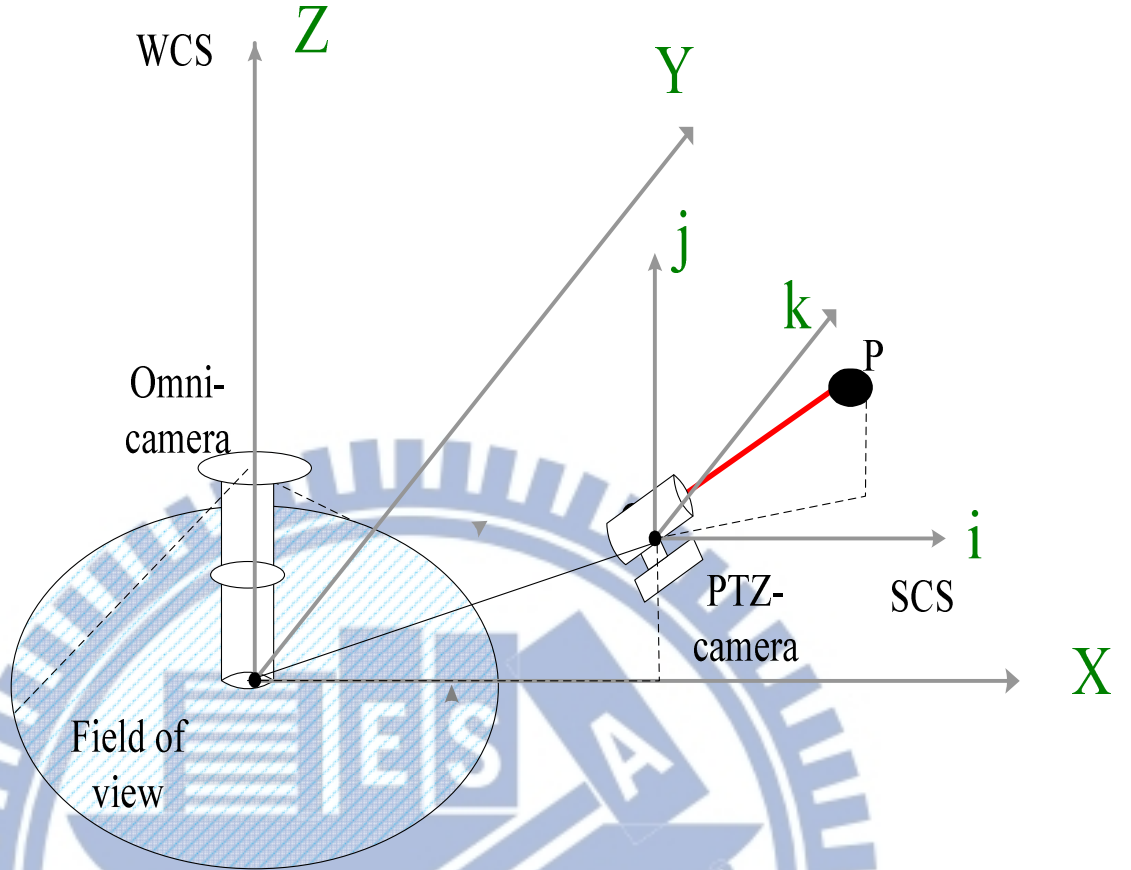


Figure 3.5 The WCS used in this study which includes the SCS.

As shown in Figure 3.6, we can acquire the tilt angle φ_c and the pan angle θ_c from the PTZ-camera device. The value θ_c is the angle between the camera direction and the direction of the y -axis, and the value φ_c represents the vertical tilting angle of the camera. The tilt angle φ and the pan angle θ of a detected space point in a PTZ-image can be acquired through the calibration method mentioned previously. The distance D between the PTZ-camera and the detected point can be obtained by manual measurement, and the distance ρ is the horizontal distance between the PTZ-camera and the detected point. Finally, we want to estimate the world coordinates (x, y, z) , and the process we propose in this study is described as an algorithm in the following. Refer to Fig. 3.5 for the notations used in the algorithm.

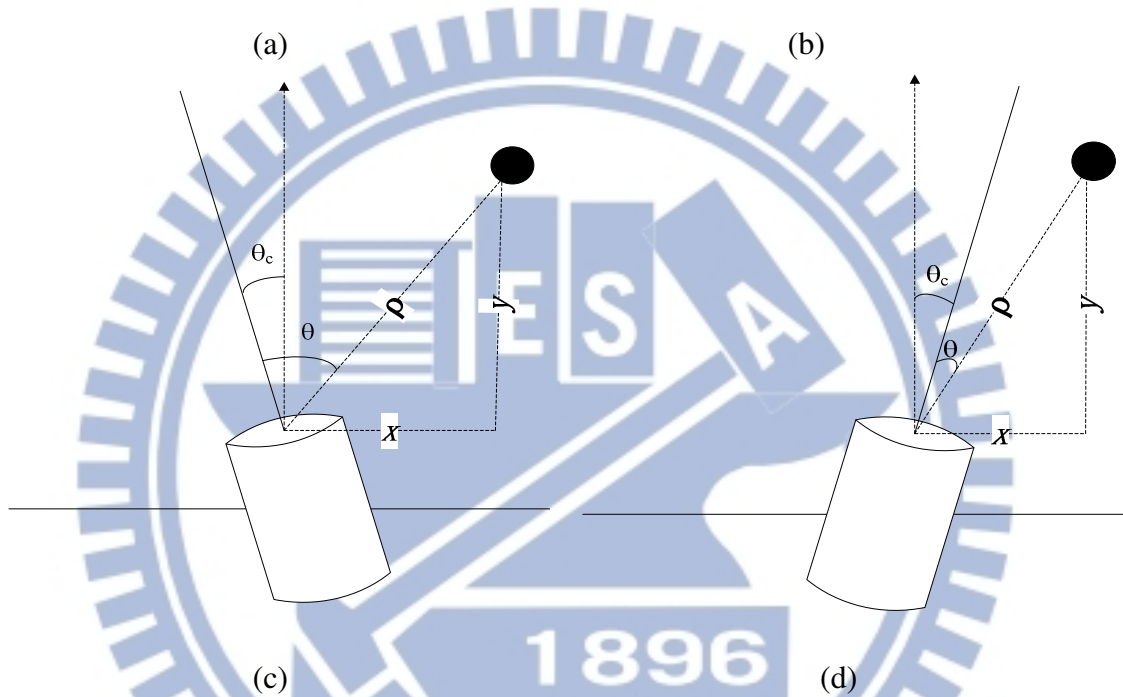
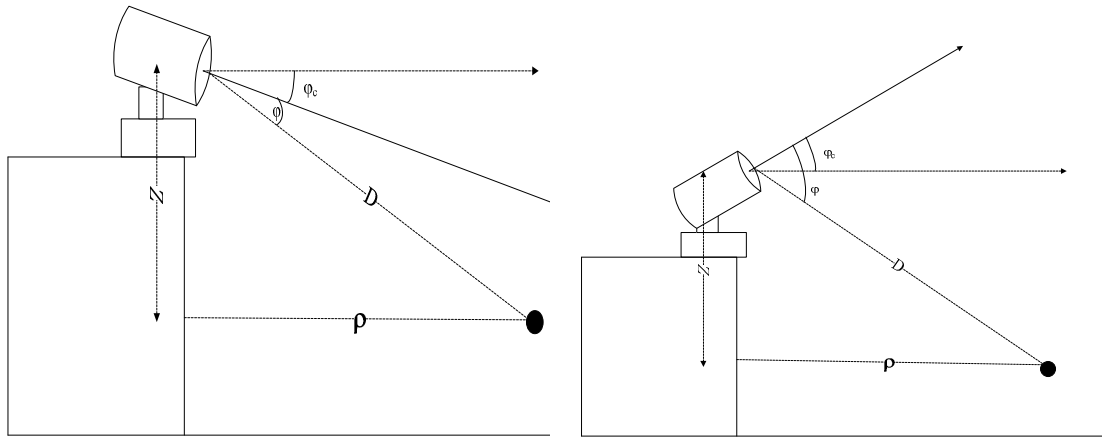


Figure 3.6 the angle between the detected point and the PTZ-camera (a) the PTZ-camera has a tilt angle of $\varphi_c < 0$ (b) the PTZ-camera has a tilt angle of $\varphi_c > 0$ (c) the PTZ-camera has a pan angle of $\theta_c < 0$ (d). the PTZ-camera has a pan angle of $\theta_c > 0$

Algorithm 3.1: computing the world coordinates of a feature point.

Input: A feature point p captured by the PTZ-image and its distance D to the PTZ-camera which is obtained manually in advance.

Output: The world coordinates (x, y, z) of the feature point in the PTZ-image.

Step 1. Calculation of distance ρ and height z ---

Using the tilt angles φ_c and φ , we can compute the distance ρ and the height z in the following way:

(a) if the $\varphi_c < 0$ as shown in Figure 3.5(a), compute ρ and z by:

$$\rho = D \times \cos(\varphi_c + \varphi); \quad (3.1)$$

$$z = D \times \sin(\varphi_c + \varphi); \quad (3.2)$$

(b) if the $\varphi_c > 0$ as shown in Figure 3.5(b), compute ρ and Z by:

$$\rho = D \times \cos(\varphi_c - \varphi); \quad (3.3)$$

$$z = D \times \sin(\varphi_c - \varphi). \quad (3.4)$$

Step 2. Calculation of x and y in the world coordinate system ---

Use the distance ρ and the pan angles θ_c and θ , compute the world coordinates x and y as follows:

(a) if $\theta_c < 0$ as shown in Figure 3.5(c), compute x and y by:

$$x = \rho \times \sin(\theta_c - \theta) + L; \quad (3.5)$$

$$y = \rho \times \cos(\theta_c - \theta) + K; \quad (3.6)$$

(b) if the $\theta_c > 0$ as shown in Figure 3.5(d), compute x and y by:

$$x = \rho \times \sin(\theta_c + \theta) + L; \quad (3.7)$$

$$y = \rho \times \cos(\theta_c + \theta) + K, \quad (3.8)$$

where the parameter L is the position of the PTZ-camera on the x -axis of the WCS and K is the position of the PTZ-camera on the y -axis of the WCS, both known in advance by manual measurement.

Using the above algorithm, we can get the world coordinates (x, y, z) of each feature point appearing in the captured PTZ-image.

3.3 Creation of Guidance Map

3.3.1 Creation of Database for Guidance Map

The guidance map is important to the proposed augmented reality based guidance system for use in tour guide driving. We have to navigate to visit the sites in the guidance area precisely according to the guidance map, and monitor the entire top-view guidance environment. A display of the guidance map proposed in this study is designed to include various data, such as:

- (1) the center points of landmarks which are labeled on the guidance map;
- (2) the feature points in the guidance area;
- (3) the current position of the video surveillance car.

Landmarks are regarded as *navigation points* in this study. In order to recognize the different landmarks, every landmark is assigned a label. We can use the labels of landmarks to check the correct positions of detected landmarks and navigate in the guidance area correctly. In the guidance map, we select a landmark to be the start point of a guidance tour with the *start label* of 0.

On the other hand, feature points can be anything theoretically. However, in our guidance system, we want to get the information of nearby buildings. So we choose edge points of buildings to be our feature points, as mentioned previously. A user can input the names of the selected feature points to the proposed system. Moreover, we have designed a “learning GUI” for the user, which will be described in Section 3.3.2. In addition, by using the camera devices, the position of a landmark can be computed. Furthermore, we measure the distance between the PTZ-camera and each feature point by hand, also mentioned previously. Accordingly, the position of feature points on the WCS can be obtained by Algorithm 3.1. Finally, the location of the video

surveillance vehicle can also be marked on the guidance map. In this way, we can always check the position of the video surveillance vehicle and update its corresponding point on the guidance map for display and observation. An illustration of the guidance map is shown in Figure 3.7.

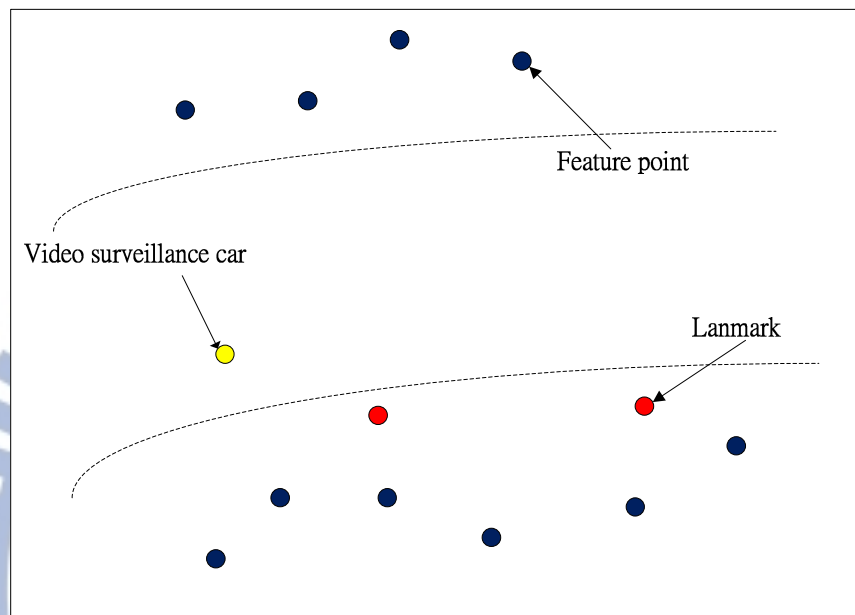


Figure 3.7 Illustration of proposed guidance map.

3.3.2 Creation of Guidance Map with PTZ-camera and Omni-camera

In order to create our guidance map, we have to generate local maps at first. A local map includes a landmark and a number of related feature points, with the landmark being defined as the origin on the local map. So, for each landmark, a local map is created. Also, we record the label of the landmark, and the name and the coordinates of every feature point chosen. On the other hand, a local network has been set up in the surveillance vehicle as mentioned previously, which integrates the two laptops for between-computer communication. The structure of the network is illustrated in Fig. 3.8. Through this network, we can use the two camera devices to get the landmark coordinates on the WCS and the coordinates of every feature point on

the local map. The detail of this process is described as an algorithm below.

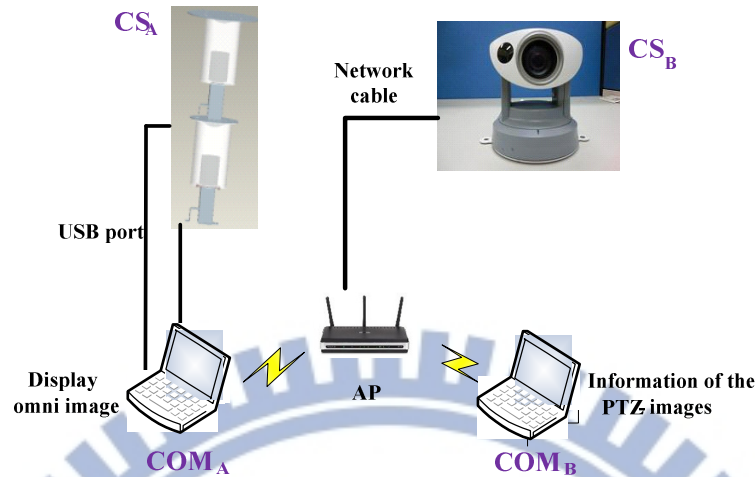


Figure 3.8 Structure of the communication between two laptops used in this study.

Algorithm 3.2: *computing the positions of a landmark and related feature points on the local map using images acquired with the two camera devices.*

Input: two omni-images I_1 and I_2 acquired by the omni-imaging devices and a perspective-view image I captured by the PTZ-camera.

Output: the coordinates of a landmark P_{lm} and those of the feature points selected of nearby buildings on a local map with respect to P_{lm} .

Steps.

- Step 1. Detect the image points of landmark P_{lm} appearing in I_1 and I_2 and compute accordingly the coordinates (x_1, y_1, z_1) of the center point of the landmark using a detection method described in Chapter 4.
- Step 2. Click a feature point P in the PTZ-image, and measure the distance d between P and the PTZ-camera by hand.
- Step 3. According to the discussions in Section 3.2.2, compute the world coordinates (x_2, y_2, z_2) of the feature point P in the SCS of the PTZ-camera in the following way:

$$x_2 = d \times \sin(\varphi_c \pm \varphi) \times \sin(\theta_c \pm \theta) + L; \quad (3.9)$$

$$y_2 = d \times \cos(\varphi_c \pm \varphi) \times \cos(\theta_c \pm \theta) + K; \quad (3.10)$$

$$z_2 = d \times \sin(\theta_c \pm \theta). \quad (3.11)$$

Step 4. Compute the coordinates (x_3, y_3) of P on the local map with respect to the landmark P_{lm} by

$$x_3 = x_2 - x_1; \quad (3.12)$$

$$y_3 = y_2 - y_1. \quad (3.13)$$

By Algorithm 3.2, we can compute the coordinates (x_1, y_1) of a landmark P_{lm} and the coordinates (x_3, y_3) of every related feature point in a local map. We record these computed data in the local map of P_{lm} .

3.3.3 Conversion of Local Maps into a Global Map

The global map is used for tour guidance in this study and it is designed to integrate all the local maps. Moreover, as shown in Figure 3.9, when we create the local maps, the last feature point of the local map and the first feature point of the next local map are taken to be an identical feature point, and this overlapping point can be connected to merge the two local maps. Moreover, we define the landmark on the first local map as the origin point on the global map. We divide the process of converting all the local maps into a global map into four steps, as described in the following algorithm.

Algorithm 3.4: *Integrate all landmarks with the global map.*

Input: a local map set $L = \{L_1, L_2, \dots, L_n\}$

Output: a global map G .

Steps.

Step 1. Translate the coordinates (x_0, y_0) of the landmark of local map L_1 into coordinates $(0, 0)$ of the desired global map G .

Step 2. Translate the coordinates (x_m, y_m) of every feature point p_m in local map L_1 into coordinates (x_m', y_m') in global map G in the following way, where the value m ranges from 0 to the number of feature points in L_1 :

$$x_m' = x_m - x_0;$$

$$y_m' = y_m - y_0.$$

Step 3. Merge local maps L_2 through L_n by the following steps with $i = 1$ initially.

3.1 Record the coordinates (x_{pl}, y_{pl}) of the last feature point P_l on the current local map L_i , and acquire the coordinates (x_{pf}, y_{pf}) of the first feature point P_f of the next local map L_{i+1} where the coordinates (x_{pl}, y_{pl}) are global map coordinates and the two feature points P_l and P_f are the same feature point in global map G , as illustrated in Figure 3.8.

3.2 Calculate the translation values j and k by

$$j = x_{pl} - x_{pf};$$

$$k = y_{pl} - y_{pf}.$$

3.3 Use the values j and k to translate every feature point (x_m, y_m) and the coordinates (x_{i+1}, y_{i+1}) of the landmark in the next local map L_{i+1} into global map G in the following way, where m ranges from 0 to the number of feature points in L_{i+1} :

$$x_m = x_m - j;$$

$$y_m = y_m - k;$$

$$x_{i+1} = x_{i+1} - j;$$

$$y_{i+1} = y_{i+1} - k.$$

3.4 Set $i = i + 1$ and go to Step 3.1 until $i = n$ which means that all local

maps L_2 through L_n have been merged into the global map G .

Step 4. Take the final global map G as output.

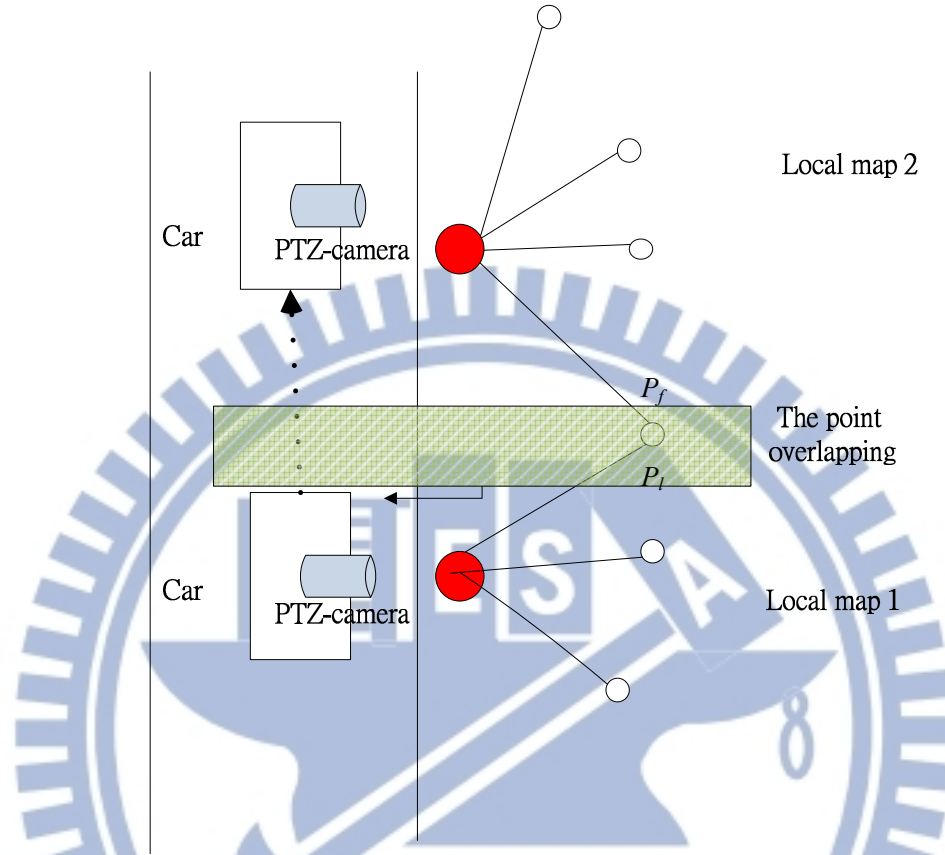


Figure 3.9 Between two local maps there is an overlapping point.

