國立交通大學

多媒體工程研究所

碩 士 論 文

使用一對雙環場攝影機成像系統

對視訊監控車周圍的物體做監測之研究

A Study on Monitoring of Nearby Objects around a Video Surveillance Car with a Pair of Two-camera Omni-directional Imaging Devices

研 究 生：袁佩瑄

指導教授：蔡文祥　教授

中 華 民 國　九 十 九 年 六 月

使用一對雙環場攝影機成像系統

對視訊監控車周圍的物體做監測之研究

A Study on Monitoring of Nearby Objects around a Video Surveillance Car with a Pair of Two-camera Omni-directional Imaging Devices

研 究 生：袁佩瑄　　Student：Pei-Hsuan Yuan

指導教授：蔡文祥　　Advisor：Wen-Hsiang Tsai

國立交通大學

多媒體工程研究所

碩士論文

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

使用一對雙環場攝影機成像系統
對視訊監控車周圍的物體做監測之研究

研究生：袁佩瑄　　指導教授:蔡文祥 博士

國立交通大學多媒體工程研究所

摘要

本論文提出了一個以視覺為基礎的視訊監控方法，該方法使用了兩組架設於監控車車頂的雙環場攝影機成像系統。首先，我們使用空間對應的方法建立起此環場攝影機成像系統的校正資訊表，稱之為全景對應表。基於此對應表以及兩個環場影像的對應點，我們提出了影像座標點與世界座標點之間的轉換方法。為了觀察監控車周遭的環境，我們另提出了建構上視圖的方法，以及將兩張上視圖合併成一張寬廣視角圖的技巧。此外，我們也使用一區域網路作兩台筆記型電腦的溝通管道，供傳輸移動指令，讓使用者能移動滑鼠即可建構出各種視角的透視圖。

另一方面，我們提出了一個自動偵測可疑人物並將其標記在上視圖的方法，這方法使用了一些影像處理的技巧，如距量保存影像二值化和灰階動態補償等。再者，利用一組雙環場攝影機成像系統所拍攝的影像組，我們能計算出可疑人物於立體空間中的距離及高度。如果監控車中的使用者想更直接的觀察可疑人物，他/她也能使用此系統而不需要走出車外即能觀察所建構出的對應透視圖。另外，我們也提出了一個自動偵測車輛的方法，在該法中，為了移除影像中的地板區域以及擷取出環場影像中的車體，我們使用了區域生長、組件標籤、圖像變換、模板匹配等影像處理技巧，並將之有效整合來獲得環場影像中車子的位置。最後，使用一組雙環場攝影機機成像系統所拍攝的影像組，我們可計算出車子位於真實世界座標中的位置。

良好的實驗結果顯示我們所提出視訊監控系統的可行性及應用靈活性。

A Study on Monitoring of Nearby Objects around a Video Surveillance Car with a Pair of Two-camera Omni-directional Imaging Devices

Student: Pei-Hsuan Yuan Advisor: Prof. Wen-Hsiang Tsai

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

ABSTRACT

Vision-based methods for video surveillance via the use of a pair of two-camera omni-directional imaging devices affixed on the roof of a video surveillance car are proposed. First, a space mapping method is used to construct the so-called pano-mapping tables of the pairs of two-camera omni-directional imaging devices. By the mapping tables and corresponding points of two omni-images, a method for converting the coordinates of the points between the image coordinate system and the world coordinate system is proposed. To see the environment around the video surveillance car, techniques for constructing top-view images and merging them into wider-area integrated ones are proposed. Also, a local network architecture for data communication between two laptop PCs, as well as a technique for constructing perspective-view images of any view direction decided by mouse clicks are proposed.

Furthermore, a method for detecting a suspicious passer-by automatically and marking his/her position on a top-view image is proposed, which is based on image processing schemes of moment-preserving thresholding and dynamic grayscale

offsetting. Moreover, the distance and height of a passer-by in 3D space is computed by image pairs taken with a two-camera omni-directional imaging device. If a user in the surveillance car wants to see a detected suspicious passer-by directly, he/she may use the system to generate a corresponding perspective-view image to inspect the suspicious passer-by without going out of the car. Additionally, a method of detecting a passing-by car automatically is proposed. To eliminate the ground region and capture the passing-by car shape in the omni-image, image processing techniques like region growing, component labeling, image transformation, template matching, etc. are used integrally and effectively to get the accurate position of the passing-by car in an omni-image. Finally, the position of the passing-by car in the real world is estimated as well using image pairs taken with a two-camera omni-directional imaging device.

Good experimental results show the flexibility and feasibility of the proposed methods for the application of video surveillance.

ACKNOWLEDGEMENTS

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

Thanks are due to Mr. Guo-Feng Yang, Mr. Che-Wei Lee, Mr. Yi-Fu Chen, Mr. Bo-Jhih You, Mr. Jheng-Kuei Huang, Mr. Chih-Hsien Yao, Miss. Mei-Hua Ho, Miss I-Jen Lai for their valuable discussions, suggestions, and encouragement. The author especially thanks Yen-Han Chou for the help in many experiments. 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 her thesis study.

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

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Motivation

As the computer technology progresses quickly nowadays, video cameras are used widely in various surveillance systems, not only to prevent crimes or disasters but also to improve machine automation for human beings' welfare. The video which is taken by a camera can be recorded forever, and useful information in the video can be extracted for *a posteriori* investigation. For example, when a car accident occurs or a demonstration is in progression, if a *video surveillance car* is available aside, the recorded video of the on-going event can be inspected simultaneously or later to find out the cause of the accident or any person who conducted violent activities in the demonstration. Moreover, this way of event recording can provide a better evidence of the possible crime than just relying on the availability of a passenger's memory of the event's detail.

In addition, most existing vision-based techniques using traditional cameras only allow recording and processing of *frontal scenes*, i.e., scenes that are seen in front of the video surveillance car. To improve this shortage, it requires four or more traditional cameras to get a complete coverage of the entire environment around the car. In order to enlarge the fields of view (FOV's) of traditional cameras, a feasible solution is to use *omni-directional cameras* (or simply, *omni-cameras*). In this study, a pair of two-camera omni-directional imaging devices is used. Each device consists of two axis-aligned omni-cameras. A wide and complete view of the environment around a video surveillance car can be covered by such a camera system.

Moreover, most of the vision-based systems are affixed to some places like ceilings or utility poles, each monitoring a specified area around for security screening. In order to increase the mobility of the surveillance area, in this study it is desired to set up a wide-area vision-based surveillance system on the roof of a video surveillance car using the previously-mentioned pair of two-camera omni-directional imaging devices. Once an event like a violent demonstration arises, a police team, for example, may drive the surveillance car to the spot of the event and monitor the event immediately. Such a system may also be utilized to detect suspicious people and vehicles around, and estimate the distance and height information of each approaching passer-by in suspect without walking out the car to measure relevant information manually. This is also a good way to avoid direct contacts between the police team and suspected people.

In addition, about the coverage of the monitored range using a common camera (like a projective one), there usually exist *blind areas* around a common car which cannot be covered by the camera's FOV. Therefore, people who are in the car cannot fully watch and understand the environment or the event occurring out there. Furthermore, human eyes can only see a scene with a certain angle span in a single direction at a time, such as a view of the front, the left side, the right side, or the rear of the car. Therefore, a vision-based video surveillance system capable of providing a vehicle user to see the environment outside the car in any direction clearly, dynamically, and conveniently is desirable. It is also advantageous for the user to see easily the top-view image of the surrounding of the video surveillance car and the perspective-view image of any specified direction without turning around his/her body.

As a summary, the research goal in this study is to develop a vision-based video surveillance system on the top of a video surveillance car with a pair of two-camera

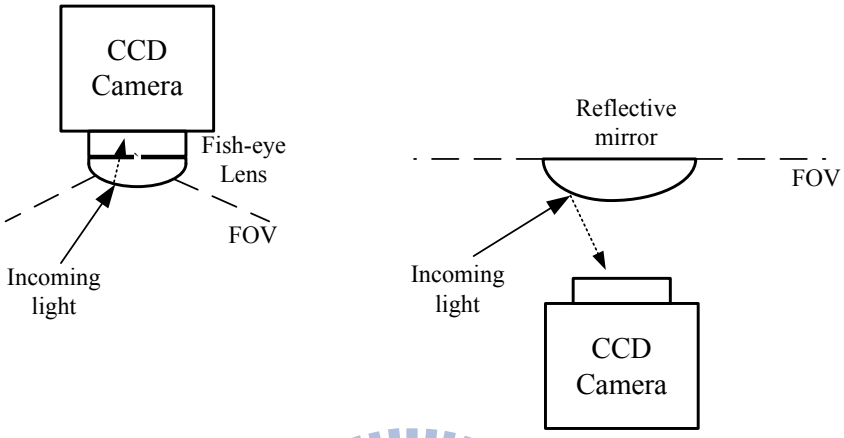
omni-directional imaging devices which has the following capabilities:

1. constructing a top-view image of the surrounding of the video surveillance car which comes from merging the two omni-images taken by the upper cameras in the pair of two-camera omni-directional imaging devices (there are two such devices on the top of the surveillance car used in this study, and each device consists of two omni-cameras, an upper one and a lower);
2. constructing a perspective-view image of any direction specified by the car user (the driver or any person in the car), with the view direction determined by mouse clicking or computer panel touching;
3. detecting any suspicious passing-by person automatically, measuring his/her position and height, and marking the result as a moving highlighted spot on the top-view image; and
4. detecting any passing-by car automatically, measuring its position, and marking it on the top-view image.

1.2 Survey on Related Studies

In this study, we use a pair of two-camera omni-directional imaging devices on the top of a video surveillance car to take omni-directional images and estimate relevant 3D data of surrounding objects (passers-by and cars). Omni-cameras can be categorized into two types, *dioptric* and *catadioptric*. A dioptric omni-camera captures incoming light with a wide-angle lens. An example of this kind of omni-camera is the *fish-eye* camera. Some works of vehicle surrounding monitoring and indoor security surveillance using fish-eye cameras can be found in Liu, Lin, and Chen [1] and Chen and Tsai [2]. In contrast, a catadioptric omni-camera captures incoming light reflected by a built-in reflective mirror. The mirror surface of a catadioptric omni-camera may

be made to be of various shapes, such as conic, parabolic, hyperbolic, etc. An illustration of the different structures and FOV's of the two kinds of cameras is shown in Fig. 1.1.



(a) (b)

Figure 1.1 Structures and FOVs of different omni-camera types [Jeng, 3]. (a) Dioptric camera. (b) Catadioptric camera

In this study, we design an imaging device which is composed of two catadioptric cameras in longitudinally coaxial alignment to derive stereo environment information. Most research works used such omni-camera pairs with *hyperbolic-shaped* reflective mirrors, such as Koyasu et al. [4] and Ukida et al. [5]; only a few works used such omni-camera pairs with *parabolic-shaped* reflective mirrors, such as Gluckman et al. [6]. The omni-camera pairs forming an imaging device which we design for use in this study will be elaborated in Chapter 2.

Traditionally, if the intrinsic and extrinsic parameters of the projective camera can be calibrated by a well-known and mature technique, e.g., Salvi et al. [7], stereo information can be obtained using the camera parameters of the omni-camera pairs (see Koyasu et al. [4]). In this study, a space-mapping technique proposed by Jeng and Tsai [8] is used to calibrate the omni-camera without calculating the intrinsic and

extrinsic parameters. In the mean time, there are some image adjustment techniques or calibration methods for adjusting misaligned catadioptric cameras or modifying the image taken by them, e.g., Wu and Tsai [9] and Mashita et al. [10].