3.3.4 Algorithm for generating the guidance map

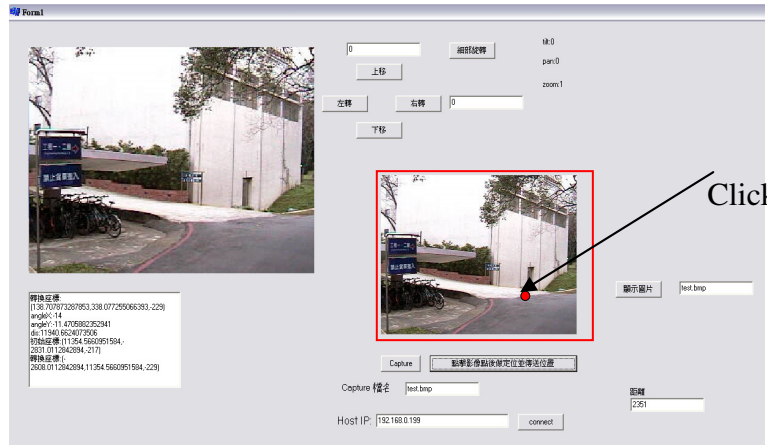
The following algorithm describes the detail of the proposed method for creating the guidance map as shown in Fig. 3.11.

Algorithm 3.5: a method for creating the guidance map in the learning process.

Input: the upper omni-image set $I_u = \{I_{u1}, I_{u2}, \dots, I_{un}\}$ and the lower omni-image set $I_l = \{I_{l1}, I_{l2}, \dots, I_{ln}\}$, and the PTZ-image set $I = \{I_1, I_2, \dots, I_n\}$, taken at n pre-selected landmarks along the way of a guided tour.

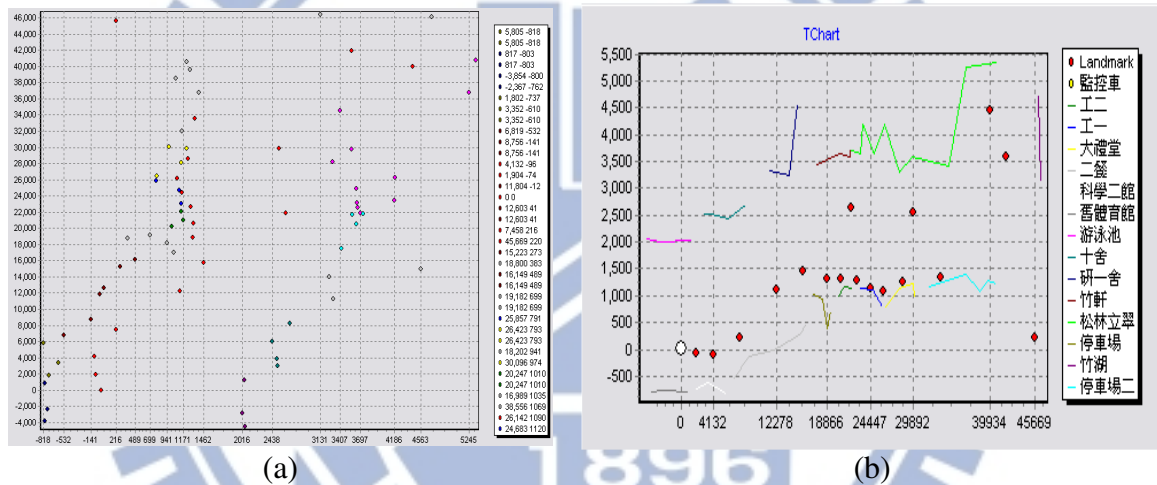
Output: a guidance map.

Steps.



(b)

Figure 3.10 The GUI designed in this study for creating local maps (b) The interface for using the PTZ-camera. (b) The interface for using the PTZ-camera.(cont'd)



(a)

(b)

Figure 3.11 Creating the guidance map. (a) Various feature points on the global map. (b) The guidance map after organizing these feature points.

Chapter 4

Automatic Detection of Circular-shaped Landmark with a Two-camera Omni-directional Imaging Device

4.1 Introduction

In this chapter, we describe the proposed method for detecting a red circular-shaped landmark in the omni-image around a video surveillance vehicle. Also described is a scheme we propose for drawing the contour of an elliptical shape on the omni-image after detecting the shape of a circular landmark in an omni-image, which appears approximately as an ellipse in the image. Specifically, the landmark is detected from the omni-image by color detection and ellipse fitting. Then, the 3D location of the center point of the landmark is estimated. Finally, the relative position of the detected landmark with respect to the video surveillance vehicle is computed. The proposed method is divided into four major stages and a flowchart of the method is shown in Fig. 4.1.

In the first stage, the process of detecting a circular shape in the real world consists of four major steps: (1) approximation of a circle in an omni-directional image by an ellipse using a method proposed by Wu and Tsai [7]; (2) YCbCr color modeling; (3) region growing; and (4) ellipse fitting.

The second stage is the process of estimating the 3D position of the detected landmark. This stage includes two major steps: (1) obtaining the center points of the ellipse on two omni-images; and (2) estimation of the 3D position of the point pair in

the two images, which correspond to a single space point in the real world. These steps of the two stages will be introduced in detail in the following sections.

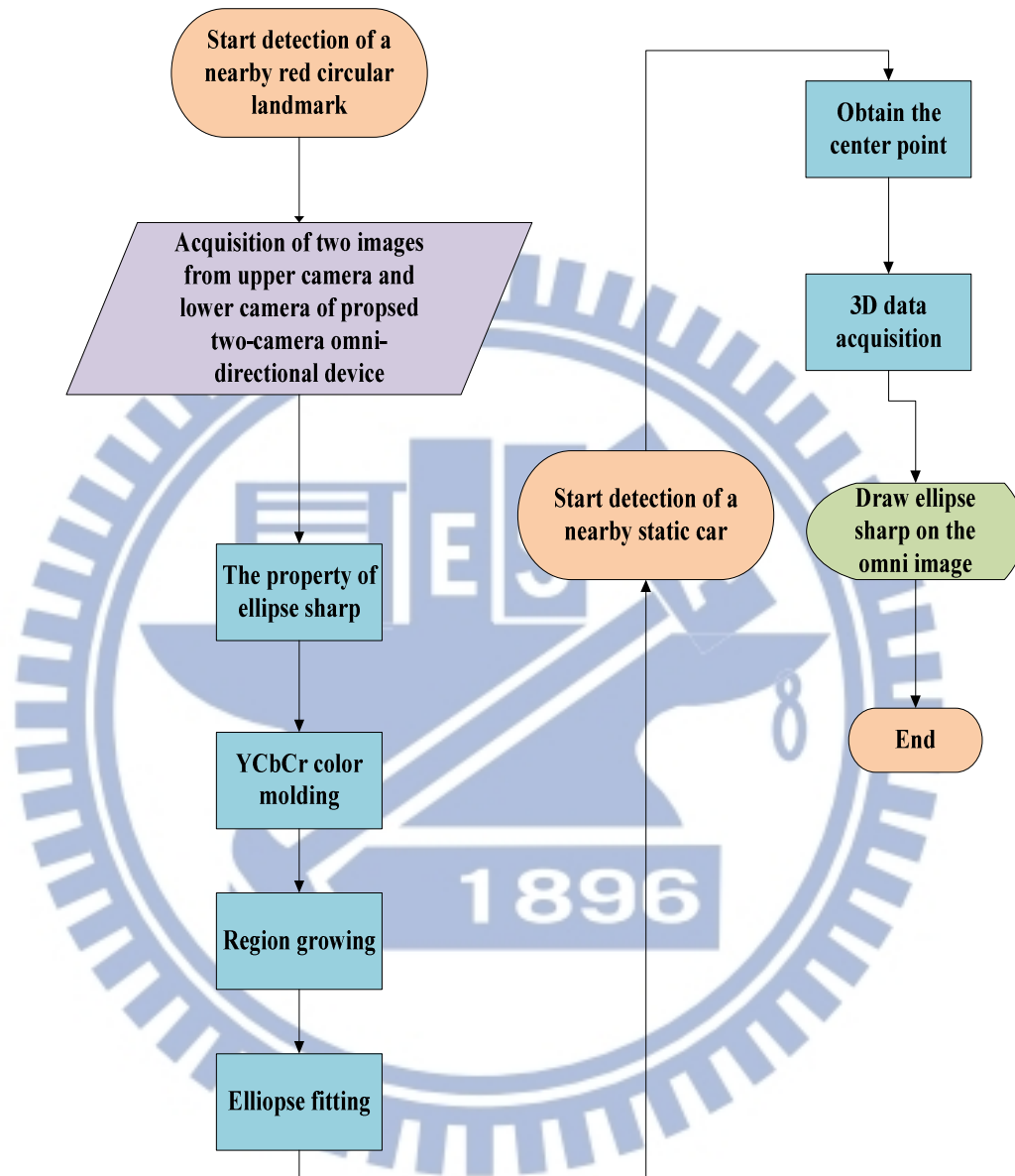


Figure 4.1 A flowchart of circular-shaped landmark detection.

4.2 Analysis of Circular-shaped landmark in Omni-directional Image

4.2.1 Approximation of Circles in Omni-directional Images by Ellipses

In this study, we place red circular-shaped landmarks along tour paths for vehicle localization in augmented reality based guided tours. The perspective shape of the circular landmark in the omni-directional image may be approximated by an ellipse, and this fact has been proved by Wu and Tsai [7]. It is reviewed subsequently.

As mentioned previously, the hyperbolical shape of the omni-directional mirror in the omni-camera coordinate system may be described as:

$$\frac{R^2}{a^2} - \frac{z^2}{b^2} = -1, \quad R = \sqrt{x^2 + y^2}, \quad c = \sqrt{a^2 + b^2}, \quad (4.1)$$

where the *omni-camera* and *omni-image coordinate systems* are specified by coordinates (x, y, z) , and (u, v) , respectively; and the projection relationship between the *omni-image coordinates* (u, v) and the *omni-camera coordinates* (x, y, z) can be described as follows [14]:

$$u = \frac{xf(b^2 - c^2)}{(b^2 + c^2)(z - c) - 2bc\sqrt{(z - c)^2 + x^2 + y^2}},$$

$$v = \frac{yf(b^2 - c^2)}{(b^2 + c^2)(z - c) - 2bc\sqrt{(z - c)^2 + x^2 + y^2}}, \quad (4.2)$$

where the parameter f is the focal length of the omni-camera.

In Figure 4.2, we define the circular-shaped landmark W and the center point of the landmark P_w and its radius R_w . Let the image of W in the omni-image be denoted by Q . Also, let (x_w, y_w, z_w) denote the omni-camera coordinates of P_w . To simplify the derivation described later, as shown in Figure 4.3, we rotate the omni-camera

coordinate system and the omni-image coordinate system horizontally through an angle of θ_w defined by:

$$\theta_w = \tan^{-1} \frac{y_w}{x_w}. \quad (4.3)$$

Then, the new omni-camera coordinates (x', y', z') may be described by:

$$\begin{aligned} x' &= x \times \cos \theta_w + y \times \sin \theta_w ; \\ y' &= y \times \cos \theta_w - x \times \sin \theta_w ; \\ z' &= z . \end{aligned} \quad (4.4)$$

And the new omni-image coordinates (u', v') may be described by:

$$\begin{aligned} u' &= u \times \cos \theta_w + v \times \sin \theta_w ; \\ v' &= u \times \sin \theta_w - v \times \cos \theta_w . \end{aligned} \quad (4.5)$$

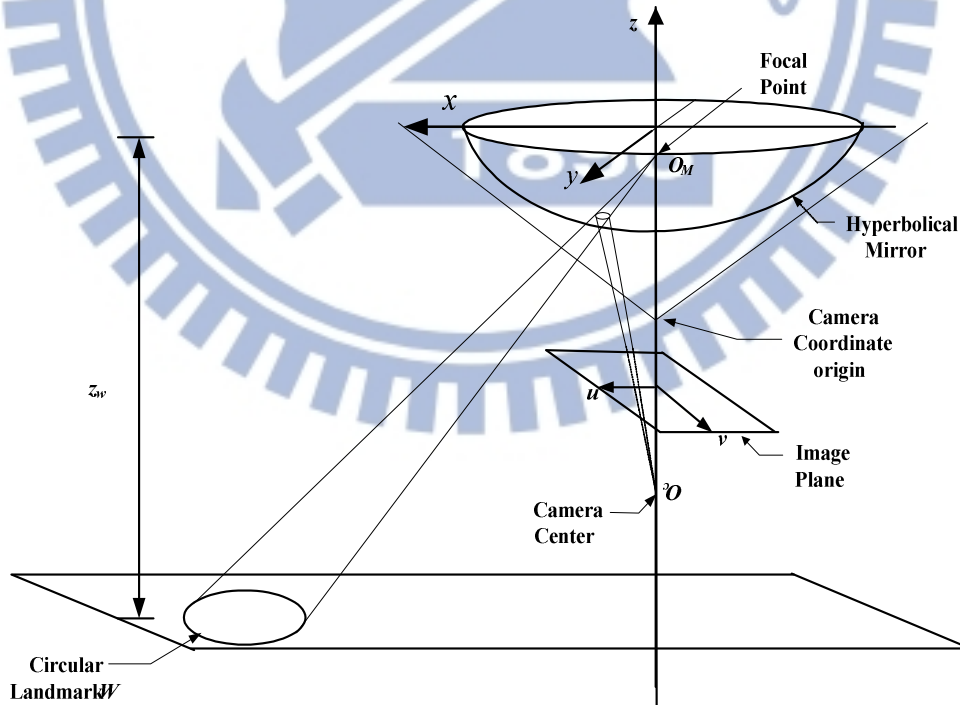


Figure 4.2: An illustration of the omni-camera coordinate system involved in this study.

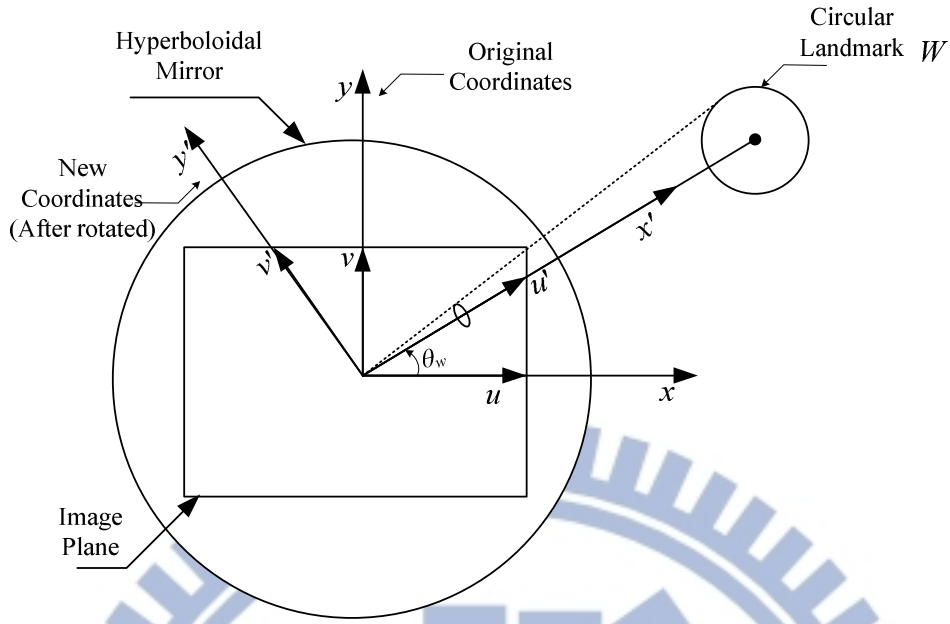


Figure 4.3: Top view from the z direction showing the relationship between new and original coordinate system with the new image coordinate system (u' , v') obtained by rotating the u -axis through an angle of θ_w with respect to the center of the circular-shaped landmark W .

Also, after this rotation, the circular shape of W in the new omni-camera coordinate system may be expressed by:

$$(x' - x'_w)^2 + (y' - y'_w)^2 = R_w^2;$$

$$z' = z_w.$$

However, the value y'_w is now zero after the rotation according to Figure 4.3, so that the above equation becomes

$$(x' - x'_w)^2 + (y')^2 = R_w^2;$$

$$z' = z_w. \quad (4.6)$$

Moreover, Equations (4.2) becomes

$$u' = \frac{x' f (b^2 - c^2)}{(b^2 + c^2)(z' - c) - 2bc \sqrt{(z' - c)^2 + x'^2 + y'^2}}$$

$$v' = \frac{y' f(b^2 - c^2)}{(b^2 + c^2)(z' - c) - 2bc\sqrt{(z' - c)^2 + x'^2 + y'^2}}. \quad (4.7)$$

In addition, the value y_w' of the center point of the circular landmark is 0. Then, by assuming that the horizontal distance from the origin of the omni-camera coordinate system to the landmark is much larger than the radius of the landmark, we have $y' \ll x'$ and the circle $(x' - x_w')^2 + y'^2 = R_w^2$ may be regarded relatively as a point whose center is located at (x_w', y_w') so that $x'^2 + y'^2 \approx x_w'^2 + y_w'^2 = x_w'^2$. As a consequence, the second equation in (4.7) above for v' can be simplified to be

$$v' = \frac{y' f(b^2 - c^2)}{(b^2 + c^2)(z' - c) - 2bc\sqrt{(z' - c)^2 + x_w'^2}},$$

or equivalently,

$$y' = v'/M \quad (4.8)$$

where M is

$$M = \frac{f(b^2 - c^2)}{(b^2 + c^2)(Z - c) - 2bc\sqrt{(Z - c)^2 + X_w'^2}}.$$

We use the same assumption mentioned above that the horizontal distance from the origin of the camera coordinate system to the landmark is much larger than the radius of the landmark, so we have $y' \ll x'$. Therefore, we can neglect the influence of the coordinate y' in the computation of u' described by the first equation in (4.7) so that

$$u' = \frac{x' f(b^2 - c^2)}{(b^2 + c^2)(z - c) - 2bc\sqrt{(z - c)^2 + x'^2}}$$

and compute u' just in terms of x' . Regarding the above equation in the form $u' = F(x')$, we may now use the Taylor series to expand the function around x_w' and ignore the terms after the second, so that we can get

$$u' \approx F(x_w') + [(x' - x_w')/1!]F'(x_w') = u_w' + (x' - x_w')F'(x_w'). \quad (4.9)$$

Equation (4.9) may be transformed easily into

$$x' \approx x'_w + \frac{(u' - u'_w)}{F(x'_w)} \quad (4.10)$$

with the first derivative F' calculated to be:

$$F'(x'_w) = A \left(\frac{1}{B} - \frac{Cx'_w}{B^2 \sqrt{D + x'_w{}^2}} \right), \quad (4.11)$$

where

$$A = f(b^2 - c^2),$$

$$B = (b^2 + c^2)(z - c) - C\sqrt{D + x'_w{}^2},$$

$$C = 2bc,$$

$$D = (z - c)^2.$$

Now with x' and y' available, we may substitute Equations (4.8) and (4.10) into Equation (4.6); and rearranging the result, we can get

$$\frac{(u' - u'_w)^2}{R_w^2 F'(x'_w)^2} + \frac{v'^2 M^2}{R_w^2} = 1 \quad (4.12)$$

which obviously generates an elliptical shape centered at $(u'_w, 0)$ with the lengths of the major and minor axes being $R_w F'(x'_w)$ and R_w/M , respectively.

4.2.2 Property of Red Circular-shaped Landmarks in Omni-directional Images Approximated by Ellipses

In this study, we choose red circular cardboards to be our landmark as shown in Figure 4.4, and the radius of each circle is 54 cm. According to Section 4.2.1, the circular landmark can be detected by an ellipse fitting method in the omni-image. Moreover, red is an obvious color in the guidance area, and so we use the YCbCr color model to detect the red landmark.



Figure 4.4 A photo of a used landmark.

In Section 4.2.1, we derived Equation (4.12) to describe an elliptical shape in the omni-image. However, if the height z is a constant, we can prove that if the distance $x_{wA'}$ from the center point of the omni-camera to the center point of the landmark W_1 is the same as the distance $x_{wB'}$ from the center point of the omni-camera to the center point of the landmark W_2 , the area of the ellipse A is also the same as the area of the ellipse B , as shown in Figure 4.5. The proof is conducted in the following.

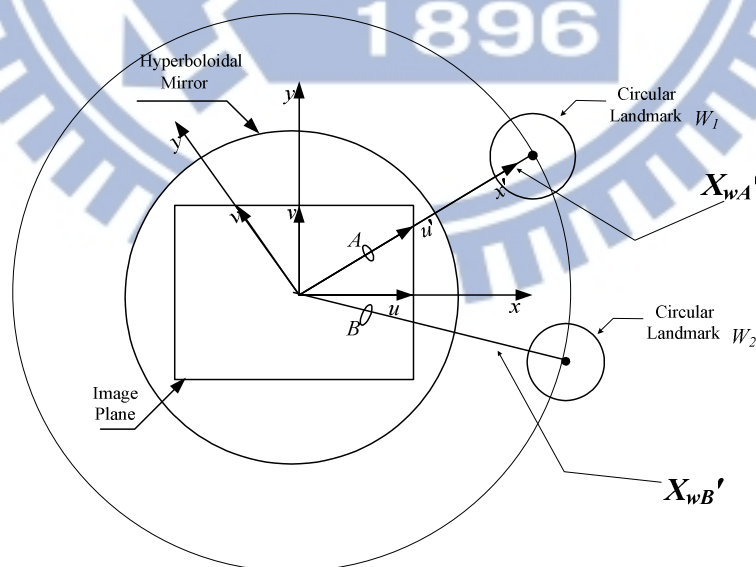


Figure 4.5 Top view from the z direction showing the area of the ellipse A is equal to the area of the ellipse B when the distance x_w' is the same.