Video surveillance has been widely investigated in the past decade. Most research works are about indoor surveillances. Onoe et al. [11] and Mituyosi et al. [12] conducted researches for such a purpose using omni-cameras, both proposing video surveillance systems may be used to monitor passing-by persons and mark their positions on the images. Besides, a video surveillance system using multiple omni-directional cameras was proposed by Morita et al. [13], and it has a capability to detect the positions of moving objects.

In spite of these works, a new capability to obtain pass-by persons' height as implemented in this study is not found in the previously-mentioned works. Furthermore, the two-camera omni-directional imaging device we use can be used to take images to compute 3D data, and this was not found in the previously-mentioned works, either.

Matuszyk and Zelinsky [14] proposed a method to monitor blind-spots around vehicles using a two-camera omni-directional imaging device with the camera axes aligned horizontally. A wider coverage was obtained because of the designed camera structure. Furthermore, a study that reconstructs 3D data of static near-by vehicles by a mobile robot using a stereo omni-directional camera (a two-mirror omni-directional imaging device) was proposed by Meguro et al. [15]. However, the proposed method for 3D space reconstruction is time-consuming, and this is undesirable in real-time video surveillance. In this study, we propose a new method to detect a passing-by car and mark its position on a top-view image without spending so much time.

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-directional camera:* a camera consists of a reflective mirror and a traditional camera with a 360-degree field of view in the horizontal plane.
2. *Omni-image*: an image which is captured with an omni-directional camera.
3. *Two-camera omni-directional imaging device:* an imaging system consisting of two omni-cameras coaxially connected in the longitudinal direction, whose detailed configuration will be described in Chapter 2.
4. *Perspective view:* an image of a scene projected onto a flat surface (such as paper) as it is seen by the eye from any direction.
5. *Top view:* an image of a scene projected onto a horizontal plane (with only *x*- and *y*-axis information) as it is seen from the top.
6. *Video surveillance car:* a car which is equipped with a pair of two-camera omni-directional imaging devices to take video images for the purpose of security surveillance of the environment around the car.
7. *Passing-by car:* a car which comes to pass by the video surveillance car.
8. *Passer-by or passing-by person:* a person who comes to pass by the video surveillance car.

1.3.2 Brief Descriptions of Proposed Approach

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

1. A passing-by car should be detected automatically and its position can be obtained.

2. A top-view image of the area around the video surveillance car should be made available to an in-car user who wants to monitor such a large area.
3. The monitored area should cover as much of the surrounding environment of the video surveillance car as possible.
4. A suspicious passer-by's distance and height with respect to the video surveillance car should be detected and his/her position in 3D space be marked on the top-view image.
5. The perspective-view image in the suspect's direction with respect to the car should be made available to the user for a clearer inspection.

In order to achieve the previously-mentioned goals, the following major steps are adopted in this study, assuming that a video surveillance car is available:

1. design a two-camera omni-directional imaging device, which consists of an upper omni-camera and a lower one to obtain or estimate relevant 3D data;
2. construct a pair of the imaging devices just mentioned, and equip them on the roof of the video surveillance car, with one on the front-right corner and the other on the rear-left corner, of the car roof;
3. calibrate each of the four omni-directional cameras in the two imaging devices on the car roof using a space-mapping technique proposed by Jeng and Tsai [8];
4. unwarp the omni-image taken by any of the four omni-cameras into perspective-view images as specified by an in-car user;
5. transfer images by a local network between two processing units (notebook PC's in this study) used for processing images taken by the pair of two-camera omni-directional imaging devices, respectively;
6. calculate the relative positions of the pair of two-camera omni-directional

imaging devices;

7. integrate into a top-view image the two top-view images which are acquired from the two *upper* omni-cameras of the two imaging devices;
8. detect any suspicious passer-by around the monitored neighborhood of the video surveillance car and show it on a display unit;
9. detect any passing-by vehicle around the monitored car's neighborhood and show it on a display unit.

Each problem above and the solution proposed for it in this study will be described in detail in the following chapters.

1.4 Contributions

Some contributions made by this study are listed in the following.

1. For the first time multiple omni-directional camera pairs are integrated and used for video surveillance applications.
2. An integrated method using the pano-mapping table based on a space-mapping technique [8] to estimate the 3D data using a two-camera omni-directional imaging device is proposed.
3. A transformation method to convert an omni-image into multiple perspective-view images is derived.
4. A method is proposed to combine two omni-images taken by two separate omni-cameras to obtain an overall top-view image around the video surveillance car.
5. A method of passer-by detection with a two-camera omni-directional imaging device is proposed.
6. A method of passing-by car detection which comprises shape identification

and 3D data extraction is proposed.

7. A local network is constructed, which integrates a pair of two-camera omni-directional imaging devices and two laptop computers for video surveillance use in this study.
8. A method is proposed to solve problems of object detection in images caused by varying light intensities in different weather conditions.
9. A method is proposed to detect the transformed video surveillance car shape from the top-view image, to remove it from the image, and to fill in a graphic car model as a substitute at the correct position, for a better inspection of the image.

1.5 Thesis Organization

In the remainder of this thesis, the system configuration and the idea of the proposed method are introduced in Chapter 2. In Chapter 3, the technique of using the pano-mapping table for unwarping of omni-images into multiple perspective-view images is described. In Chapter 4, the proposed automatic method for detection of a suspicious passer-by with a two-camera omni-directional imaging device is described. In Chapter 5, the proposed method for integration of the two omni-images taken respectively with the two upper omni-cameras into a single top-view image is presented. In Chapter 6, the proposed automatic method for detection of a passing-by car with a two-camera omni-directional imaging device is presented. In Chapter 7, experimental results and discussions are included. Finally, conclusions and some suggestions for future works are given in Chapter 8.