As mentioned above, an elliptical shape in the omni-image may be described by

the equation in (4.12). In addition, the area $area_i$ of an elliptical shape denoted by i can be computed as follows:

$$area_i = R_w^2 F'(x_{wi}')^2 \times \frac{R_w}{M_i} \times \pi. \quad (4.13)$$

The lengths of the major and minor axes are $R_w F'(x_{wi}')$ and R_w/M_i , respectively, and R_w is the radius of the circular landmark. Because we use landmarks of the same size, the value R_w is a constant. In addition, the values $F'(x_{wi}')$ and M_i are computed in terms of x_{wi}' , so that $F'(x_{wA}')$ and M_A are equal to $F'(x_{wB}')$ and M_B , respectively. Finally, we reach the desired conclusion that the area of A is equal to the area of B . This completes the proof. Through the proof, we also know that if the areas of the ellipses are the same in the omni-image, the distances x_{wi} are also the same.

Additionally, we define u_{wi}' as the distance between the center point of the ellipse i and the image center in the omni-image. Then, u_{wi}' can be computed as follows:

$$u_{wi}' = \frac{x_{wi}' f(b^2 - c^2)}{(b^2 + c^2)(z - c) - 2bc\sqrt{(z - c)^2 + x_{wi}'^2}}. \quad (4.14)$$

By Equation (4.14), we can see that if the areas of two ellipses i are the same in the omni-image, the distance x_{wi} and u_{wi}' also are the same.

By the proof as described above, we can record the relationship between the area of the ellipse i and the distance x_{wi}' to simplify our landmark detection work. At first, we set the maximum height z_{mam} between the center point of the landmark and the omni-camera, which is a half of the height of the used video surveillance vehicle in this study. Second, we can use the detection method to measure the area of the elliptical shape and the value u_{wi}' in the omni-image with the value x_{wi}' ranging from 100m to 500m. The resulting values are shown in Table 4.1.

Table 4.1 The computed values of the area of the elliptical shape and u_{wi}' .

x_{wi}' (cm)	Area of ellipse(pixel)	u_{wi}' (pixel)
100	440	99
150	320	155
200	285	184
250	144	234
300	128	256
350	120	270
400	98	290
200	78	300
500	55	315

Furthermore, we can use the least squares error (LSE) fitting method to fit the area of the elliptical shape i and the distance u_{wi}' with a curve. And then we can use the resulting curve for enhancing the landmark shape detection method. The LSE fitting method is explained as follows.

At first, we generalize a straight line to a curve described by a k th-degree polynomial as follows:

$$y = a_0 + a_1 x + \dots + a_k x^k, \quad (4.15)$$

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

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}. \quad (4.16)$$

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

$$Y = XA. \quad (4.17)$$

The above equation can be solved by pre-multiplying by the matrix transpose as follows:

$$X^T Y = X^T X A. \quad (4.18)$$

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

$$A = (X^T X)^{-1} X^T Y. \quad (4.19)$$

The result of curve-fitting of the previously-mentioned set of the areas of the ellipses and the values u_{wi}' is shown in Eq. (4.20) below and illustrated in Figure 4.6, where the value x is $1/(\text{area of ellipse } i)$ and the value $f(x)$ is u_{wi}' :

$$f(x) = 3.0802803 \times x^3 + 57690.503906 \times x^2 - 3879739 \times x + 91113576. \quad (4.20)$$

When an ellipse is detected in the omni-image, we can obtain the area of the ellipse i and the value u_{wi}' . Then, the value $(1/\text{area ellipse } i)$ may be substituted into Equation (4.20). Finally, if the value $f(x)$ is checked to be larger than value u_{wi}' , then the ellipse in the omni-image may be regarded as our circular landmark.

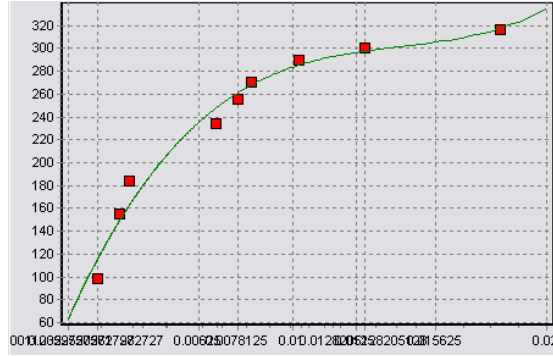


Figure 4.6 The results of curve fitting of the areas of the ellipses and the u'_{wi} .

4.3 Red Circular-shaped Landmark Detection

4.3.1 YCbCr Color Model

Many common color models are used in the field of computer vision, for examples, RGB, HSI, HSV, YCbCr, CMY, etc. Every color model has its own characteristics and can be used to a specific set of applications.

In this study, we choose YCbCr to be the color model for detecting the red landmarks. Transforming images in the RGB color model into ones in the YCbCr color model can reduce the complexity of color pixel classification. Moreover, the YCbCr color model is not sensitive to changing of lighting. And it can be used to capture the blue and red colors more easily.

In the YCbCr color model, Y represents the luminance, Cb represents the chrominance of blueness, and Cr represents the chrominance of the redness. The values of Y ranges from 16 to 235 and those of Cb and Cr range from 16 to 240. The RGB color model can be transformed into the YCbCr color model by:

$$\begin{aligned}
Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B + 16; \\
C_b &= -0.169 \times R - 0.331 \times G + 0.499 \times B + 128; \\
C_r &= 0.499 \times R - 0.418 \times G - 0.0813 \times B + 128.
\end{aligned}
\tag{4.21}$$

The RGB values used as input in this study are within the range of [0, 1], and the output values of the transformations are within the range of [16, 235] for Y and [16, 240] for Cb and Cr.

We choose the Y range to be from 126 to 235, because if Y is smaller than 126, the resulting image is too dark, and some value sets of Cb and Cr are not useful when Y is smaller than 126. Moreover, we allow Cr to range from 220 to 240 and let Cb be not limited. Because each landmark is obviously red, we can only detect the area of the red color. Figure 4.7 shows a 3D YCbCr color model [15], and Figure 4.8 shows an illustration of our landmark detection work.

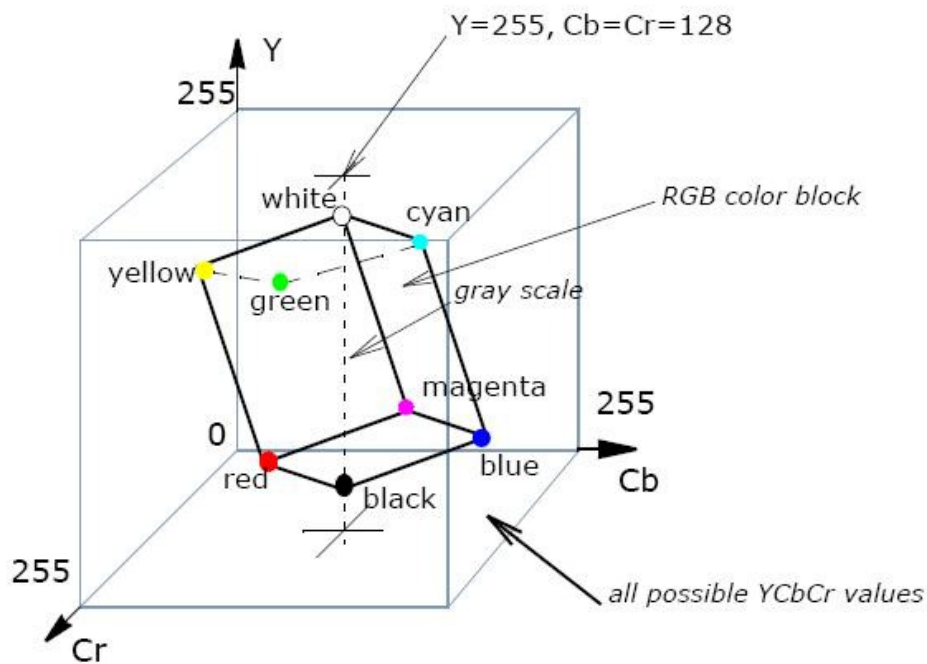


Figure 4.7: An illustration of the 3D YCbCr color model in [15].

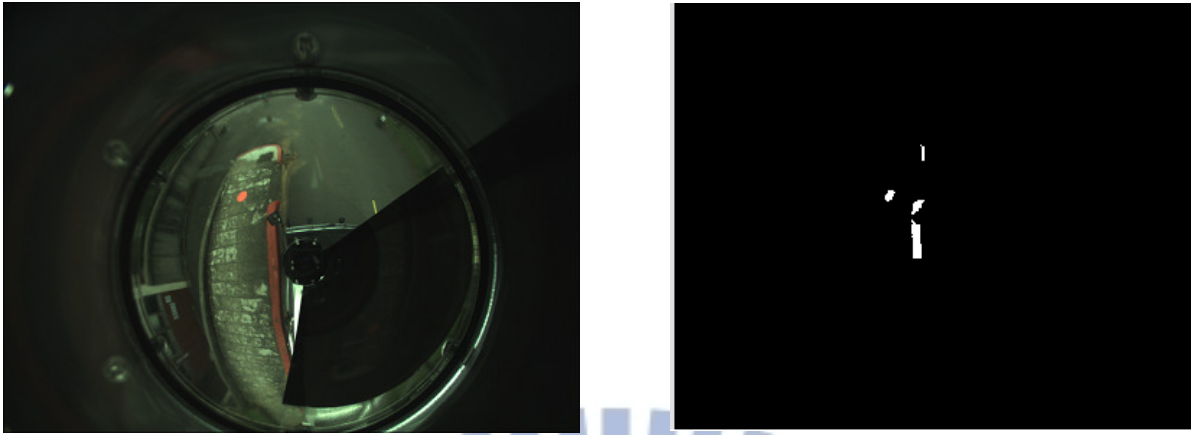
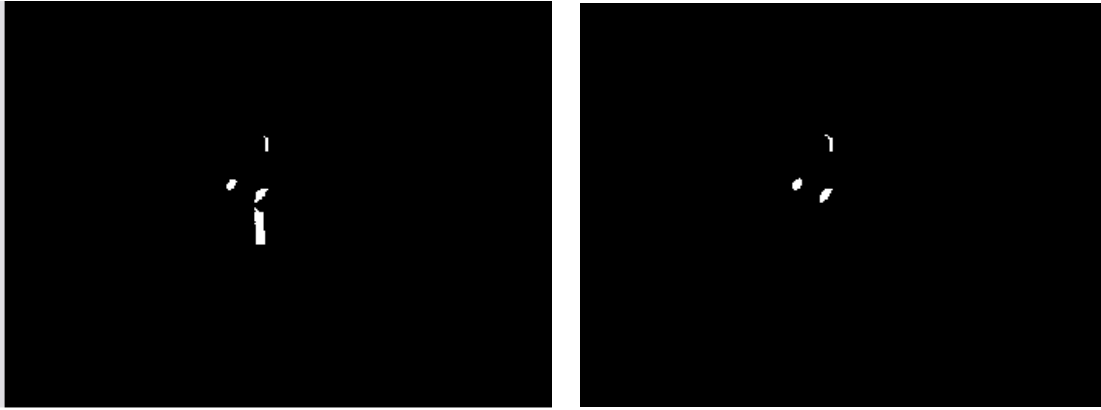


Figure 4.8: An illustration of red-colored region detection using YC_bC_r color model (a) The original omnivision image. (b) The binary image of the detection result.

4.3.2 Region Growing

After the red-colored region detection process, we can obtain red-colored regions in a binary image with “white” denoting the red color. However, the white region in the binary image may not be the elliptical-shaped region because the previously-proposed method of YC_bC_r color detection cannot eliminate the non-elliptical-shaped region clearly all the time. For example, it is possible that the color of the border on the roadside is red so that the white region in the resulting binary image will sometimes also include part of the border of the roadside. An example is shown in Fig. 4.9(a). However, these useless components are usually smaller or bigger than the elliptical-shaped region. To find the region of the detected ellipse, we have to remove such useless components in the binary image. For this, we use a region growing method and pre-select a threshold of the region, to find a proper connected component and regard it as an ellipse shape. The image resulting from such a way of removing useless components from Fig. 4.9(a) is shown in Fig. 4.9(b). The proposed method of region growing is described in the following algorithm.



(a)

(b)

Figure 4.9 The binary images of the landmark detection. (a) The image before region growing. (b) The image after region growing.

Algorithm 4.1 Region growing in the binary image.

Input: a binary image I_o and two threshold values TH_1 and TH_2 for eliminating improper regions.

Output: an image I_r including regions whose sizes are larger than TH_1 and smaller than TH_2 .

Steps.

Step 1. Initialize a new image I_r whose size is the same as that of image I_o for use in recording the searched regions.

Step 2. Divide the range 2π of the azimuth angles into N intervals and define the i -th azimuth angle θ_i as

$$\theta_i = \frac{2\pi}{N} \times i, \quad i = 1, 2, \dots, N-1. \quad (4.22)$$

Step 3. Scan each radial line l_i through the image center in the entire range of 2π according to a pre-selected azimuth angle interval to find objects in image I_o in the following way:

search any point p labeled by “1” in image I_o : if point p is also labeled by “1” in image I_r , then pass point p and continue scanning the line l_i ;

otherwise, record point p and go to Step 4.

Step 4. Grow the region from the scanned point p in the following way as illustrated in Fig 4.10.

4.1 Get a recorded point p .

4.2 Search the pixels around p .

4.3 Record the neighboring pixels around p if the pixels are labeled by “1” in image I_o and image I_r .

4.4 Go to Step 4.1 to repeat the growing process until every recorded point is checked.

4.5 If the region size is computed to be larger than threshold value TH_1 or smaller than threshold value TH_2 , then keep the region; else, ignore the region.

Through the region growing method, we can find proper connected component regions. Then, the useless components in the binary image I_o can be removed, and the resulting region is just the detected landmark shape.

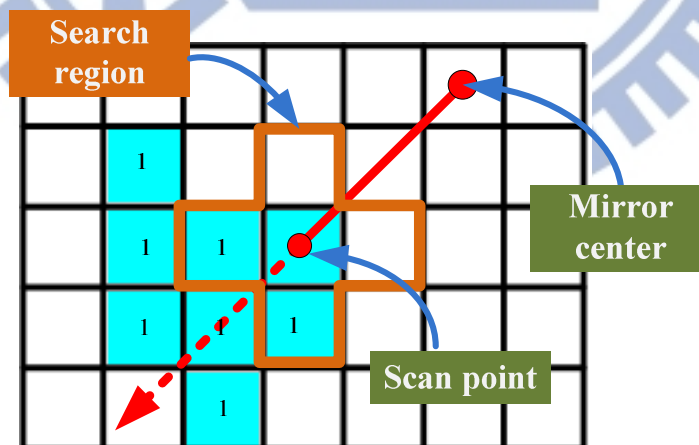


Figure 4.10 An illustration of the region growing process — the blue region represents the elliptical-shaped region and the white region represents the non-elliptical-shaped region. Once the scan point is found to be in the elliptical-shaped region, the region growing process starts.

4.3.3 Elliptical Shape Fitting in Omni-directional Images

After we use region growing to remove the bigger and smaller components, we have to obtain the real elliptical shape and the center point of the ellipse from the remaining components in the omni-image. In this section, we describe how we obtain a best-fit elliptical shape from the components for use as the desired landmark. The ellipse fitting method is described as an algorithm in the following.

Algorithm 4.2 *Ellipse fitting in the binary image.*

Input: A region set $R = \{R_1, R_2, \dots, R_n\}$ as shown in Figure 4.9(b), where R_i denotes the i -th region in set R .

Output: an ellipse region R_{ellipse} .

Steps.

- Step 1. Compute the area A_i of region R_i in unit of pixel.
- Step 2. Use the Canny edge detection method to get the edge points of region R_i .
- Step 3. Find a point P_1 with the maximal y coordinate value and a point P_2 with the minimal y coordinate value within region R_i in the omni-image, and compute the width w_i of region R_i as that of the line segment from P_1 to P_2 .
- Step 4. Find a point P_3 with the maximal x coordinate value and a point P_4 with the minimal x coordinate value within region R_i in the omni-image, and compute the height h_i of region R_i as that of the line segment from P_3 to P_4 .
- Step 5. Compare w_i with h_i , and take the bigger value of them as the major axis and the smaller one as the minor axis; and check if the minor axis length is $0.8 \times$ (the length of the major axis) and vertical to the major axis; if so, take the intersection of the major axis and the minor axis as the center point of the ellipse E_i .

- Step 6. Obtain the distance u_i between the center image and the center point of the ellipse E_i and substitute area A_i into Eq (4.20) of the curve, and if the value $f(x)$ is larger than value u_i , continue; else, go to fit the next region R_{i+1} .
- Step 7. Create a bigger version of E_i , denote it by E_{bi} , and compute the area of E_{bi} as $1.2 \times (\text{the area of } E_i)$.
- Step 8. Fit each region R_i with an ellipse shape, compute the number, in_i , of pixels of region R_i within E_i ; compute the number, out_i , of pixels of region R_i within E_{bi} and outside E_i as follows:

$$\begin{aligned} in_i &= R_i \cap E_i, \quad \forall R_i \in R; \\ out_i &= R_i \cap (E_{bi} - E_i), \quad \forall R_i \in R. \end{aligned} \quad (4.23)$$

- Step 9. Calculate a weight W_i for each region R_i in R by

$$W_i = \frac{in_i - out_i}{\text{area of } E_i}. \quad (4.24)$$

- Step 10. Decide the R_j in R as the desired ellipse region R_{ellipse} if the weight W_i of R_j is the largest among all regions in R and W is larger than a pre-defined value TH .

By the ellipse shape fitting method as described above, we can detect the ellipse region in the omni-image, as illustrated by the example shown in Figure 4.11.

4.4 Distance Estimation between a Surveillance Vehicle and a Circular-shaped Landmark

4.4.1 Review of Adopted Pano-mapping Method for Omni-image and 3D Data Acquisition Process

In order to calibrate the omni-camera, the pano-mapping method proposed by Jeng and Tsai [5] is adopted in this study. First of all, we record the relationship between the pixel in the omni-image and the elevation and azimuth angles of the corresponding world-space point with respect to the focal center of the mirror. The coordinates of the world-space points in the omni-images, called *landmark points*, are measured manually with respect to a selected origin in the world space. To facilitate selecting the landmark point pairs, a user interface is provided, as shown in Fig. 4.12. And the more landmark points are selected, the more accurate the table is.

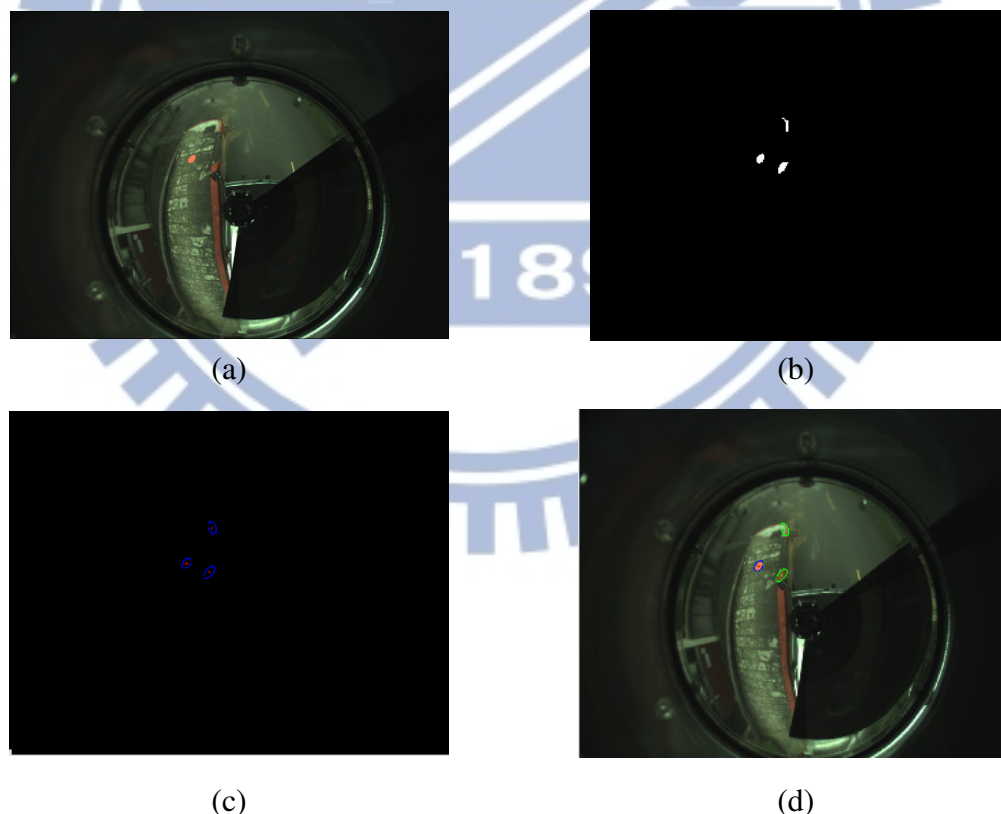


Figure 4.11 Detection of the landmark by ellipse shape fitting. (a) The image before region growing. (b) The image after region growing (c) Generating an ellipse for every region. (d) Deciding the best-fit ellipse shape, where the blue shape is the best-fit ellipse, and the green shape is an erroneous ellipse.



Figure 4.12 interface to for user to select the landmark points.

Due to the nonlinear shape of the hyperbolic-shaped mirror, the radial-directional mapping must be represented as a non-linear function f_r , called *radial stretching function*. As shown in Figure 4.13, each elevation angle corresponds to a radial distance. In this study, f_r is approximated by the following 5th-degree polynomial equation:

$$r = f_r(\rho) = a_0 + a_1 \times \rho^1 + a_2 \times \rho^2 + a_3 \times \rho^3 + a_4 \times \rho^4 + a_5 \times \rho^5. \quad (4.25)$$

More specifically, each elevation angle ρ of a scene spot P at world coordinates (X, Y, Z) corresponds to the radius r of the corresponding point p in the omni-image. Moreover, Jeng and Tsai [5] proposed an algorithm to compute the desired coefficients a_0 through a_5 , by which we can find out the relationship between the radius r and the elevation angle ρ by the use of a non-linear function f_r . However, the mirror is not perfect with rotational symmetry in the entire angle range from 0° to 360° . So we divided the 360° range of azimuth angles of the mirror equally into six parts, each with 60° , and then applied the above process to obtain six radial stretching functions f_{r1} through f_{r6} for the six parts with each f_{ri} described by the coefficients a_{0i} through a_{5i} with $i = 1, 2, \dots, 6$.