Chapter 2
System Configuration, Camera Design, and Idea of Proposed Method

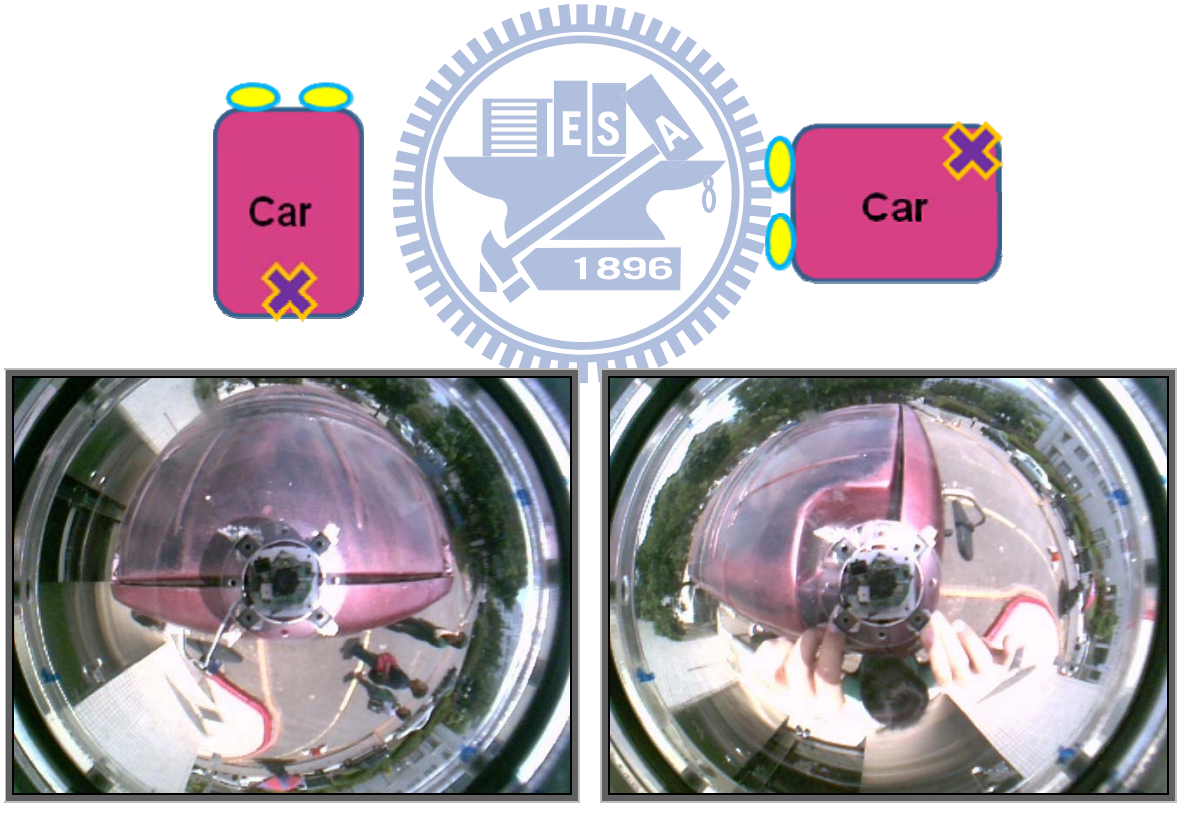
2.1 Idea of Proposed Monitoring of Nearby Objects around a Mobile Surveillance Car

In order to monitor the surrounding area of the video surveillance car, we equip on the roof of the car a pair of two-camera omni-directional imaging devices constructed in this study, as shown in Figure 2.1. More details about the devices and the proposed idea of using them are described here.

(a) (b)

Figure 2.1 The video surveillance car used in this study is equipped with a pair of two-camera omni-directional imaging devices. (a) A front view of the video surveillance car. (c) A side view of the video surveillance car.

First, the locations on the car roof where the two imaging devices should be affixed need to be determined. In Figure 2.2, it is illustrated that the video surveillance car body almost accounts for a half of an omni-image taken by an imaging device which is affixed at the middle of the rear edge of the car roof, but it only accounts for a quarter of an omni-image taken by the same imaging device which instead is affixed at the right-rear corner of the car roof. Consequently, an imaging device affixed at a corner of the car roof will have a better view range than one affixed at the front (or back) middle of the car roof. In this study, one of the imaging devices is so affixed at the right-front of a surveillance car roof, and the other is affixed at the left-rear.



(a) (b)

Figure 2.2 Positions of cameras on the video surveillance car roof and the corresponding images of them. (a) The image captured at the rear-middle of the video surveillance car roof. (b) The image captured at the right-rear of the car roof.

The imaging devices, after being affixed, can be used to estimate relevant 3D data of objects (the detail will be described in Chapter 2.3) Then, an integrated top-view image can be obtained to view the surrounding environment of the video surveillance car from the top (the detail will be described in Chapter 5). Also, any passers-by can be detected automatically and be marked on the top-view image (the detail will be described in Chapter 4). If a user wants to see a suspicious passer-by directly, a corresponding perspective-view image may be generated for inspection (the detail will be described in Chapter 3). An example of the top view and a generated perspective view is shown in Figure 2.3.

Furthermore, passing-by cars can also be detected and marked on the top-view image by algorithms proposed in this study, such as region growing, template matching, etc. (the detail will be described in Chapter 6).

(a) (b)

Figure 2.3 Images of monitoring a passer-by. (a) Top-view image showing surrounding area of the video surveillance car with red mark indicating the passer-by's position. (b) A corresponding perspective-view image containing the passer-by.

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 the video surveillance car, a pairs of two-camera omni-directional imaging devices, and two laptop computers. The software includes the program used to integrate the vision-based system, the drivers of the omni-cameras, and the program developed by the hhARTRAY Company which is a provider of CCD cameras. However, each two-camera omni-directional imaging device is controlled by a laptop computer, so we construct a local network configuration to handle the problem of communication.

2.2.1 Hardware configuration

The entire hardware structure of the proposed video surveillance system used in this study is shown in Figure 2.4. The video surveillance car we use is named *Delica* made by Mitsubishi Co. It is a 469cm×169cm×196cm van with a working desk and a power supply designed especially for this study. Its capacity is eight people. For the purpose of connection between four omni-cameras outside the video surveillance car and the two computers inside, four extension cords are used to cross the video surveillance car.

Each of the two-camera omni-directional imaging devices affixed on the video surveillance car roof includes two omni-cameras, and each of the omni-cameras is composed of a lens, a CMOS camera, a mirror, an acrylic tube, and a shelf. The detailed descriptions of these imaging devices, the parameters of the mirrors, and the optical principle of the imaging devices will be described in detail in Chapter 2.3.

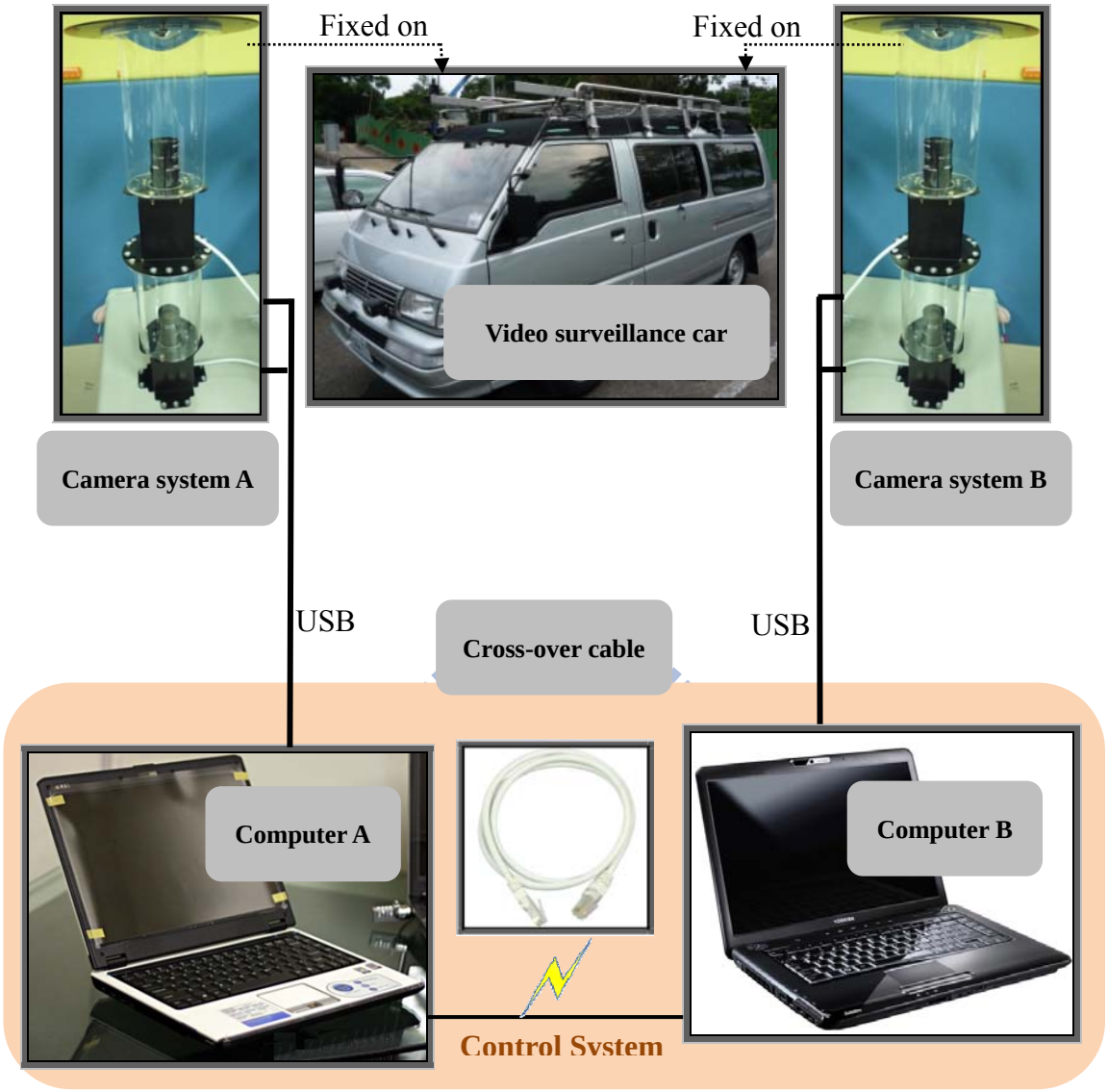


Figure 2.4 Structure of the proposed monitoring system.

As to the control unit, two notebook PC's are used to integrate the entire video surveillance system. In Figure 2.4, *Computer A* is a *F6E* laptop computer produced by ASUSTeK Computer Inc., and *Computer B* is an *A300* laptop computer produced by Toshiba Co. The performance specifications of these computers are shown in Table 2.1. The cross-over cable used for communicating two computers is a *Cat-6* cable for the gigabit Ethernet.

Table 2.1 Specifications of the used laptop computers.