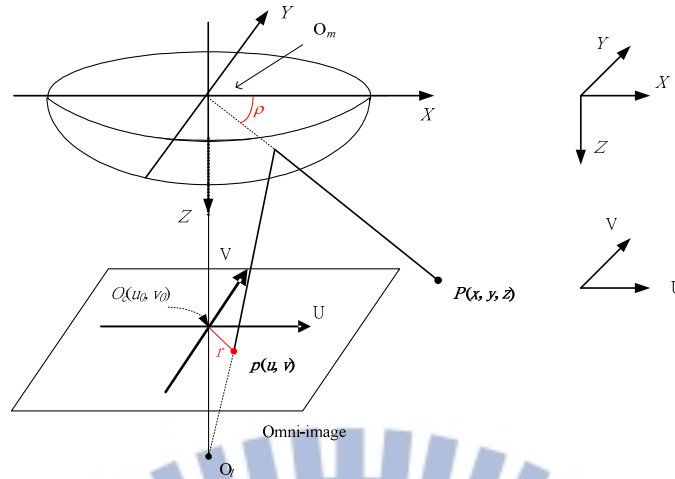


Figure 4.13 Mapping between a radius distance r and elevation angle ρ .

The procedure of constructing the pano-mapping table with the radial stretching function for each of the six azimuth angle ranges is described here. The pano-mapping table is a 2-dimensional table with its horizontal and vertical axes specifying the azimuth angle θ and the elevation angle ρ , respectively. An illustration of the pano-mapping table is shown in Fig. 4.14, and an example of the pano-mapping table is shown in Table 4.2.

Each entry E_{ij} with indices (i, j) in the pano-mapping table corresponds to an azimuth-elevation angle pair (θ_i, ρ_j) . The azimuth-elevation pair represents an infinite set of points on a light ray with the azimuth angle θ_i and the elevation angle ρ_j with respect to the focal center in the WCS. We divide the range 2π of the azimuth angles into M intervals and the range of the elevation angles between two pre-selected limits, ρ_s and ρ_e , into N intervals. Due to the property of rotational invariance of omni-imaging, the azimuth angle ϕ of the scene point P in the WCS with respect to the X -axis is identical to the azimuth angle θ of the corresponding point p in the image with respect to the u -axis. That is, we have $\theta = \phi$.

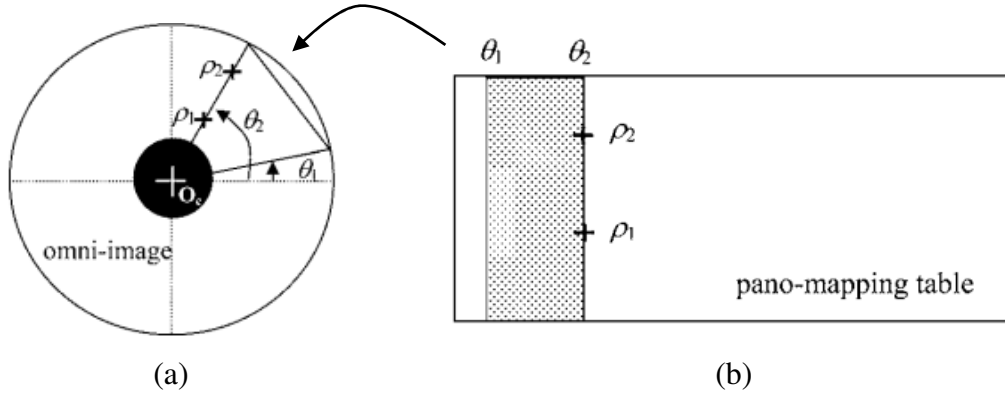


Figure 4.14 Illustration of mapping between the azimuth-elevation angle pair of the omni-image and the horizontal and vertical axes of the pano-mapping table, respectively.

Finally, the six sets of coefficients can be estimated for the six radial stretching functions, and the corresponding pano-mapping table can be filled with the corresponding image coordinates.

Table 4.2 An example of the pano-mapping table.

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

After creating the pano-mapping table, we can now describe the adopted method to compute 3D information from a two-camera omni-directional imaging device. As shown in Figure 4.15(a), the point P projected on each hyperbolic-shaped mirror forms a pair of corresponding points in the upper omni-image and the lower omni-image captured with a two-camera omni-imaging device. The elevation angles of point P on the hyperbolic-shaped mirrors are defined as α_1 and α_2 , respectively. Also, the center of the upper hyperbolic-shaped mirror is assumed to be the origin of the world coordinates $(0, 0, 0)$. It is desired now to compute the stereo depth data of

point P in terms of the two elevation angles α_1 and α_2 .

To obtain 3D information of a scene point $P(x, y, z)$, finding two elevation angles α_1 and α_2 by looking up a pano-mapping table is required. As shown in Figure 4.15(b), the distance d between the point P and the upper mirror center c_1 is computed by the triangulation principle shown in Figure 4.15(a) using the equation below:

$$\frac{d}{\sin(90^\circ + \alpha_2)} = \frac{b}{\sin(\alpha_1 - \alpha_2)}, \quad (4.26)$$

where the parameter b is the baseline of the stereo imaging device. The equation of (4.27) may be reduced to be the following equation by trigonometry:

$$d = \cos \alpha_2 \times \frac{b}{\sin \alpha_1 \times \cos \alpha_2 - \cos \alpha_1 \times \sin \alpha_2}, \quad (4.27)$$

$$d = \frac{1}{\cos \alpha_1} \times \frac{b}{\tan \alpha_1 - \tan \alpha_2}.$$

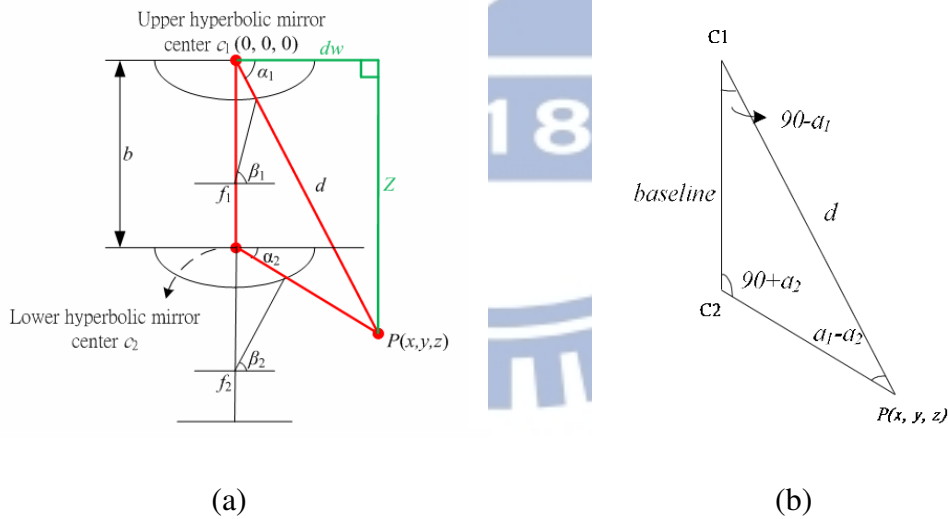


Figure 4.15 Computation of 3D information using the two-camera omnidirectional imaging device. (a) The ray tracing of a scene point P in the imaging device with a hyperbolic-shaped mirror. (b) A triangle in detail (part of (a)).

As a result, the horizontal distance dw and the vertical distance Z may be computed as follows:

$$Z = d \sin \alpha_1 = \frac{\tan \alpha_1}{\tan \alpha_1 - \tan \alpha_2} \times b, \quad (4.28)$$

$$dw = d \cos \alpha_1 = \frac{1}{\tan \alpha_1 - \tan \alpha_2} \times b.$$

Assume that point P at world coordinates (x, y, z) is projected on a point I at image coordinates (u, v) in the image coordinate system (ICS). Then, we can use point I to calculate the azimuth angle θ . A triangulation which is illustrated in Figure 4.16, includes an azimuth angle θ between the X -axis and point I . As a result, the azimuth angle θ can be computed by the following equation:

$$\theta = \cos^{-1} \frac{\mu}{\sqrt{u^2 + v^2}} = \sin^{-1} \frac{v}{\sqrt{u^2 + v^2}}. \quad (4.29)$$

According to the characteristic that the axis of the camera is aligned vertically with the axis of the hyperbolical-shaped mirror as well as the rotation-invariance property of omni-imaging, the azimuth angle of a point in the ICS is the same as that of the corresponding point in the WCS. We can calculate the parameters x and y by the distance dw and the azimuth angle θ in the WCS as follows:

$$x = dw \times \cos \theta = \frac{1}{\tan \alpha_1 - \tan \alpha_2} \times b \times \cos \theta, \quad (4.30)$$

$$y = dw \times \sin \theta = \frac{1}{\tan \alpha_1 - \tan \alpha_2} \times b \times \sin \theta.$$

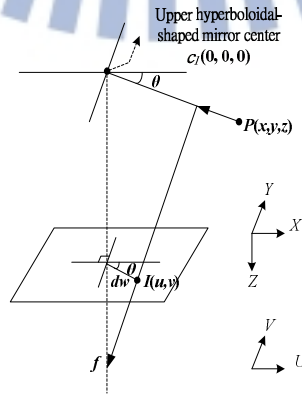


Figure 4.16 System configuration of upper omni-camera with a hyperbolical-shaped mirror.

4.4.2 Calculation of Landmark Distance

To show the relative position of the video surveillance vehicle and the detected landmark, it is required to get the 3D position information of the detected landmark. By using the 3D data acquisition method, high or low terrains can be ignored because we can obtain the 3D data of the corresponding point pairs.

Once the 3D information of the detected landmark is obtained, we can get the omni-camera coordinates (x, y, z) of the landmark. Because the omni-camera coordinate system is regarded as the WCS and the guidance map is 2D in the WCS, we can use the 3D information to compute the distance D between the video surveillance vehicle and the landmark as follows.

$$D = \sqrt{x^2 + y^2}. \quad (4.31)$$

We can display the guidance map from the top view, as shown by the example of Figure 4.17.

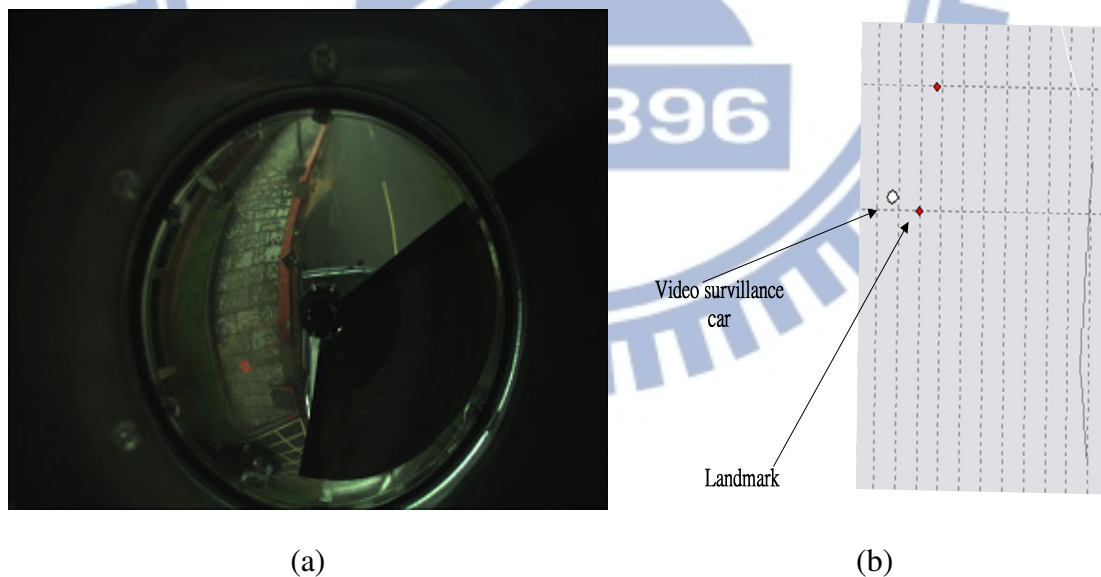


Figure 4.17 the relative position of the landmark and the video surveillance car. (a) The omni-image. (b) The position of the video surveillance car on the guidance map.

Chapter 5

Using Augmented Reality and a Two-camera Omni-directional Imaging Device for Park Guidance

5.1 Introduction

In Chapter 4, we have described the proposed method for landmark detection to obtain the relative position of a landmark with respect to the video surveillance vehicle, but we don't know the position of the detected landmark on the guidance map. In this chapter, we propose a method to obtain the position of a detected landmark on the guidance map and display accordingly the names of currently-visited buildings while the video surveillance vehicle is being driven.

In Chapter 4, we have described the proposed method for landmark detection to obtain the relative position of a landmark with respect to the video surveillance vehicle, but we don't know the position of the detected landmark on the guidance map. In this chapter, we propose a method to obtain the position of a detected landmark on the guidance map and display accordingly the names of currently-visited buildings while the video surveillance vehicle is being driven.

Moreover, we project the image displayed on the iPad onto the windshield for viewing to implement *augmented-reality tour guidance* in a park. However, the area of the iPad screen is not big enough to cover the entire windshield. In order to obtain a correct projection of the names of currently-visited buildings onto an appropriate spot on the windshield, we propose a method to estimate the position of the iPad in the WCS and the coverage of the projected image on the windshield.

The remainder of this chapter is organized as follows. In Section 5.2, we

introduce the techniques we propose for implementing of the augmented reality task here. In Section 5.3, we describe the propose method for navigating the video surveillance vehicle on the guidance map and displaying the names of currently-visited buildings at correct positions on the windshield.

5.2 Implementation of Augmented Reality Tour Guidance in a Surveillance Vehicle with an iPad

5.2.1 Idea of Projecting Images Displayed on iPad onto Car Windshield

In order to implement augmented-reality tour guidance in a park in a video surveillance vehicle, we use an iPad to simulate a head-up-display device by projecting an image displayed on an iPad onto the car windshield. To show the names of the buildings currently being visited by a visitor, who is supposed to be the driver on the video surveillance vehicle, the building names are displayed on the iPad as an image which then are projected onto the car windshield (with reflective paper attached) for the driver to observe. It is in this sense that augmented reality is accomplished in this study; or we can say that the real *scene* seen by the driver through the windshield is *augmented* by the *projected building names* appearing on the windshield.

The driver can look up at the names of nearby buildings projected onto the windshield conveniently and comfortably rather than look down at the LED monitor, reducing the possibility for dangerous events to occur during driving. The timing for the name of a building to appear is controlled by the proposed system such that when the vehicle is passing the building, its name will be projected for observation by the

driver for an appropriate time period (also computed by the proposed system).

Moreover, as shown in Figure 5.1(a), the iPad screen can be divided into two parts: the left part and the right part, with the vertical middle line of the iPad screen as the boundary, which then are used for displaying the names of the left and the right currently-visited buildings, respectively. However, in order to get the relative positions of the building names and that of the projected image on the WCS, we have to compute correctly the position of the projected image of the iPad on the windshield in the WCS. So, we propose a method in this study to estimate the corner points of the projected image in the WCS. Accordingly, we can also compute the coverage of the projected image on the windshield, as described in Section 5.2.2.

The position of the projected image on the windshield can be estimated with respect to the position of an eye of the driver's. Therefore, as shown in Figure 5.1(b), if the driver sits on the left front seat of the video surveillance vehicle, he/she can see the leftmost point of the iPad image projected on the windshield. Meanwhile, the rightmost point of the iPad image projected on the windshield can be seen by a guest on the right front seat of the video surveillance vehicle. Using these two positions of the front seats, we can find the maximum coverage of the projected image on the windshield. More details are described in the next section.

5.2.2 Coordinate Estimation for Projecting iPad onto the Windshield

First of all, we introduce the vehicle coordinate system (VCS) on the video surveillance vehicle described by coordinates (x, y, z) , as shown in Figure 5.2. The origin O of the VCS is the lower right corner of the windshield and the xy -plane of the VCS is parallel to the ground. Let θ denote the angle between the windshield-plane

and the xy -plane. Moreover, we place the iPad on the xy -plane, and the mirror image is a reflected duplication of the xy -plane by the windshield. Then, the angle between the mirror image plane and the windshield-plane is also the θ .

In order to estimate the position of the projected image on the WCS, we propose a method to compute the coordinates of a point on the projected image at first. The detail of this process is described as an algorithm below.

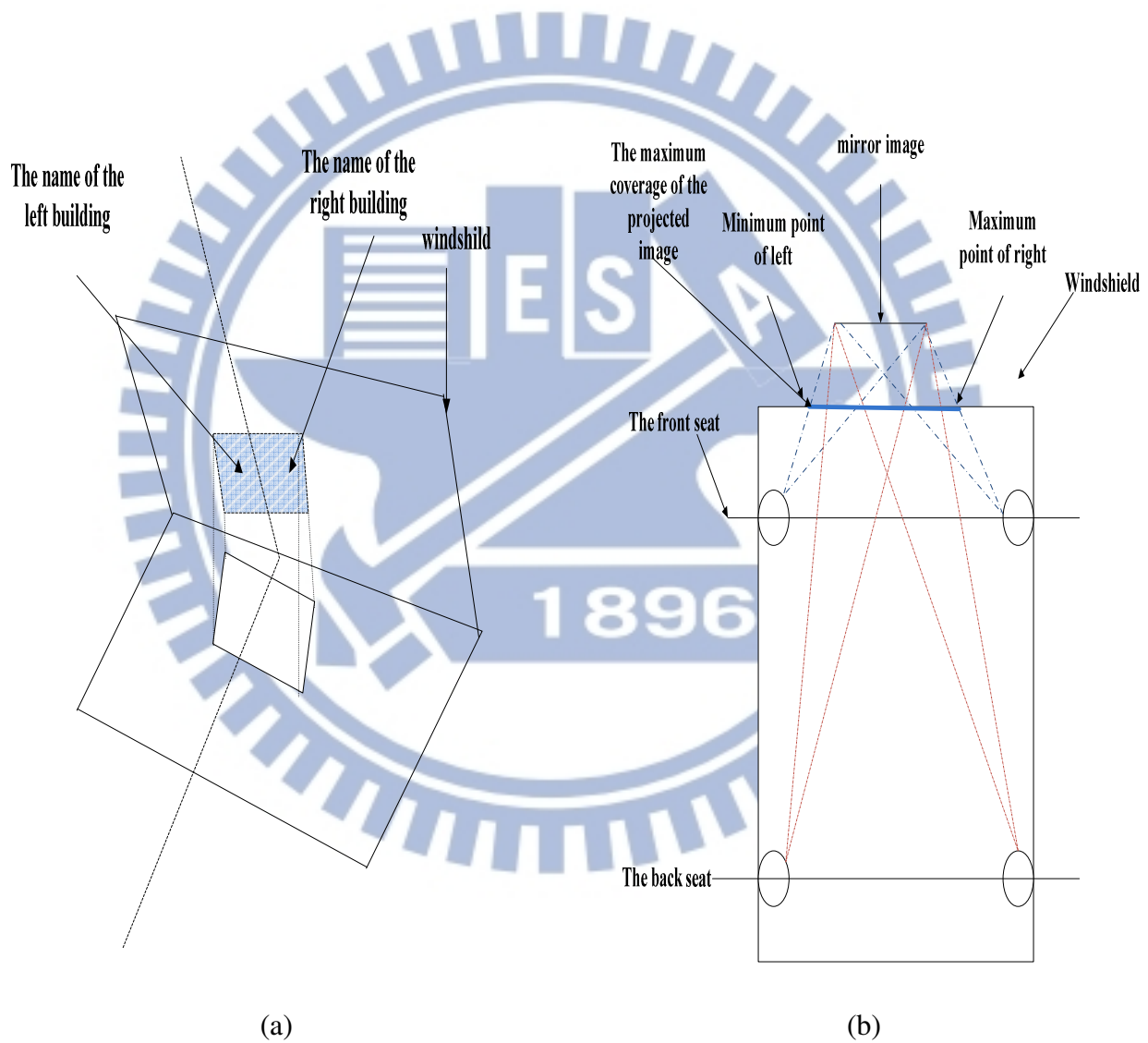


Figure 5.1: An illustration of the augmented reality display of the image on the iPad. (a) The projected image on the windshield (b) The relationship between the eyes of a visual angle and the positions of the seats.

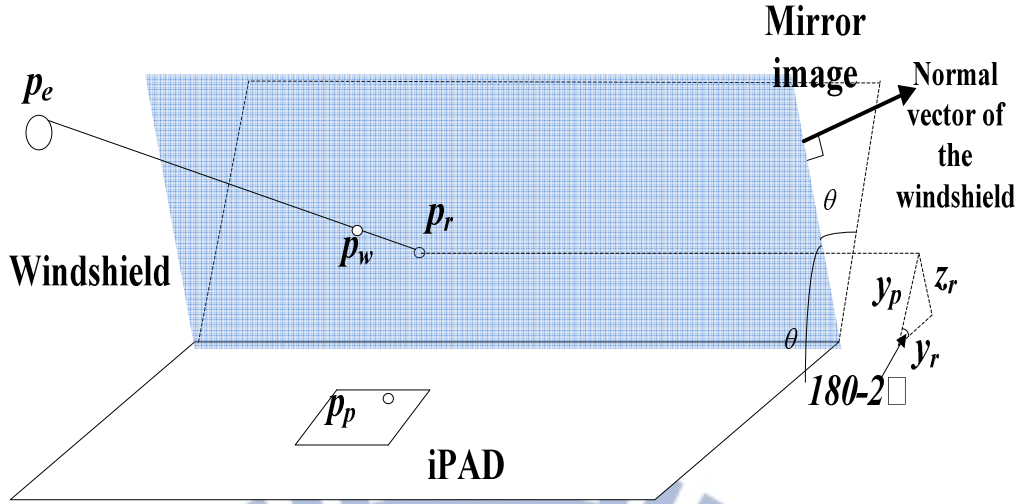


Figure 5.2 An illustration of the projection of iPad image onto the windshield.

Algorithm 5.1: computing the position of a point of the iPad image projected on the windshield.

Input: the angel θ between the windshield-plane and the xy -plane, the coordinates (x_e, y_e, z_e) of an eye p_e , and the coordinates (x_p, y_p, z_p) of a point p_p in the iPad image, called an *iPad image point*, which is on the xy -plane..

Output: the coordinates (x_w, y_w, z_w) of a point p_w on the projected image in the VCS.

Steps.

Step 5. Compute the coordinates (x_r, y_r, z_r) of a point p_r on the *mirror image* in the VCS, as shown in Figure 5.2, by

$$\begin{aligned} x_r &= x_p; \\ y_r &= y_p \times \cos(180^\circ - 2 \times \theta); \\ z_r &= y_p \times \sin(180^\circ - 2 \times \theta). \end{aligned} \quad (5.1)$$

Step 6. Compute the equation to describe the windshield-plane in the CCS as follows, where $\vec{n} = (0, \sin(90^\circ - \theta), \cos(90^\circ - \theta))$ denotes the normal vector of the windshield-plane:

$$y \times \sin(90^\circ - \theta) + z \times \cos(90^\circ - \theta) = 0. \quad (5.2)$$

Step 7. Compute the equations to describe the line from point p_e to point p_r as follows, where t is a real-valued parameter:

$$\begin{aligned}x &= x_e + (x_e - x_r) \times t; \\y &= y_e + (y_e - y_r) \times t; \\z &= z_e + (z_e - z_r) \times t.\end{aligned}\tag{5.3}$$

Step 8. Substituting (5.3) into (5.2), we get

$$(y_e + (y_e - y_r) \times t) \times \sin(90 - \theta) + (z_e + (z_e - z_r) \times t) \times \cos(90 - \theta) = 0,$$

which may be solved to get the parameter t as follows:

$$t = -\frac{y_e \times \sin(90^\circ - \theta) + z_e \times \cos(90^\circ - \theta)}{(y_e - y_r) \times \sin(90^\circ - \theta) + (z_e - z_r) \times \cos(90^\circ - \theta)}.\tag{5.4}$$

Step 9. Substituting Equation (5.4) into (5.3), we get the coordinates (x_w, y_w, z_w) of point p_w on the projected image as

$$\begin{aligned}x_w &= x_e + (x_e - x_r) \times t; \\y_w &= y_e + (y_e - y_r) \times t; \\z_w &= z_e + (z_e - z_r) \times t,\end{aligned}\tag{5.5}$$

where t is computed by (5.4) above.

In order to estimate the position and the maximum coverage of the projected image in the WCS, we need some additional information. First of all, the known length of the diagonal q of the iPad and the known ratio, a to b , of the height to the width of the iPad can be used to obtain the real values of the height h and the width w of the iPad as follows:

$$\begin{aligned}(q \times 2.54)^2 &= (a \times x)^2 + (b \times x)^2; \\h &= a \times x; \\w &= b \times x.\end{aligned}\tag{5.6}$$

In Chapter 2, we can obtain the width C_w of the video surveillance vehicle which

is also the width of the windshield. Moreover, we need to measure manually some parameter shown in Figure 5.3 which are described as follows:

- (4) the angle θ between the windshield and the xy -plane;
- (5) the distance y_e between the front seat of the video surveillance vehicle and the windshield;
- (6) the distance y_p between the position of the nearest point to the y -axis of the iPad and the y -axis;
- (7) the height h_e between the eye and the xy -plane, which we pre-set to be 17.5cm;
- (8) the coordinates (x_o, y_o, z_o) of the center point of the omni-camera on the VCS.

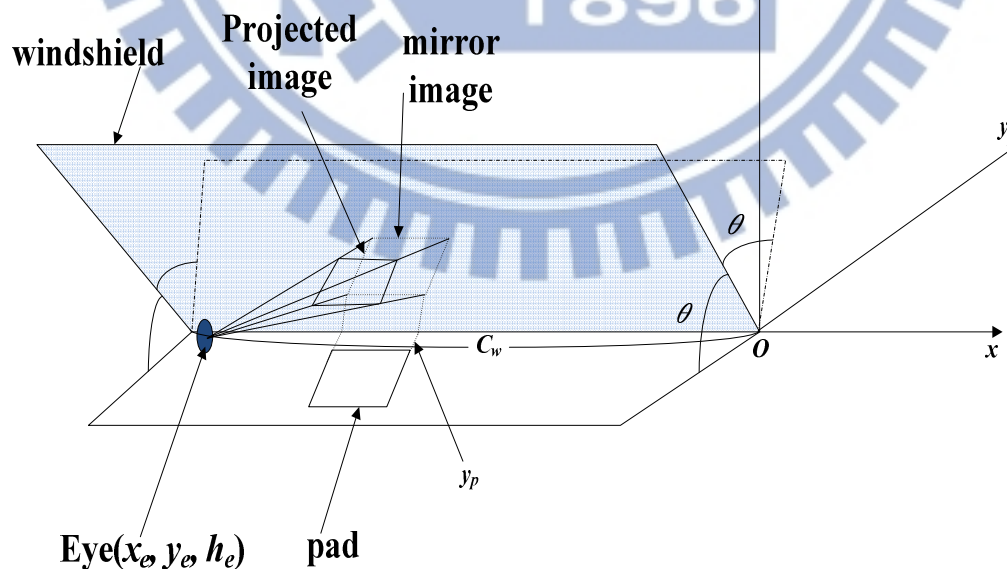


Figure 5.3 An illustration of the projected image.

Generally speaking, the height of the iPad is parallel to the y -axis and the middle

line of the iPad can be aligned with the center line of the windshield. Through the above assumptions and using the measured parameters, we can obtain the coordinates of the four corner points of the iPad as shown in Figure (5.4) in the VCS as follows:

$$\begin{aligned}
 p_1(x_{upper-left}, y_{upper-left}, 0) &= \left(\frac{C_w}{2} + \frac{h}{2}, y_p, 0\right) \\
 p_2(x_{lower-left}, y_{lower-left}, 0) &= \left(\frac{C_w}{2} + \frac{h}{2}, y_p + w, 0\right) \\
 p_3(x_{upper-right}, y_{upper-right}, 0) &= \left(\frac{C_w}{2} - \frac{h}{2}, y_p, 0\right) \\
 p_4(x_{lower-right}, y_{lower-right}, 0) &= \left(\frac{C_w}{2} - \frac{h}{2}, y_p + w, 0\right)
 \end{aligned} \tag{5.7}$$

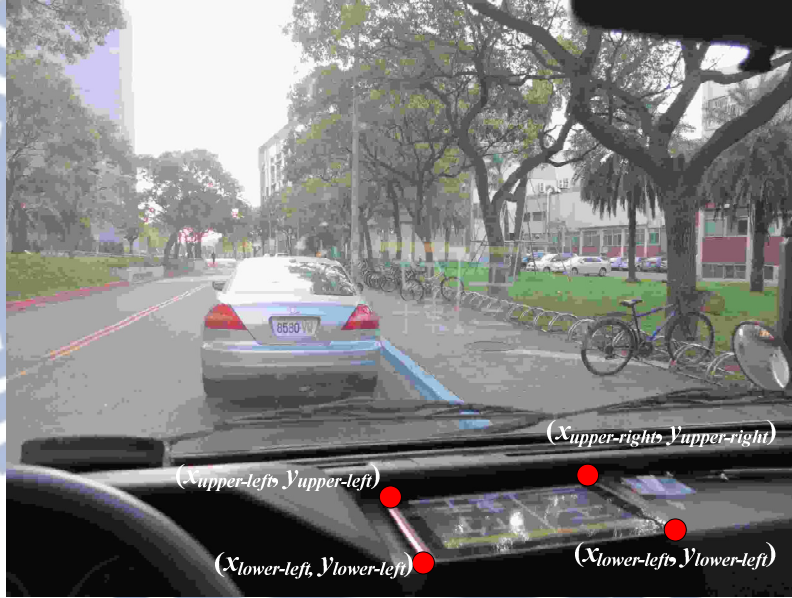


Figure 5.4 The four corner points of the iPad.

Furthermore, we can use the coordinates (C_w, y_e, h_e) of the leftmost point p_{el} and the coordinates $(0, y_e, h_e)$ of the rightmost point p_{er} of the eye to compute the leftmost point and the rightmost point of the projected image on the windshield. Using Algorithm 5.1, we can obtain the four corner points of the projected image on the VCS as follows:

- (1) using the point p_{el} and the point p_1 , we can compute the coordinates $(x_{w1}, y_{w1},$

- z_{w1}) of the upper left point p_{w1} of the projective image;
- (2) using the point p_{el} and the point p_2 , we can compute the coordinates (x_{w2}, y_{w2}, z_{w2}) of the lower left point p_{w2} of the projective image;
 - (3) using the point p_{er} and the point p_1 , we can compute the coordinates (x_{w3}, y_{w3}, z_{w3}) of the upper right point p_{w3} of the projective image;
 - (4) using the point p_{er} and the point p_2 , we can compute the coordinates (x_{w4}, y_{w4}, z_{w4}) of the lower right point p_{w4} of the projective image,

where p_{w1} and p_{w2} are the leftmost corner points, and the p_{w3} and p_{w4} are the rightmost corner points in the VCS as shown in Figure 5.5. We can use the four corner points on the projected image to compute the coverage of the projected image on the windshield easily.

Furthermore, the coordinates (x_o, y_o, z_o) of the center point of the omni-camera on the VCS is the origin of the WCS. So we can use the coordinates (x_o, y_o, z_o) to compute the coordinates of the corner points of the projected image in the WCS by

$$(x'_{corner}, y'_{corner}, z'_{corner}) = (x_{corner} - x_o, y_{corner} - y_o, z_{corner} - z_o) \quad (5.7)$$

where the coordinates $(x_{corner}, y_{corner}, z_{corner})$ specify the corner points of the projected image on the VCS and the coordinates $(x'_{corner}, y'_{corner}, z'_{corner})$ specify the corner points of the projective image in the WCS.

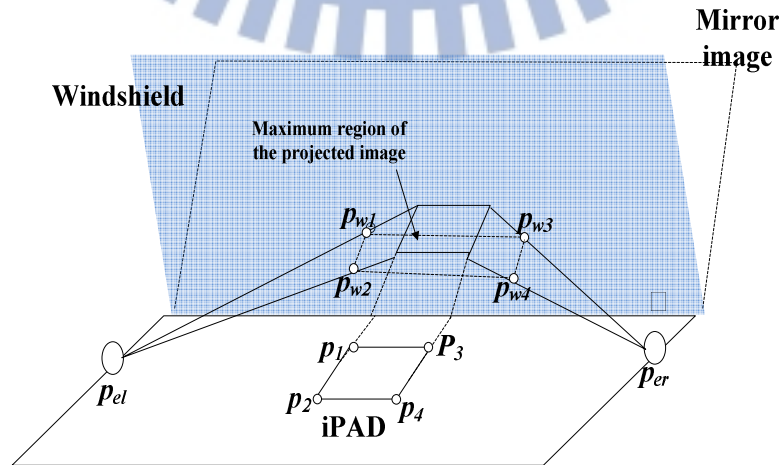


Figure 5.5 Illustration of the coverage of the projected image on the windshield.

5.3 Showing Names of Buildings to the Left and Right of Surveillance Vehicle on the Windshield

5.3.1 Computing Accurate Position of Surveillance Vehicle on Guidance Map

In Chapter 4, we described how to detect red circular-shaped landmarks and obtain the relative positions of the landmarks with respect to the video surveillance vehicle. However, we have to get the position of the video surveillance vehicle on the guidance map, so that we can locate it on the map for tour guidance. For this aim, we propose a method to check the next landmark, and obtain the positions of the landmark and the video surveillance vehicle on the guidance map.

In Chapter 3, it was mentioned that we assign a label to every landmark and select a landmark to be the start point of a guidance tour with the start label of 0. Also, we assume that the guidance path is a combination of straight line segments (or we may say it is piecewise linear). If the label of the landmark can be acquired, we can obtain the position of this landmark on the guidance map.

The proposed method for computing the position of the video surveillance vehicle on the guidance map is described in the following, which consists of two major steps.

A. Selection of the detection region for landmark analysis

In order to check the next landmark in the omni-image, it is required to choose a detection region in the omni-image. Because we place the landmarks on the

right-hand sidewalk and the omni-image content is dissymmetric, we select a *detection region* in the omni-image, and classify it into the *left-front region* and the *left-back region*, as illustrated in Fig. 5.6. Then, the landmark checking process has three major steps:

(1) at the landmark with the start label, if a landmark is detected to be in the left-front region, we know that the video surveillance vehicle will pass through the landmark;

(2) if a landmark in the left-back region of the omni-image can be detected, we know that the video surveillance vehicle will leave the current landmark;

(3) if a landmark in the left-front region of the omni-image can be detected again, we can know that the video surveillance vehicle has reached the next landmark.

Then, the label of the landmark is changed to the next one and we may now go to the second step to check the next landmark again.



(a)

(b)

Figure 5.6 Illustration of selected detection regions. (a) The left-front region. (b) The left-back region.

B. Estimation of the position of the video surveillance vehicle on the guidance map

As mentioned above, we can obtain the position of the detected landmark on the

guidance map. Therefore, we can use the detected landmark and the turning direction to estimate the position of the video surveillance vehicle on the guidance map. The detail of this process is described as an algorithm below. The method for analyzing car turning directions is to be proposed in Chapter 6.

Algorithm 5.2: computing the position of the video surveillance vehicle on the guidance map.

Input: the coordinates (x_{l1}, y_{l1}) of the detected landmark on the guidance map and the coordinates (x_{l2}, y_{l2}) of the detected landmark in the omni-image.

Output: the coordinates (x_c, y_c) of the video surveillance vehicle on the guidance map.

Steps.

- Step 1. Let the car direction be parallel to the positive x -axis on the guidance map.
- Step 2. By using the car turning direction as shown in figure 5.7, the car direction on the guidance map can be obtained in the following way.
 - (a) When the video surveillance vehicle is turning to the right and obtains the label of the next landmark, the car direction rotates 90 degrees clockwise.
 - (b) On the contrary, when the video surveillance vehicle is turning to the left and obtains the label of the next landmark, the car direction rotates 90 degrees counterclockwise.
 - (c) If the video surveillance vehicle is moving forward, the car direction remains unchanged.
- Step 3. Use the coordinates (x_{l1}, y_{l1}) and the coordinates (x_{l2}, y_{l2}) to compute the coordinates (x_c, y_c) of the video surveillance vehicle on the guidance map in the following way.
 - (a) If the car direction is parallel to the positive x -axis, compute x_c and y_c by:

$$\begin{aligned}x_c &= x_{l1} - x_{l2}; \\y_c &= y_{l1} - y_{l2}.\end{aligned}\tag{5.9}$$

(b) If the car direction is parallel to the negative x -axis, compute x_c and y_c by:

$$\begin{aligned}x_c &= x_{l1} + x_{l2}; \\y_c &= y_{l1} - y_{l2}.\end{aligned}\tag{5.9}$$

(c) If the car direction is parallel to the positive y -axis, compute x_c and y_c by:

$$\begin{aligned}x_c &= x_{l1} + x_{l2}; \\y_c &= y_{l1} + y_{l2}.\end{aligned}\tag{5.10}$$

(d) If the car *direction* is parallel to the negative y -axis, compute x_c and y_c by:

$$\begin{aligned}x_c &= x_{l1} - x_{l2}; \\y_c &= y_{l1} + y_{l2}.\end{aligned}\tag{5.11}$$

Using the above two major steps of computations, we can get the coordinates (x_c, y_c) of the video surveillance vehicle on the guidance map as shown in Figure 5.8.

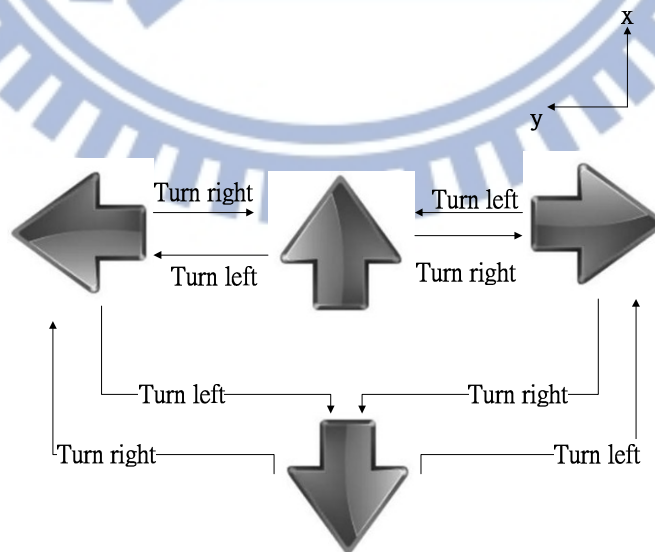


Figure 5.7 Illustration of the car direction on the guidance map.

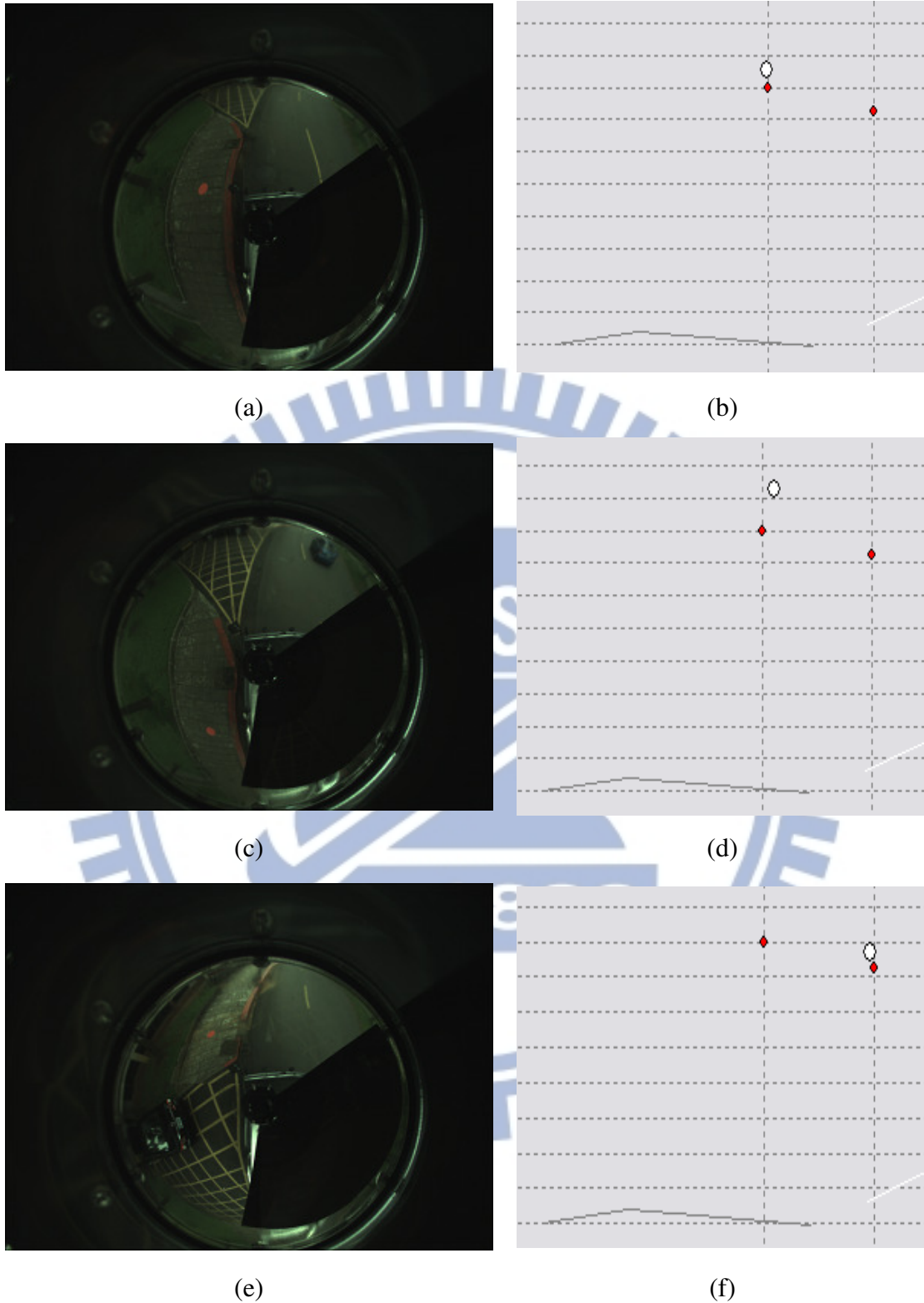


Figure 5.8 (a) The detected landmark of the left-front region. (b) The position of the video surveillance vehicle by (a) (c) The detected landmark of left-back region. (d) the position of the video surveillance vehicle by (c) (e) The detected landmark of left-front region again (f) the position of the video surveillance vehicle by (e)

5.3.2 Decision of Left and Right Buildings

In order to display the names of the currently-visited buildings on the left and right road sides on the projected image, we compute the line l from the leftmost corner points p_{w1} to the rightmost corner points p_{w2} in the projected image. If the line l and the borders of the buildings have an intersection point as shown in Figure 5.9, it means that the vehicle is passing the building, and so we can display the name of the building on the iPad screen right away for projection onto the windshield. In more detail, it is mentioned at first that we can compute the line l by

$$l: \begin{cases} x = x'_{w1} + (x'_{w1} - x'_{w2})t; \\ y = y'_{w1} + (y'_{w1} - y'_{w2})t, \end{cases} \quad (5.12)$$

where t is a real-valued parameter. Then, the detail of this process is described as an algorithm below.