	ASUS F6E	TOSHIBA A300
CPU	Intel Core 2 Duo T5850/ 2.16 GHz	Intel Core 2 Duo T9400/ 2.53 GHz
Chipset	Intel PM 965	Intel GM965
RAM	4 GB DDR2 / 667 MHz	2 GB DDR2 / 800 MHz
GPU	IntelGMA X3100	ATIRadeon HD 3650 / 512 MB
Network	Gigabit LAN	Fast Ethernet LAN

2.2.2 Software configuration

We use the Borland C^{++} builder as the development tool in this study to acquire omni-images and analyze them. It is fast and convenient to develop a GUI-based program using the Borland C^{++} builder. The programming language we use is C^{++}, a widely-used language. The operating system we use is Window XP.

To access the images taken by the cameras, the computers have to set up the drivers of the *ARTCAM-200SO* cameras and the *ARTCAM-200SS* cameras. The Artcam Co. provides a development tool called *Capture Module Software Developer Kit* that assists developers in communication with the embedded system of the camera, using a USB connection. In addition, the SDK is an object-oriented toolkit and usable under Windows 2000 or XP in many languages like C^{++}, C, VB.NET, C#.NET and Delphi. Using the SDK, we can preview the image of each camera's view and capture the current image data. It is also convenient to use it to develop any function with images grabbed with the cameras as input.

2.2.3 Network configuration

A network configuration is needed for communication between the two laptop computers because four omni-images are acquired from the two pairs of two-camera omni-directional imaging devices (CS_A and CS_B in Figure 2.5), and each imaging device is processed by a separated notebook PC (COM_A and COM_B in Figure 2.5). The network we propose for this is shown in Figure 2.5.

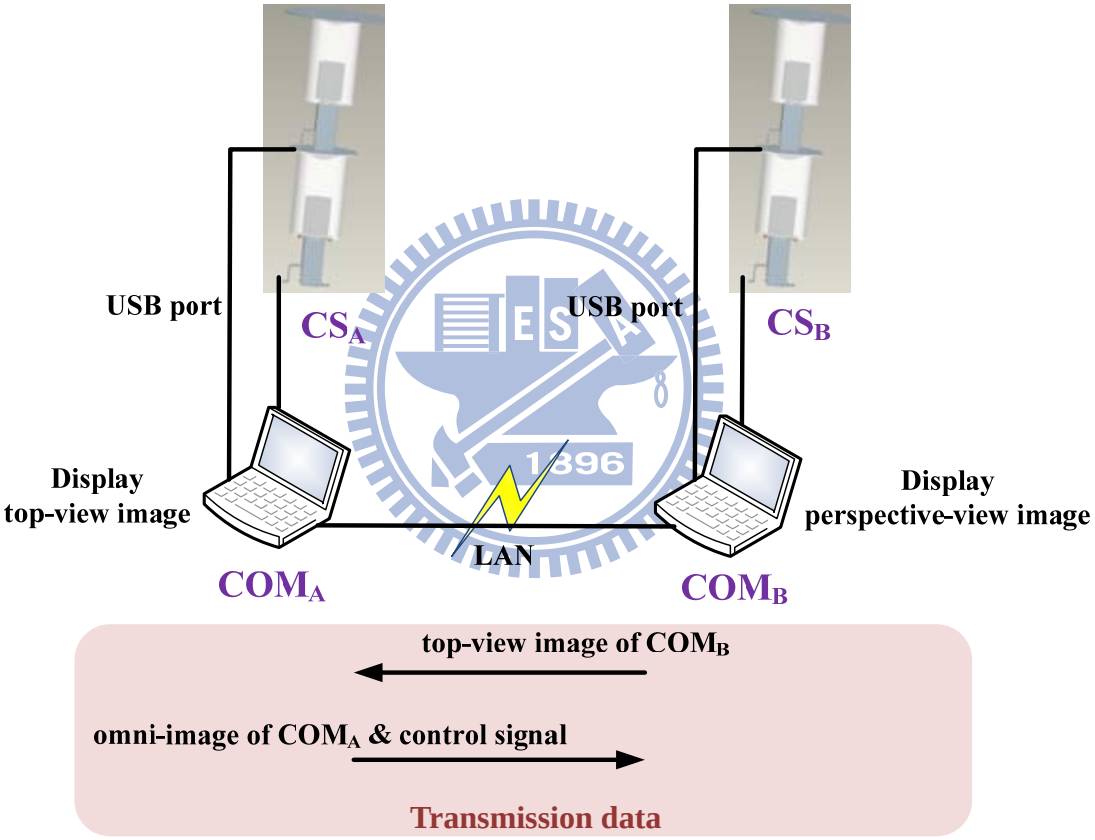


Figure 2.5 The entire proposed system and the network architecture of transmission.

As shown, COM_A is used to display the top-view image of the surrounding area of the video surveillance car, and COM_B to display the perspective-view images in a specified direction of the car. Therefore, COM_B transforms the omni-images gathered from CS_B into a top-view image and transmits the result to COM_A which then merges

the two top-view images (one by CS_A and other by CS_B) into an integrated top-view image of the car surrounding. On the other side, COM_A transmits the omni-image gathered from CS_A and a control signal to COM_B, so that COM_B knows the view's direction and constructs the corresponding perspective-view image.