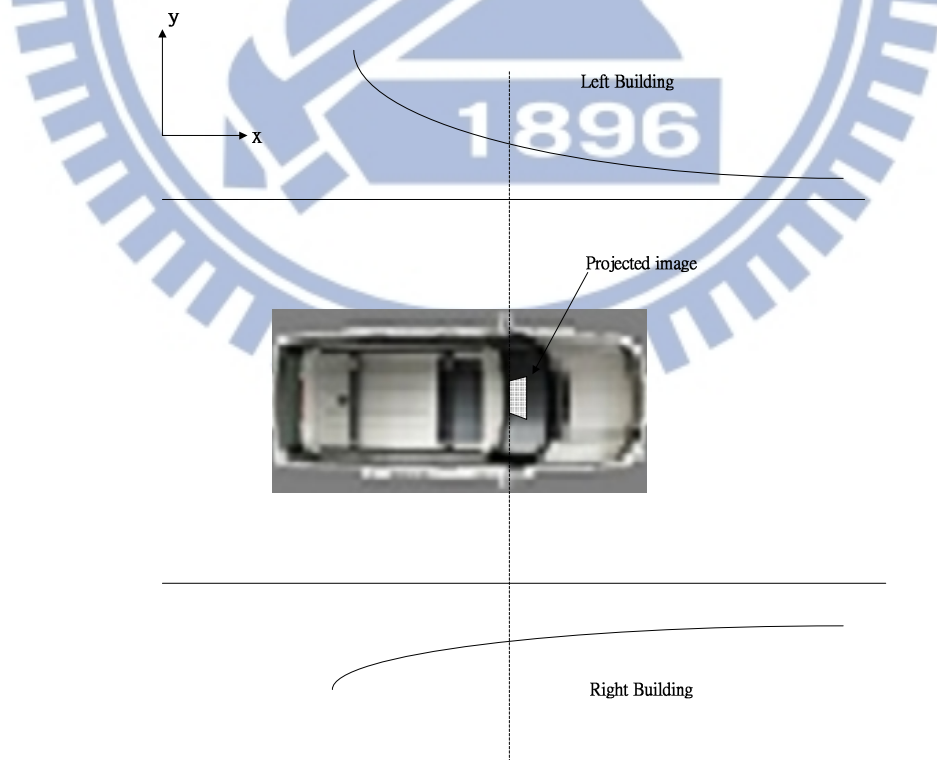


Figure 5.9 Illustration of displaying the names of buildings.

Algorithm 5.3: obtaining the names of the currently-visited buildings on the left and right road sides.

Input: the equation of a line l on the guidance map and the data of the guidance map.

Output: the names of the currently-visited buildings on the left and right road sides.

Steps.

Step 1. Obtain all the feature points on the guidance map.

Step 2. Obtain the car direction and the coordinates (x_l, y_l) of the detected landmark, and check the feature points of the left and right buildings in the following way.

(a) If the car direction is parallel to the positive x -axis or the negative x -axis, take as the right or left building points those feature points with the values of the y coordinates smaller than the value y_l , respectively, and sort these feature points by the x coordinates; otherwise, take as the left or right buildings points those feature points with the values of the y coordinates larger than the value y_l , respectively.

(b) If the car direction is parallel to the positive y -axis or the negative y -axis, take as the right or left building points those feature points with the values of the x coordinates smaller than the value x_l , respectively, and sort these feature points by the y coordinates; otherwise, take as the left or right buildings points those feature points with the values of the x coordinates larger than the value x_l , respectively.

Step 3. Obtain the coordinates (x_j, y_j) of all the left building points p_j with j ranging from 0 to the number of left building points.

Step 4. Use the equations to describe the line from point p_j to point p_{j+1} as follows, where s is a real-valued parameter:

$$\begin{aligned}x &= x_j + (x_j - x_{j+1})s; \\y &= y_e + (y_j - y_{j+1})s.\end{aligned}\tag{5.13}$$

Step 5. Compute the intersection point by:

$$\begin{aligned}x &= x'_{w1} + (x'_{w1} - x'_{w2})t = x_j + (x_j - x_{j+1})s; \\y &= y'_{w1} + (y'_{w1} - y'_{w2})t = y_j + (y_j - y_{j+1})s;\end{aligned}\tag{5.14}$$

Step 6. Solve the parameter s as follows; and if $0 \leq s \leq 1$, then take the name of the feature point p_{j+1} as the name of the left building; otherwise, set $j = j + 1$ and go to Step 4:

$$s = \frac{(y'_{w2} - y'_{w1})(x_j - x'_{w1}) + (x'_{w2} - x'_{w1})(y'_{w1} - y_j)}{(x'_{w2} - x'_{w1})(y_{j+1} - y_j) - (y'_{w2} - y'_{w1})(x_{j+1} - x_j)}.\tag{5.15}$$

Step 7. Obtain the coordinates (x_k, y_k) of every right buildings point p_k , with k ranging from 0 to the number of right buildings points; and according to the similar scheme of Steps 4 through 6, obtain the name of the right building.

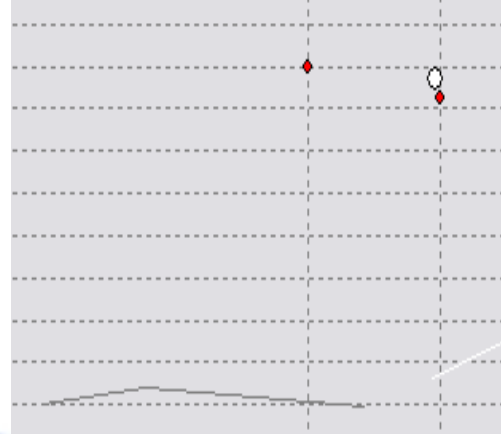
Using the above algorithm, we can get the names of the currently-visited buildings on the left and right road sides as shown in Figure 5.10.



Figure 5.10 An experimental result of displaying the names of the buildings. (a) The name of the current-visited buildings. (b) The position of the video surveillance vehicle by (a) (c) The name of the next-visited buildings. (d) The position of the video surveillance vehicle by (c).

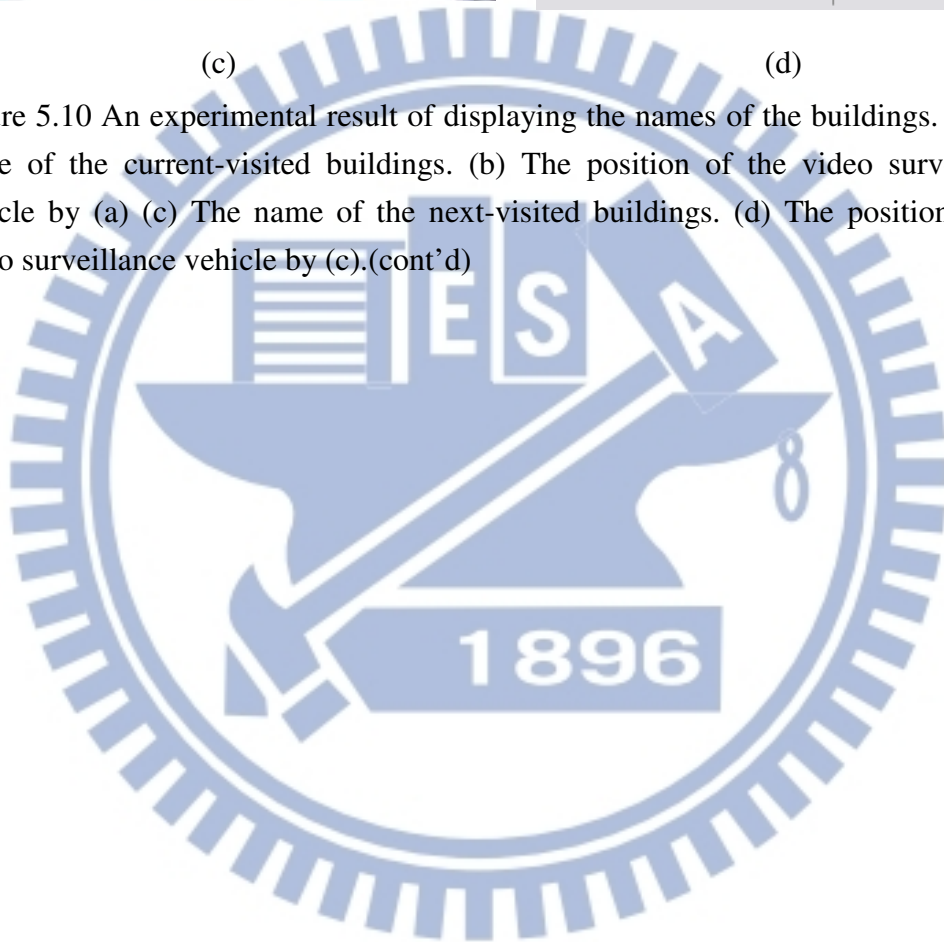


(c)



(d)

Figure 5.10 An experimental result of displaying the names of the buildings. (a) The name of the current-visited buildings. (b) The position of the video surveillance vehicle by (a) (c) The name of the next-visited buildings. (d) The position of the video surveillance vehicle by (c).(cont'd)



Chapter 6

Navigation by Guidance Map Traversals Using Vehicle Turning Information

6.1 Introduction

In Chapter 5, we describe how we obtain the position of the video surveillance vehicle on the guidance map and project the names of the currently-visited buildings onto the windshield. However, to navigate in a park area using the video surveillance vehicle according to a guidance map, we have to “traverse” paths in the map correctly so that the vehicle will not get lost in the navigation. Specifically, when meeting a branching road during driving, we have to know from the guidance map whether the landmark *to visit next* is the right landmark or the left one. Making this decision requires the information about the turning direction of the vehicle at the road branch.

In this chapter, we will describe the method by Chen and Tsai [17] which we adopt to analyze the omni-image to decide the turning direction of the vehicle. In addition, we will integrate the information of the vehicle turning direction with the guidance map to conduct correct traversals on the guidance map so that the driver will not get lost and correct building names can be projected onto the windshield for observation.

The remainder of this chapter is organized as follows. In Section 6.2, we introduce the technique proposed by Chen and Tsai [17] for analyzing the optical flows in omni-images to determine the vehicle turning direction. In Section 6.3, we propose a method for organizing the graph of the guidance map and computing the

correct position of the landmark by graph traversals.

6.2 Analysis of Vehicle Turning by Motion Vectors in Omni-images

6.2.1 Review of Vehicle Direction Analysis by Optical Flows

When meeting a branching road in a guidance tour, it is desired to analyze the motions in the consecutively acquired omni-images to determine the vehicle turning direction. Chen and Tsai [17] applied optical flow analysis to implement this idea. Based on their method, we divide the vehicle direction estimation work into four steps: (1) computing the motion vectors; (2) transforming these vectors from the omni-image coordinate system into the world coordinate system; (3) eliminating the outliers of these vectors; and (4) estimating the moving direction. These steps will be introduced in the following section.

6.2.2 Vehicle Turning Decision

C. Estimation of motion vectors by optical flows

The optical flow analysis method may be used to estimate the motion vectors of objects, surfaces, and edges caused by the relative motions between two consecutive omni-images. Assume that the displacements of concerned objects in the image are small and the illumination is stable. Under such conditions, the motion vectors between two consecutive image frames which are taken at times t and $t + dt$ can be estimated by the optical flow analysis method in the following way. If the image

intensity is continuous and can be differentiated, the image intensity at time instant t is constrained by

$$I(x, y, t) = I(x + dx, y + dy, t + dt), \quad (6.1)$$

where the function I is the image intensity, x and y specify the location of the point in the image, and t is the sampling time. The image constraint at $I(x + dx, y + dy, t + dt)$ in Equation (6.1) can be expressed as a truncated Taylor series in the following way:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt. \quad (6.2)$$

By Equations (6.1) and (6.2), it follows that:

$$I_x(p)V_x + I_y(p)V_y = -I_t(p) \quad (6.3)$$

where $V_x = dx/dt$ and $V_y = dy/dt$ represent the *velocity or optical flow* of $I(x, y, t)$ and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$ are equal to $I_x(p)$, $I_y(p)$, and $I_t(p)$, respectively, all of point p at coordinates (x, y) .

However, the values V_x and V_y are two unknown values, and cannot be solved uniquely using the data of the single point p . However, the Lucas-Kanade method [19] may be adopted here to solve the problem, which divides an image into small regions and assumes that the displacements of the image content within a small neighborhood of the concerned point p are small and approximately constant. Accordingly, we may set a window around point p with n pixels, p_1, p_2, \dots, p_n inside the window. Then, the local image motion vector (V_x, V_y) at p with image coordinates (x, y) must satisfy the following equations according to Eq. (6.3):

$$\begin{aligned} I_x(p_1)V_x + I_y(p_1)V_y &= -I_t(p_1), \\ I_x(p_2)V_x + I_y(p_2)V_y &= -I_t(p_2), \\ &\vdots \\ I_x(p_n)V_x + I_y(p_n)V_y &= -I_t(p_n). \end{aligned} \quad (6.4)$$

Eqs. (6.4) can be expressed in a matrix form:

$$Av = b,$$

where

$$A = \begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \vdots \\ -I_t(p_n) \end{bmatrix}. \quad (6.5)$$

or equivalently,

$$v = (A^T A)^{-1} A^T b. \quad (6.6)$$

Accordingly, the motion vectors of consecutively acquired images using Eq. (6.6) can be estimated above.

D. Transformation of motion vectors

To estimate the moving direction of the video surveillance vehicle, the motion vectors produced by the optical flow analysis method are used. However, the images captured with the omni-cameras are distorted. As a result, before computing the direction angle of these motion vectors, the transformation of the motion vectors from the omni-image plane to the world coordinate system as shown in Fig. 6.1(b) is necessary. In Section 4.4.1, we discussed how we transform the omni-image coordinates into the world coordinates. The configuration of such a transformation of the motion vector of a real-world point on the ground is shown in Fig. 6.1(a). Assume that we transform the image coordinates into the world coordinates (X, Y) , and transform the next image coordinates into the world coordinates (X', Y') . Then, the directional angle A_i of the motion vector V_i with respect to the X -axis can be computed by

$$\begin{aligned}
A_i &= \sin^{-1} \left(\frac{Y' - Y}{\sqrt{(Y' - Y)^2 + (X' - X)^2}} \right); \\
&= \cos^{-1} \left(\frac{X' - X}{\sqrt{(Y' - Y)^2 + (X' - X)^2}} \right).
\end{aligned}
\tag{6.7}$$

where the value i ranges from 0 to the number of motion vectors.

By Equation (6.7), all the motion vectors produced by optical flow analysis can be transformed into the WCS and their directional angles can be computed for analyzing the moving direction of the video surveillance vehicle as discussed next.

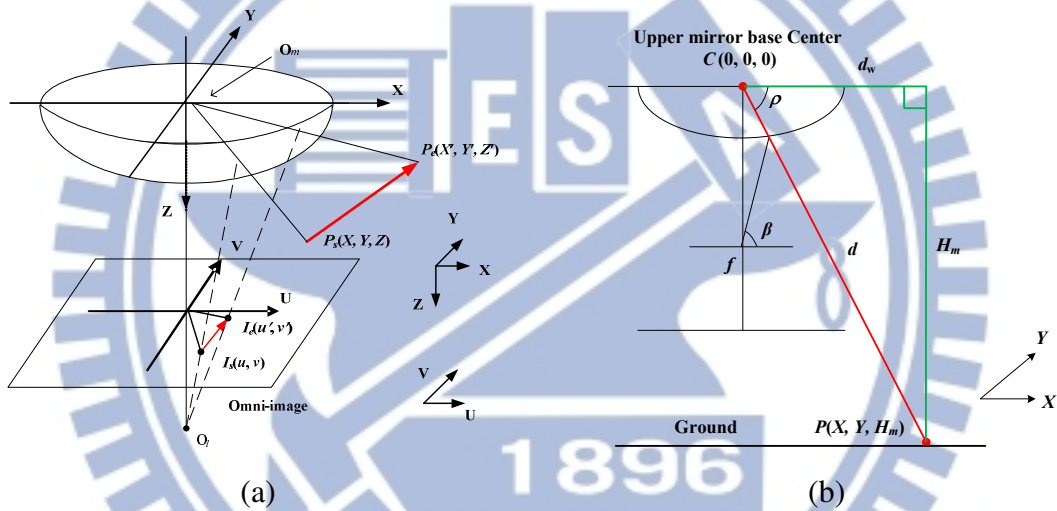


Figure 6.1 Transformation of a motion vector from the ICS into the WCS. (a) An illustration of the camera system and the motion vector. (b) The ray tracing of a scene point P on the ground projected on the hyperbolic-shaped mirror.

E. Elimination of Outliers

When we drive the video surveillance vehicle, the omni-cameras will often shake due to rough road conditions. Therefore, the shake of the omni-cameras might cause creations of short-length *noise motion vectors*. In order to eliminate such noise to increase the accuracy of the vehicle moving direction estimation result, only motion vectors with lengths be larger than a pre-selected threshold value TH are selected for vehicle direction estimation.

To eliminate the *outliers* of motion vectors resulting from noise, each directional angle of the remaining motion vectors is regarded as a feature and the standard deviation value is computed accordingly, as shown in Fig. 4.5. More specifically, let the angles of these motion vectors be denoted as A_i , and let the total number of motion vectors be denoted as n . Then, the mean value \bar{A} of these motion vectors may be computed as follows:

$$\bar{A} = \frac{1}{n} \sum_{i=1}^n A_i. \quad (6.8)$$

Once the mean value is computed, the standard deviation value S_n of the motion vector data can be calculated as follows:

$$S_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (A_i - \bar{A})^2}. \quad (6.9)$$

where S_n is the threshold value TH .

If the value A_i of a certain motion vector lies outside the range $[\bar{A} - TH, \bar{A} + TH]$, an outlier will be checked and discarded. After all the outliers are eliminated, we compute the mean value of the remaining data as the desired directional angle of the moving direction of the video surveillance vehicle.

F. Estimation of the Moving Direction of the Vehicle

The moving directions of the video surveillance vehicle may be categorized into three classes — turning to the right, turning to the left, and moving forward. And the ranges of the directional angles of the three classes are determined by our experimental experiences. They are listed in Table 6.1, which may be used to classify the results of the directional angles derived by Eq. (6.8) into the three vehicle moving directions.

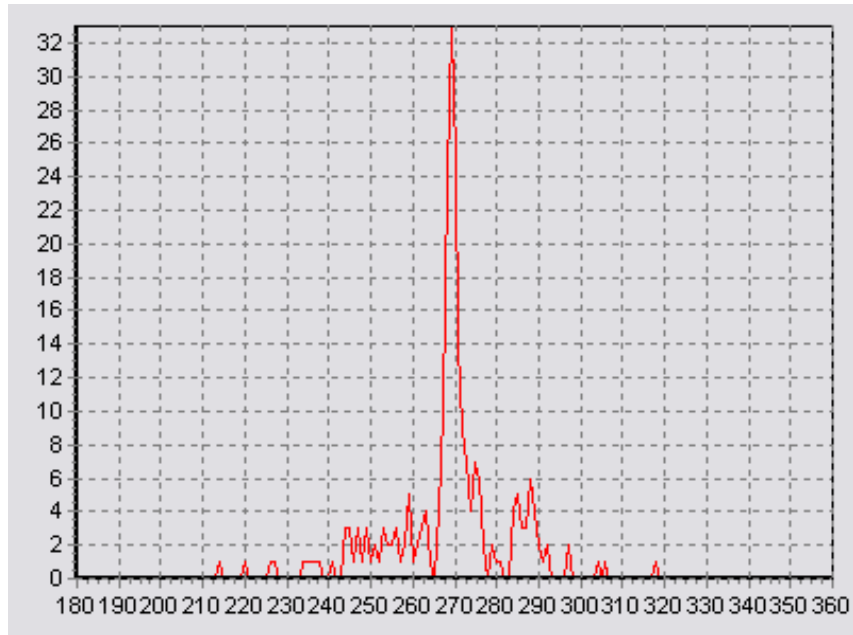


Figure 6.2 A distribution chart of the direction angles of motion vectors.

Table 6.1 The range of the angles of the three vehicle moving directions.

State	Degree
Moving forward	261° ~ 279°
Turn to the left	180° ~ 260°
Turn to the right	280° ~ 360°

Besides, a finite state machine (FSM) can be used to determine the moving direction of the video surveillance vehicle [17]. The finite state machine adopted for use in this study is composed of six states, which can be categorized into three classes: (1) turning to the right; (2) turn to the left; and (3) moving forward. As illustrated in Figure 6.3, the input to the FSM is taken to be “1”; else, to be “0.” Here, the input “1” is used to specify the same turning direction at the current state, and the input “0” to specify a different turning direction at the current state.

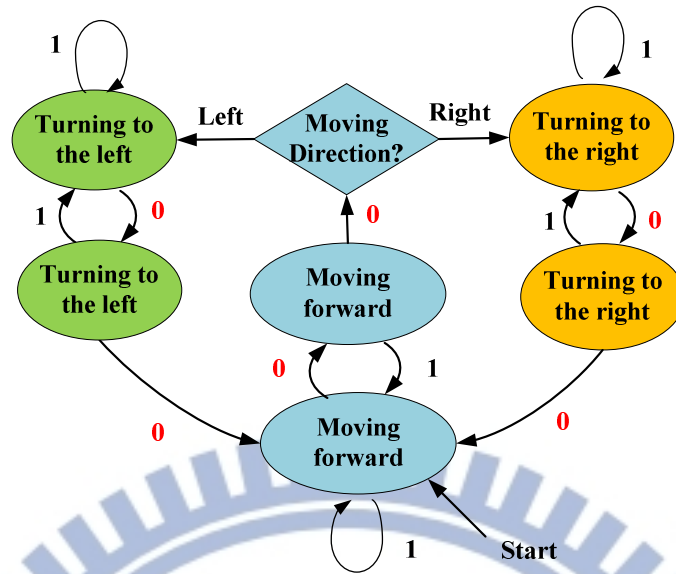


Figure 6.3 A graph of finite state machine proposed to determine the moving direction [17].

6.3 Organization of Guidance Map for Park Navigation by Graph Traversals

6.3.1 Creation of Guidance Map Graph

In order to detect the direction of the vehicle movement, right, left, or forward, on the guidance map, we have to *organize* the guidance map into a graph which we call the *guidance map graph*. At first, we have to specify the relationship between the label of the current landmark and that of the previous landmark. For example, the 8th landmark is the *forward* landmark of the 7th landmark, and the 9th landmark is the *left* landmark of the 7th landmark. That is, with respect to the current landmark, there are three types of next landmarks, *forward*, *left*, and *right*. As shown in Figure 6.4(a), for the purpose to check the turning direction of the vehicle near the current landmark *on the guidance map*, a *label* is needed to record the labels of the right landmark, the left landmark, and the forward landmark. Using the labels of the landmarks of the

three directions, we can connect these landmarks to construct a guidance map graph, as shown in Figure 6.4(b). Such a guidance map graph may be used to generate a *guidance path* for each park tour.

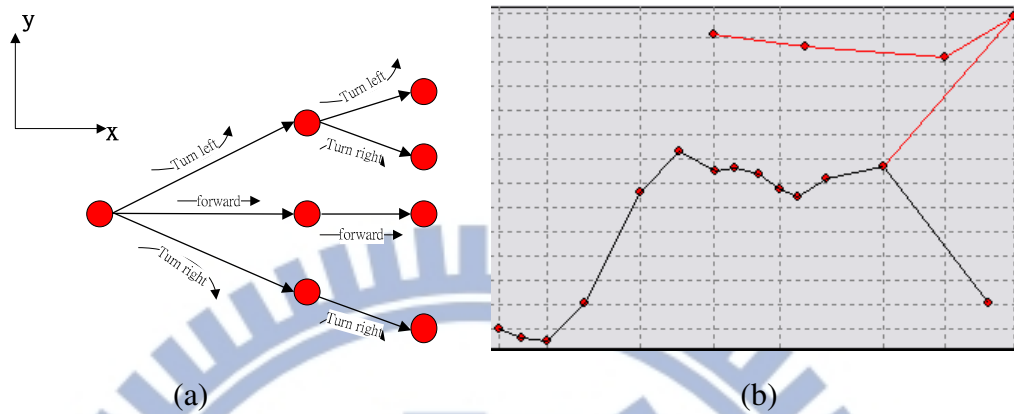


Figure 6.4 The graph of the guidance map. (a) An illustration of organizing guidance map. (b) The guidance map graph of our experimental environment.

6.3.2 Updating of Landmark Labels on Guidance Map Graph during Driving

Using the method described in Section 5.2.2, we can know that the video surveillance vehicle reaches the next landmark. However, if we meet a branching road, the label of the next landmark is *ambiguous* unless the system knows which direction, left, right, or forward, the vehicle is turning into. So we have to analyze the omni-image to decide the turning direction of the vehicle so that the system can “traverse” the guidance map graph correctly. Using the guidance map graph, we create some rules for this purpose to obtain the corresponding label of the next landmark as described in the following:

- (1) When the image analysis result of the vehicle moving direction is “turning to the right,” the next landmark is the *right* one with respect to the current landmark (i.e., the next landmark is labeled “right”).
- (2) If the vehicle moving direction is analyzed to be “turning to the left,” the next

landmark is the *left* one with respect to the current landmark (i.e., the next landmark is labeled “left”).

- (3) If the vehicle moving direction is “turning to the forward,” the next landmark is the *forward* one with respect to the current landmark (i.e., the next landmark is labeled “forward”).

We can decide the label of the next landmark as shown by the flowchart in Fig. 6.5. However, the road is not a perfect straight line, and sometimes the vehicle turning direction may be detected erroneously. So when the next landmark is reached and the turning direction is obtained, we propose to check the turning direction once again using the input omni-image to make sure the turning is correct, as described in the flowchart shown in Fig. 6.5. That is, we make sure the vehicle’s turning by a *double* check using two cycles of navigation. In this way, while we decide the next label of the landmark by turning direction detection, the probability of erroneous estimations of the vehicle turning direction may be decreased.

An experimental result is shown in Figure 6.6, where the next landmark is reached and the vehicle direction is changed from the forward to left, so that we obtain the left landmark with respect to the current landmark *on the guidance map graph*.

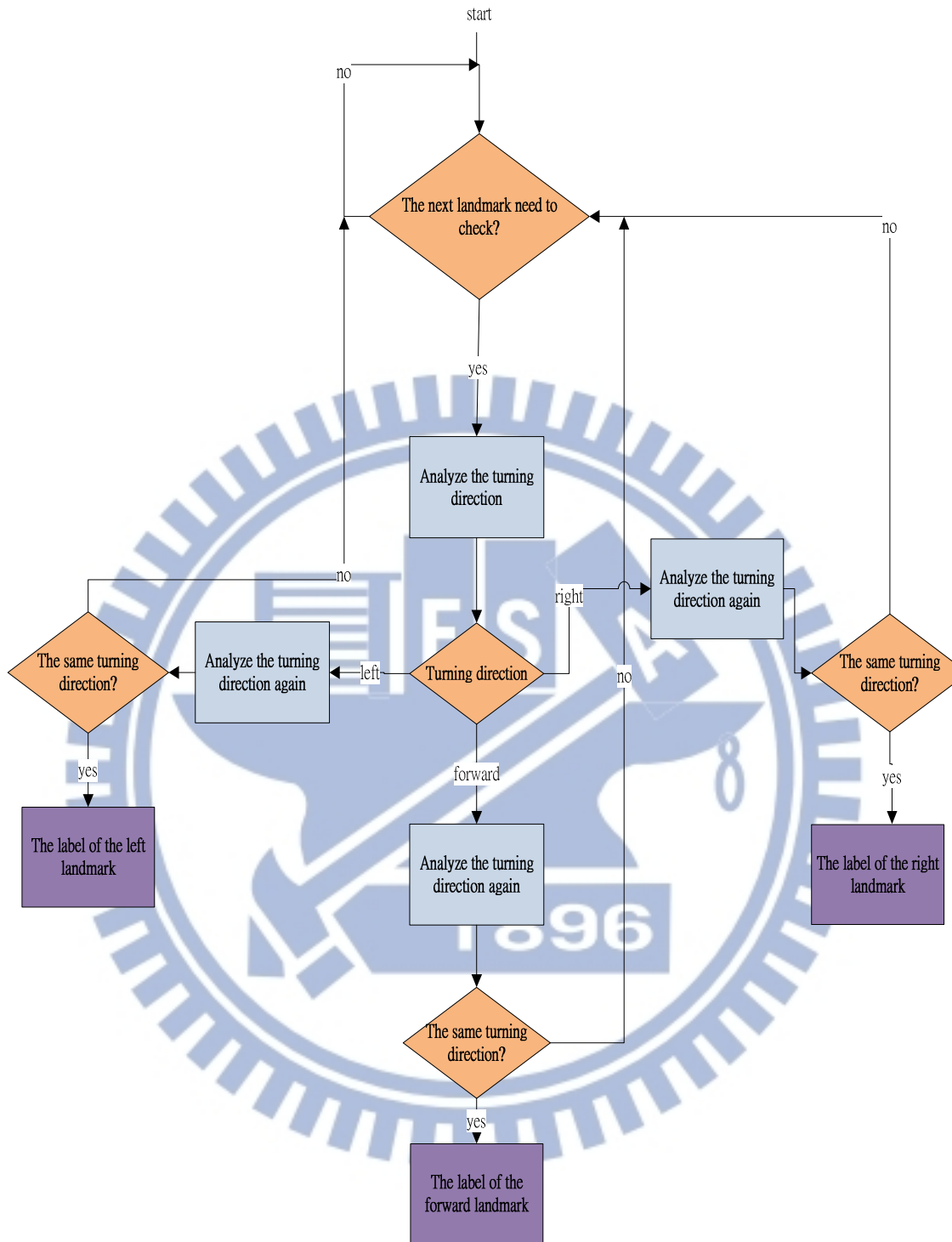
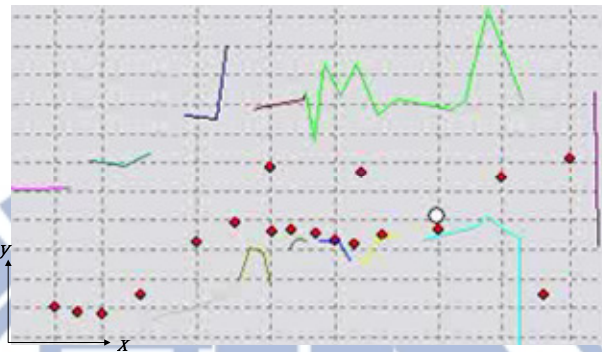
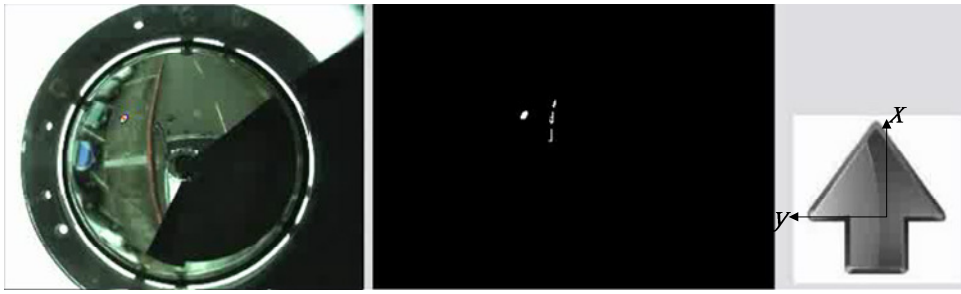
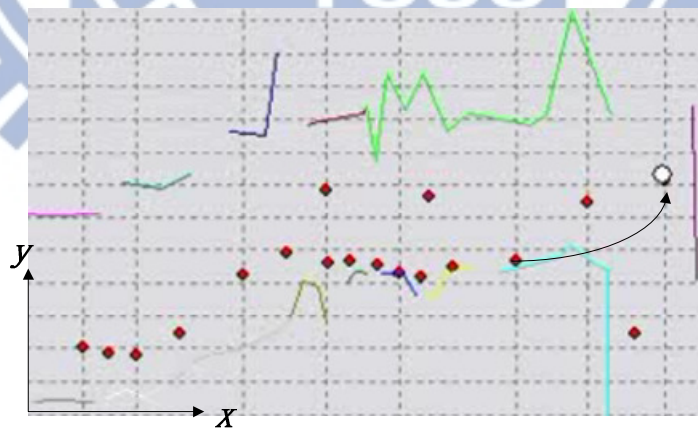
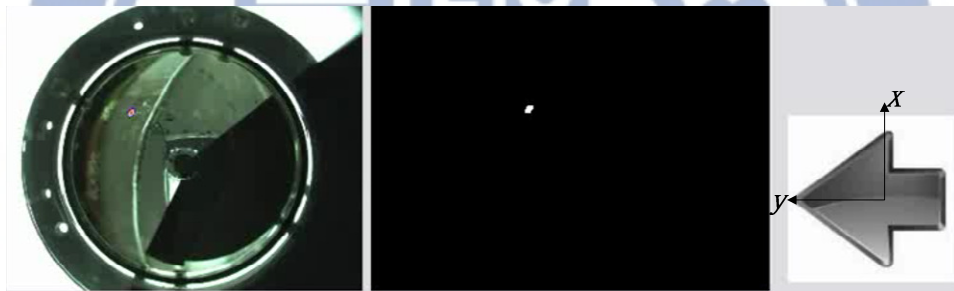


Figure 6.5 A flowchart of deciding the label of the next landmark on the guidance map graph for graph traversals.



(a)



(b)

Figure 6.6 An experimental result of turning direction determination. (a) The position of the video surveillance vehicle and the *detected* forward direction. (b) The position of the video surveillance vehicle and the *decided* left direction.

Chapter 7

Experimental Results and Discussions

7.1 Experimental Results

In this chapter, we will show some experimental results of applying the proposed augmented reality guidance system for park touring on a video surveillance vehicle. We will show the results of creating the guidance map for our experimental environment, which is part of the National Chiao Tung University campus, and the results of the navigation process based on landmarks set up for this study. An illustration of the guidance area consisting of a branch road, fourteen buildings, and some landmarks on the sidewalk is shown in Figure 7.1.

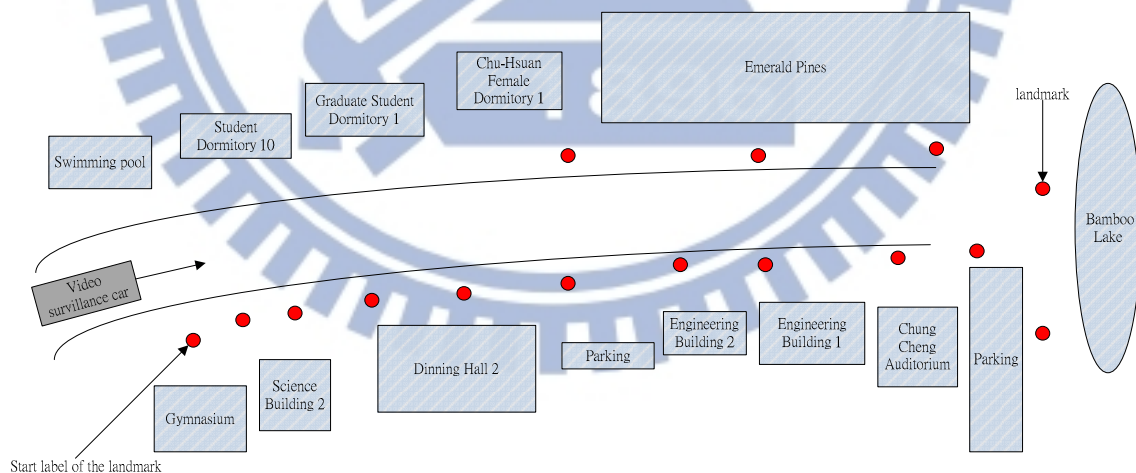


Figure 7.1. Illustration of the experimental guidance area.

In order to construct an unknown guidance area, a user interface of the guidance map creating system was constructed, which is shown in Figure 7.2. The used vehicle system has two omni-cameras and a PTZ-camera. First of all, the two omni-cameras

were used to acquire omni-images, detect the red circular-shaped landmarks in them, and obtain the position of the currently-visited landmark. Then, the PTZ-camera was used to capture feature points on the along-path buildings. Afterward, we integrate the position of the current landmark with the positions of the feature points to create a local map. In this way, we detected all the landmarks to create a set of local maps, from which we created the guidance map as shown in Figure 7.3.



(a)

(b)

Figure 7.2 A learning interface of the experiment. (a) An interface for using the omni-camera. (b) An interface for using the PTZ-camera.

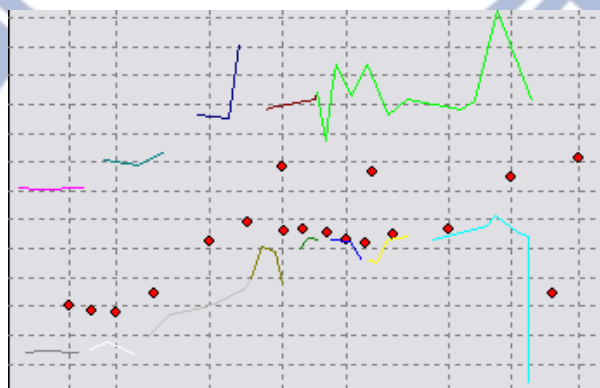


Figure 7.3 A guidance map of the experiment environment.

In the augmented reality guidance process, the video surveillance vehicle started from the landmark with the *start label* of 0. After a user presses the start button, the

system will detect the landmark, and compute accordingly the position of the video surveillance vehicle on the guidance map, as shown in Figures 7.4 and 7.5.



Figure 7.4 An experimental result of landmark detection. (a) An elliptical-shaped landmark in the omni-image. (b) Image of landmark detection by region growing.

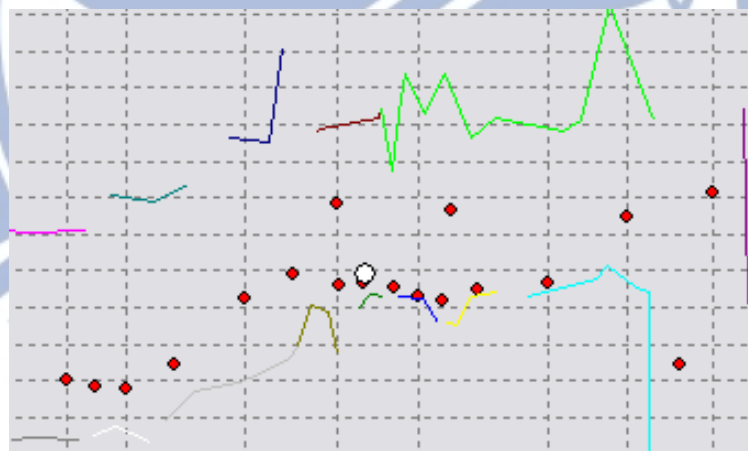


Figure 7.5 A guidance map of the experimental environment.

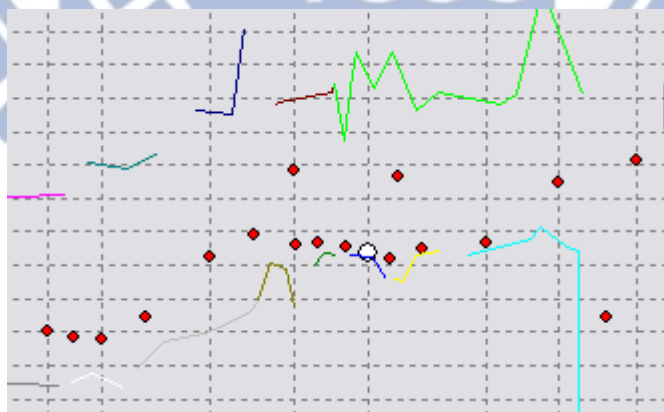
Next, when the road is a strait line and the video surveillance vehicle reaches the next landmark, the guidance map can be updated by marking the new position of the video surveillance vehicle and the label of the currently-visited landmark on the guidance map as shown in Figures 7.6 through 7.8.



(a)

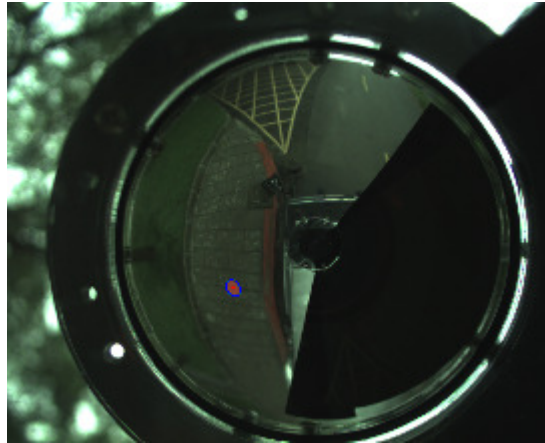


(b)



(c)

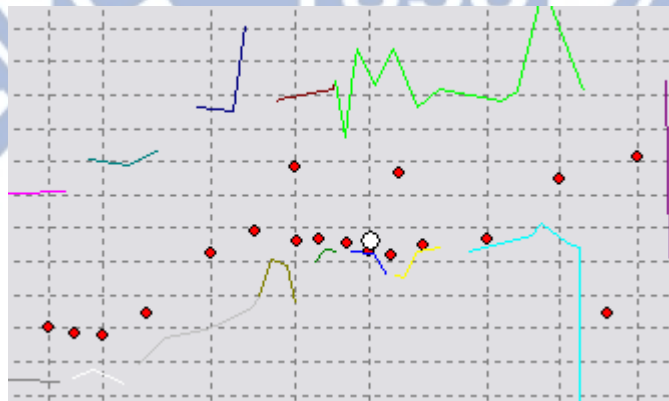
Figure 7.6 An experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.



(a)

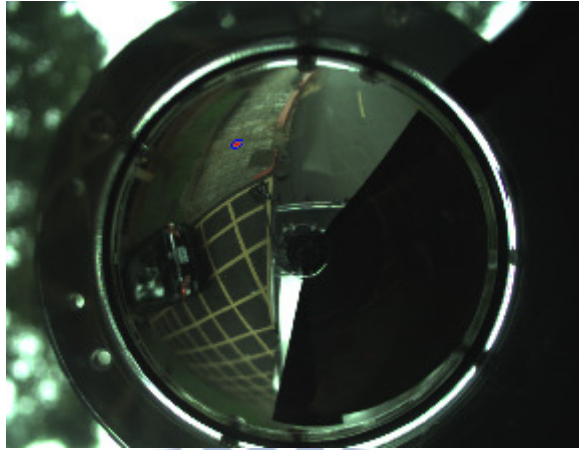


(b)



(c)

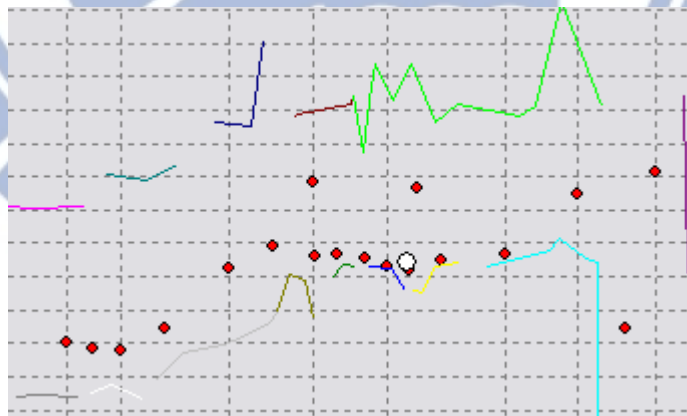
Figure 7.7 Another experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.



(a)



(b)



(c)

Figure 7.8 A third experimental result of guidance tour. (a) The current landmark in the right-front region. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.