2.3 Design of a Pair of Two-camera Omni-directional Imaging devices

2.3.1 System configuration

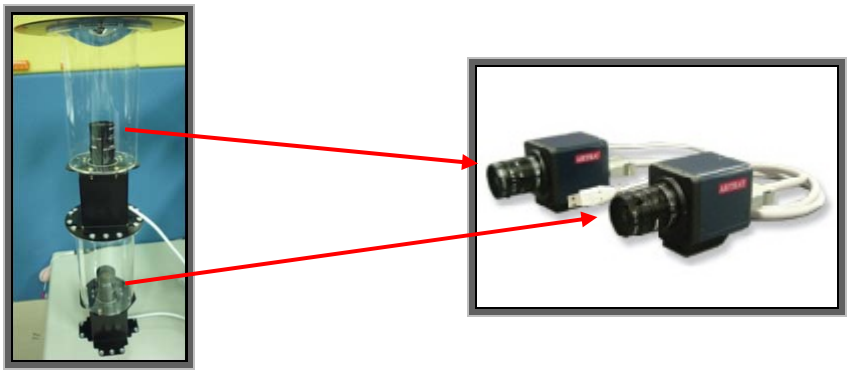
Each of the two-camera omni-directional imaging devices consists of two omni-cameras combined coaxially in the longitudinal direction, as shown in Figure 2.6(a). The entire system includes four lenses of model *LV0612H*, two CMOS cameras of model *ARTCAM-200SO*, and two CMOS cameras of model *ARTCAM-200MI*. Two lenses of the four and the two *ARTCAM -200SO* CMOS cameras are shown in Figure 2.6(b).

The *LV0612H* is a mega-pixel lens with the following arguments: 1/2", 6mm, and F1.2. The specifications of the COMS cameras are shown in Table 2.2. *Camera system A* in Figure 2.4 is formed with the two ARTCAM-200SO cameras, and affixed on the right-front of the video surveillance car roof. *Camera system B* is formed with the two ARTCAM-200MI cameras, and affixed on the left-rear of the video surveillance car roof.

2.3.2 Camera Design Principle

To explain how we design the omni-camera used in our omni-directional

imaging devices (there are four of this kind of camera in our system), we derive the related formulas in the following first.



(a) (b)

Figure 2.6 (a) Two-camera omni-directional imaging device. (b) Two lenses and two *ARTCAM-200SO* CMOS cameras.

Table 2.2 Specifications of the used COMS cameras

	ARTCAM-200SO	ARTCAM-200MI
Resolution	2.0 M pixels	2.0 M pixels
Dimension	33mm × 33mm × 50mm	33mm × 33mm × 50mm
CMOS sensor size	1/2” (6.4×4.8mm)	1/2” (6.4×4.8mm)
Mount	C-mount	C-mount
Frame per second	8 fps	5 fps
Direct show camera	Yes	No

The structure of each omni-camera with a hyperbolic-shaped mirror is illustrated in Figure 2.7, with the world coordinate system (WCS) specified by (X, Y, Z). The hyperbolic shape of the mirror in the camera coordinate system may be described [8] as:

$$\frac{R^2}{a^2}-\frac{Z^2}{b^2}=-1, \qquad R=\sqrt{X^2+Y^2}, \tag{2.1}$$

where a and b are the parameters of the hyperbolic shape. The parameter d, as shown in Figure 2.7(b), is the distance between the optical center of the lens and the mirror center, whose value can be obtained by a simple formula $d = 2c$ where $c=\sqrt{a^2+b^2}$. Also, it is noted that the axis of the camera is aligned with the axis of the hyperbolic mirror, and the camera center is fixed at one of the two focal points of the mirror.

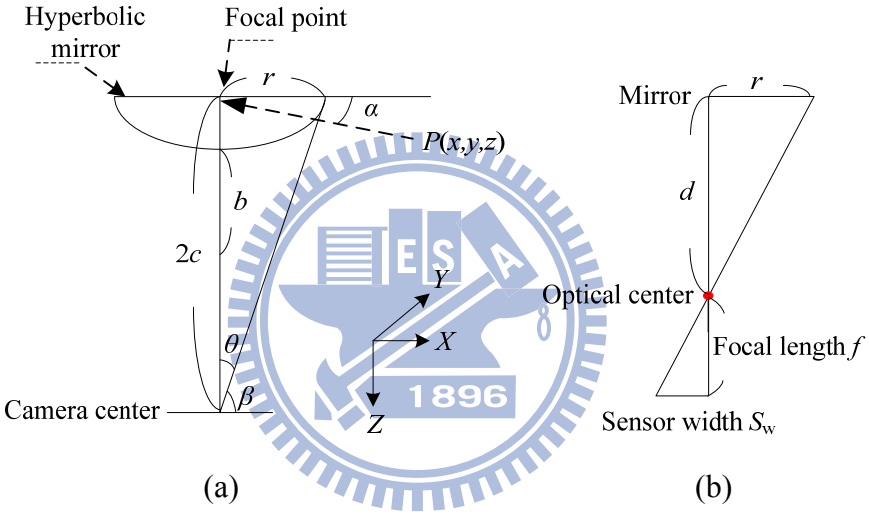


Figure 2.7 An illustration of used omni-camera structure. (a) Geometry of the omni-camera vision. (b) Geometry between the mirror and the CMOS sensor in camera.

By the geometry of the shape of a hyperboloid described by Eq. (2.1), the value ρ, which specifies an elevation angle shown in Figure 2.7 (a), can be computed by the following formula:

$$\tan\alpha=\frac{(b^2+c^2)\sin\beta-2bc}{(b^2-c^2)\cos\beta}. \tag{2.2}$$

Furthermore, the angles θ and β in Figure 2.7 (a) can be computed as follows:

$$\theta = \tan^{-1}\frac{r}{2c}; \tag{2.3}$$

$$\beta = \frac{\pi}{2} - \theta. \tag{2.4}$$

In Figure 2.7 (b), by trigonometry, we have

$$\frac{d}{r} = \frac{f}{S_w}, \tag{2.5}$$

where, f is a focal length, r is the radius of the circular area of the base of the mirror, and S_w is the width of a CMOS sensor.

Now we can explain how we design the omni-cameras we use in this study according to the above theoretical derivations. The goal is to design a mirror of the hyperbolic shape and determine the distance from the camera to the mirror. Specifically, we have to derive the parameters, a, b, and c, of the hyperbolic shape so that we can ask an optics manufacturer to produce a mirror of such parameters for us. Note that the distance from the camera to the mirror, denoted as d above, is just $2c$ because we put the camera at such a position that its optical center of the lens is located just at a focal point of the hyperbolic shape, as shown in Figure 2.7.