Furthermore, the names of the currently-visited buildings displayed on the iPad were projected onto the windshield for the driver to observe. When the video surveillance vehicle passed through the buildings, the names of the buildings can be changed and projected onto the windshield at correct timings. Finally, when the driver met the branch road as shown in the map, we analyzed the turning direction and checked the guidance map graph accordingly. In this way, we obtained the label of the (right, left, or forward) next landmark and traverse the graph correctly. An example of vehicle turning direction determination and guidance map graph updating is shown in Figures 7.9 through 7.11.

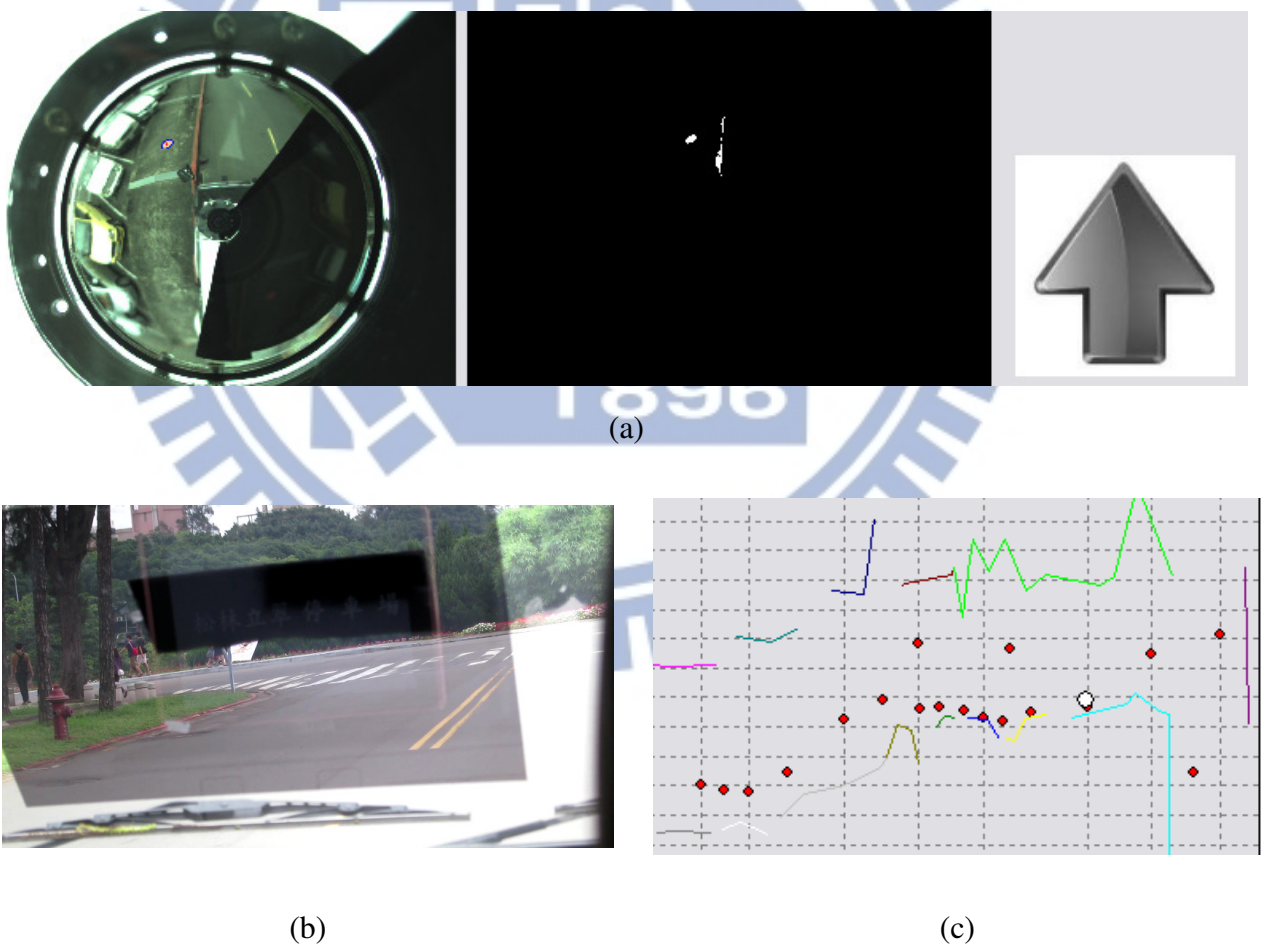
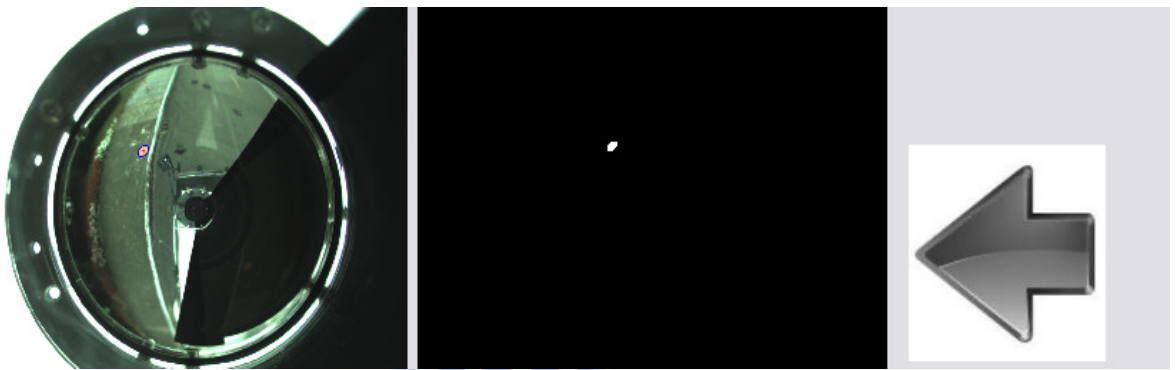


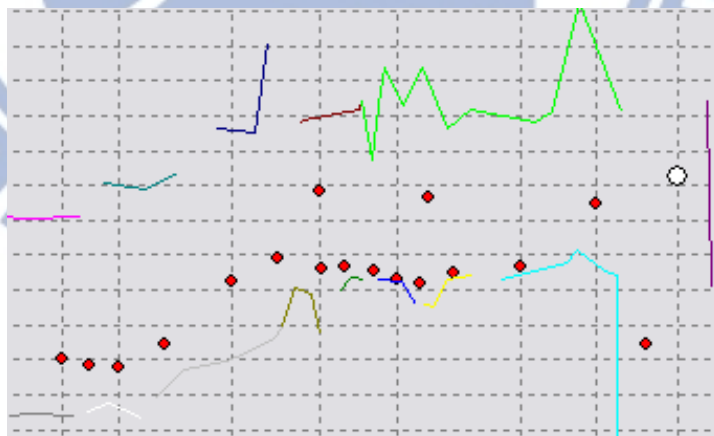
Figure 7.9 An experimental result of vehicle turning determination. (a) The omni-image, the image of detection result, and the decided vehicle direction of positive x -axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.



(a)

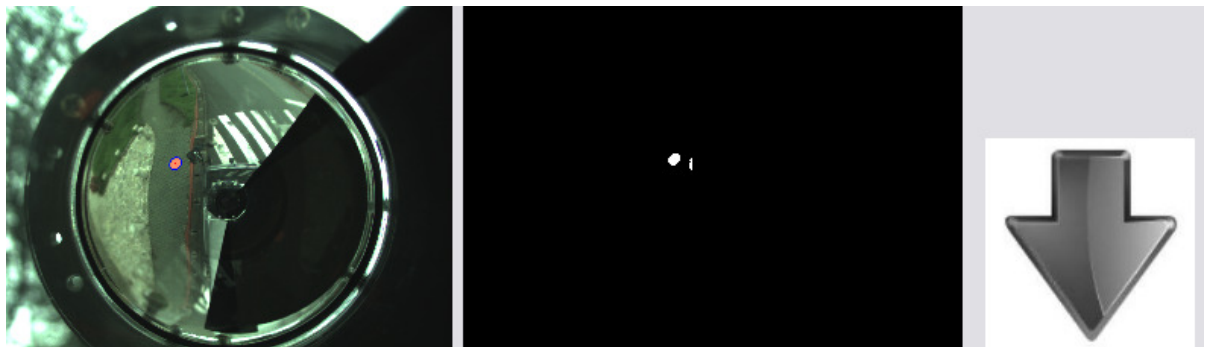


(b)



(c)

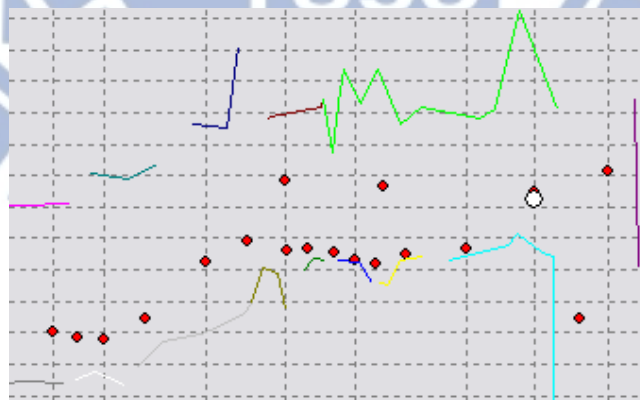
Figure 7.10 Another experimental result of vehicle turning determination. (a) The omnivision image, the image of detection result, and the decided vehicle direction of positive x -axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.



(a)



(b)

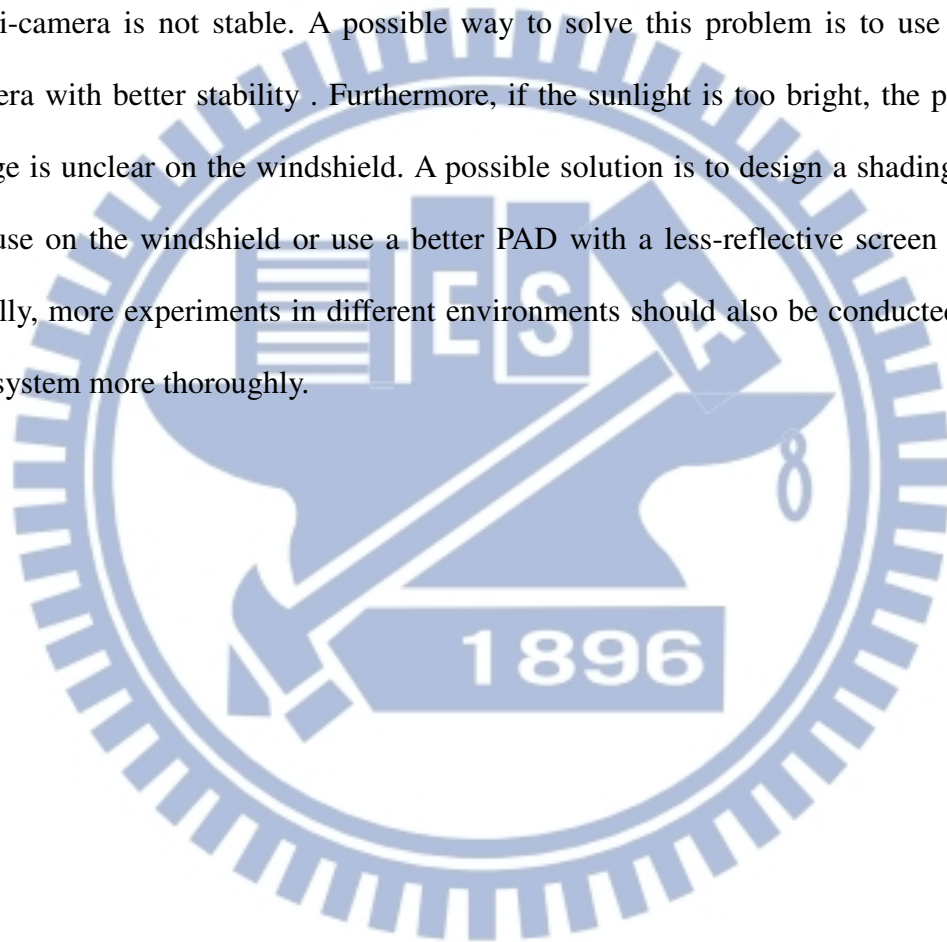


(c)

Figure 7.11 A third experimental result of vehicle turning determination. (a) The omnivision image, the image of detection result, and the decided vehicle direction of positive x -axis. (b) The image of the iPad projected on the windshield. (c) The position of the vehicle on the guidance map.

7.2 Discussions

By analyzing the above-mentioned experimental results of vehicle navigation, we find some problems. Firstly, for landmark detection, we detect the red-surfaced landmarks in the campus of National Chiao Tung University. More kinds of landmarks with different colors or shapes should be tried. Also, the exposure of the omni-camera is not stable. A possible way to solve this problem is to use another camera with better stability . Furthermore, if the sunlight is too bright, the projected image is unclear on the windshield. A possible solution is to design a shading device for use on the windshield or use a better PAD with a less-reflective screen surface. Finally, more experiments in different environments should also be conducted to test our system more thoroughly.



Chapter 8

Conclusions and Suggestions for Future Works

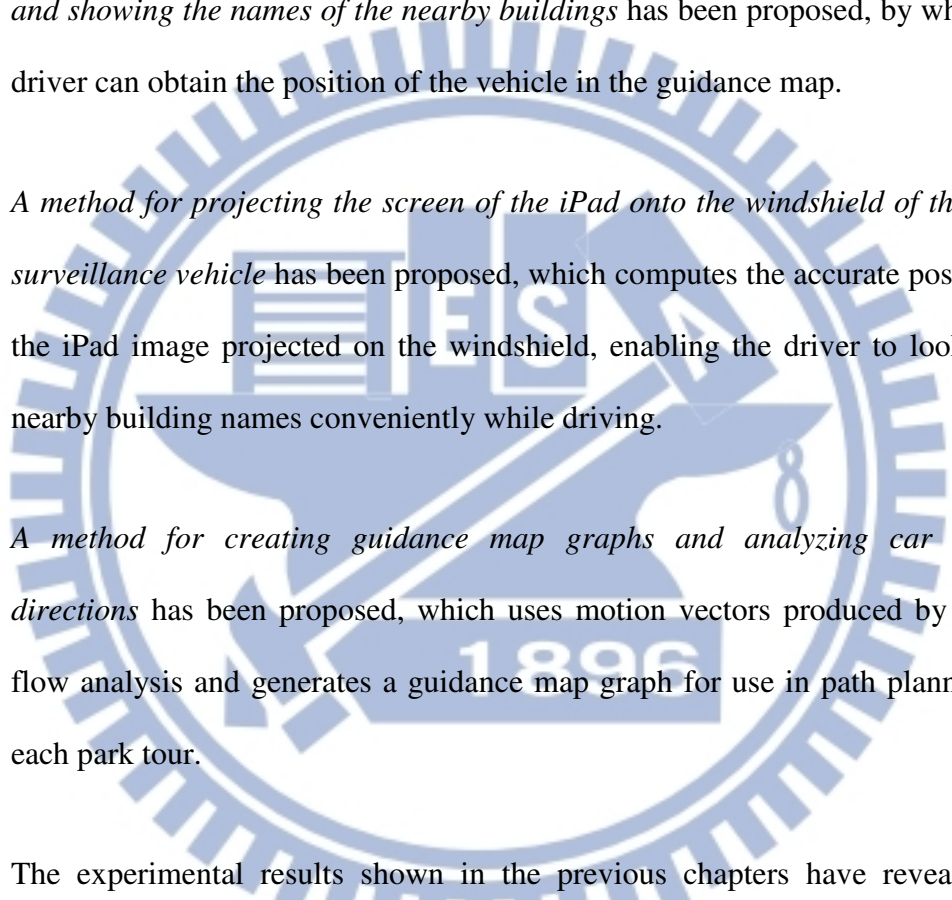
8.1 Conclusions

In this study, an augmented-reality based tour guidance system in park areas using a video surveillance vehicle and computer vision techniques has been proposed. While driving the vehicle in a park, a user can get from the system tour guidance about the names of the buildings appearing along the way on the two roadsides. The building names are displayed on the screen on an iPad which is then projected onto the windshield for the driver to observe. The effect of this projection is equivalent to augmenting the road scene seen by the driver with the projected building names.

A two-camera omni-directional imaging device and a PTZ camera equipped on the vehicle roof are used for acquiring omni-images and PTZ-images for the purposes of vehicle localization and guidance map construction. Vehicle localization is accomplished through 3D image analysis using omni-images acquired of a series of red circular-shaped landmarks attached on flat roadside objects (sidewalks, pedestrian chairs, etc.). Guidance map construction is carried out by use of the PTZ camera for measuring feature points on nearby buildings.

To implement the above functions, several methods have been proposed as summarized in the following.

- (1) *A method for converting the PTZ-camera coordinates into the omni-camera coordinates* has been proposed, which helps creating the global guidance map from the local maps of each landmark easily.

- 
- (2) *A method for detecting red elliptical-shaped landmarks in omni-images* has been proposed, which use the YC_bC_r color model to capture red color in acquired omni-images and detect the elliptical shape in the image by region growing and ellipse fitting.
- (3) *A method for computing the accurate position of the vehicle in the guidance map and showing the names of the nearby buildings* has been proposed, by which the driver can obtain the position of the vehicle in the guidance map.
- (4) *A method for projecting the screen of the iPad onto the windshield of the video surveillance vehicle* has been proposed, which computes the accurate position of the iPad image projected on the windshield, enabling the driver to look up at nearby building names conveniently while driving.
- (5) *A method for creating guidance map graphs and analyzing car turning directions* has been proposed, which uses motion vectors produced by optical flow analysis and generates a guidance map graph for use in path planning for each park tour.

The experimental results shown in the previous chapters have revealed the feasibility and applicability of the proposed system.

8.2 Suggestions for Future Works

According to the experience obtained this study, in the following we make suggestions of some interesting issues, which are worth further investigation in the future.

1. Increasing the speed of computation of landmark detection for real-time

applications, e.g., by parallel computing.

2. Developing the capability of detecting and tracking the landmarks of different shapes.
3. Developing more applications of augmented reality techniques using the omni-camera system and the video surveillance vehicle.
4. Adding the capability of detecting various landmarks on a fast-moving video surveillance vehicle.
5. Enhancing the image analysis capability to detect more information of the vehicle surround, e.g., road lamps, drain covers, etc.
6. Showing the perspective view image of nearby buildings on the iPad and attaching the names of nearby buildings to the perspective image.



References

- [1] Gandhi and M. M. Trivedi, "Motion analysis for event detection and tracking with a mobile omni-directional camera," *ACM Multimedia Systems Journal, Special Issue on Video Surveillance*, vol. 10, no. 2, pp. 131–143, 2004.
- [2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of 7th International Joint Conference on Artificial Intelligence*, Vancouver, Canada, pp. 674 – 679, 1981.
- [3] L. He, C. Luo, F. Zhu, Y. Hao, J. Ou and J. Zhou, "Depth map regeneration via improved graph cuts using a novel omnidirectional stereo sensor," *Proceedings of 11th IEEE International Conference on Computer Vision (ICCV2007)*, pp. 1-8, Oct. 14-21, Rio de Janeiro, Brazil, 2007.
- [4] P. H. Yuan, K. F. Yang, and W. H. Tsai, "A Study on Monitoring of Nearby Objects around a Video Surveillance Car with a Pair of Two-camera Omni-directional Imaging Devices," *Proceedings of 2010 International Computer Symposium (ICS)*, National Chiao Tung University, Hsinchu, Taiwan, pp. 325-330, Dec. 2010.
- [5] S. W. Jeng and W. H. Tsai, "Using pano-mapping tables for unwarping of omni-images into panoramic and perspective-view images," *Journal of IET Image Processing*, Vol. 1, No. 2, pp. 149-155, June 2007
- [6] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, Vol. 13, No.2, pp. 251-263, April 1997.
- [7] C. J. Wu, "New Localization and Image Adjustment Techniques Using Omni-Cameras for Autonomous Vehicle Applications," *Ph. D. Dissertation*, Institute of Computer Science and Engineering, National Chiao Tung University,

Hsinchu, Taiwan, Republic of China, July 2009.

- [8] C. T. Ho and L. H. Chen, "A high-speed algorithm for elliptical object detection," *IEEE Transactions on Image Processing*, Vol. 5, No. 3, pp. 547-550, March 1996.
- [9] 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*
- [10] T. Grosch, "PanoAR: Interactive Augmentation of Omni-Directional Images with Consistent Lighting," *Proc. Computer Vision Computer Graphics Collaboration Techniques and Applications (Mirage '05)*, University of Koblenz-Landau, Germany, pp. 25-34, 2005.
- [11] Lee, J.W., You, S., Neumann, U, "Tracking with Omni-Directional Vision for Outdoor AR Systems," *Proceedings of IEEE ACM Int'l Symposium on Mixed and Augmented Reality (ISMAR 2002), Darrnstadt, Gkrmany, October 2002.*
- [12] G. Reitmayr and T.W. Drummond, "Going out: Robust model based tracking for outdoor augmented reality," *Proc. IEEE Int'l Symp. Mixed and Augmented Reality (ISMAR)*, Santa Barbara, California, USA, pp. 109-118, 2006.
- [13] M. Tonnis, C. Sandor, G. Klinker, C. Lange, and H. Bubb. "Experimen-tal evaluation of an augmented reality visualization for directing a car driver's attention." *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 56-59, Vienna, Austria, Oct. 2005.
- [14] H. Koyasu, J. Miura, and Y. Shirai, "Recognizing moving obstacles for robot navigation using real-time omni-directional stereo vision," *Journal of Robotics and Mechatronics*, Vol. 14, No. 2, pp. 147-156, June 2002.

[15] Intel® IPP, “Image Color Conversion,” Available online:

http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch6/ch6_color_models.html

[16] C. F. Chen, W. H. Tsai, “A Study on Surrounding Environment Monitoring by a Video Surveillance Car with Two 2-camera Omni-imaging Devices,” *Proceedings of 2012 Conference on Computer Vision, Graphics and Image Processing, Chiayi, Taiwan, 2012.*

[17] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” *Proceedings of 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada, pp. 674 – 679, 1981.*