Because the projective camera we use has a focal length f of 6 mm and a sensor width S_w of 2.4mm, and because the circular area of the base of the mirror has a radius r of 4 cm, according to Eq. (2.5) and $d = 2c$, we can derive d and c as

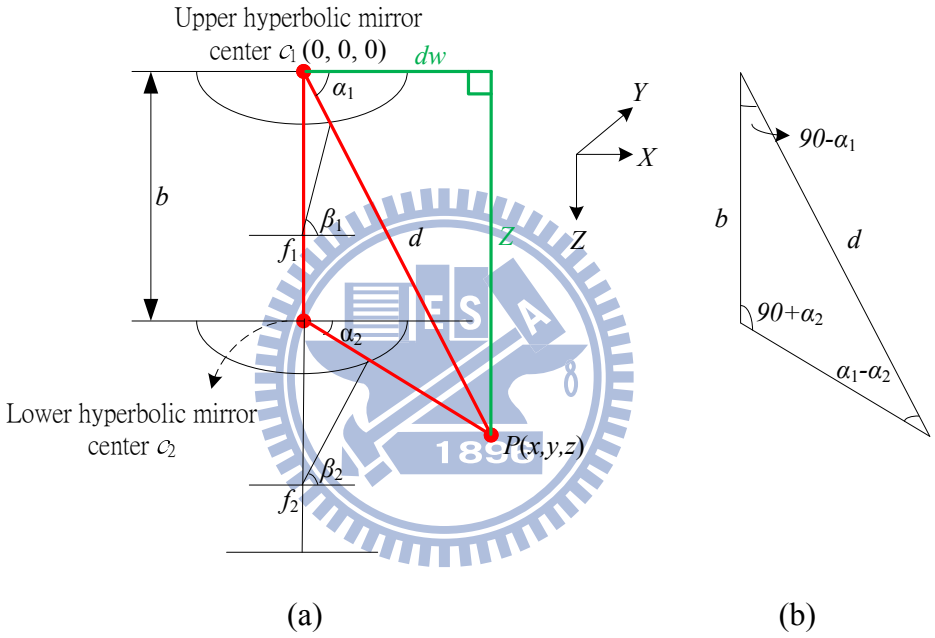
$$d = \frac{f}{S_w} \times r = 10\,cm, \quad c = 5\,cm. \tag{2.5}$$

Also, according to Eqs. (2.3) and (2.4), the values of the angles θ and β can be computed to be $\theta = 0.3805$ and $\beta = 1.1902$. In Eq. (2.2), we can assume $\alpha = 0$, and by using Eq. (2.4), we can reduce Eq. (2.2) to be the following equation with only one variable b:

$$(b^2+25)\times 0.9287-10b=0 \tag{2.6}$$

from which b can be solved to be b = 3.3851. And by c = 5 = $\sqrt{a^2+b^2}$, a can be solved to be 3.6797. Thus, the parameters of the hyperbolic mirror designed in this study are all obtained, that is, a = 3.6797 and b = 3.3851.

2.3.3 3D data acquisition



(a) (b)

Figure 2.8 Computation of depth 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)).

In this section, it will be briefly described how to use two elevation angles of a scene point P to get relevant 3D data. Note that these elevation angles can be obtained by using a pano-mapping table (it will be described in Chapter. 3). Specifically, as shown in Figure 2.8(a), each image point P is a projection of a corresponding point on the hyperboloid, which can be defined by the elevation angles α_1 and α_2. The upper hyperbolic mirror center is assumed to be the WCS center (0, 0, 0). The desired goal

is to use α_1 and α_2 to get (x, y, z).

In Figure 2.8(b), by the triangulation principle, the distance d between the scene point P and the center of a hyperbolic-shaped mirror c_1 may be computed as

$$\frac{d}{\sin(90+\alpha_2)} = \frac{b}{\sin(\alpha_1-\alpha_2)}, \tag{2.6}$$

where b is the disparity of the stereo imaging device. In the system we proposed, b = 24.2 cm. Eq. (2.6) can be reduced to the following equation:

$$d = \frac{1}{\cos\alpha_1} \times \frac{1}{\tan\alpha_1 - \tan\alpha_2} b, \tag{2.7}$$

and the horizontal distance dw and vertical distance Z in Figure 2.8 (a) may thus be computed by:

$$\begin{aligned} dw &= d\cos\alpha_1 = \frac{1}{\tan\alpha_1 - \tan\alpha_2} b; \\ Z &= d\sin\alpha_1 = \frac{\tan\alpha_1}{\tan\alpha_1 - \tan\alpha_2} b. \end{aligned} \tag{2.8}$$

A system configuration of the upper omni-camera with a hyperbolic-shaped mirror is shown in Figure 2.9, with the WCS specified by (X, Y, Z) and the image coordinate system (ICS) specified by (U, V). The $I(u, v)$ is an image point projected by a scene point $P(x, y, z)$.

A triangulation which includes the angle θ in Figure 2.9 can be described by the pixel coordinates (u, v) as follows:

$$\sin\theta = \frac{v}{\sqrt{u^2+v^2}}; \cos\theta = \frac{u}{\sqrt{u^2+v^2}}. \tag{2.9}$$

As a result of Eq. (2.9), the azimuth θ in the ICS can be computed by the pixel coordinates (u, v) as follows:

$$\theta = \sin^{-1}\left(\frac{v}{\sqrt{u^2+v^2}}\right) = \cos^{-1}\left(\frac{u}{\sqrt{u^2+v^2}}\right). \tag{2.10}$$

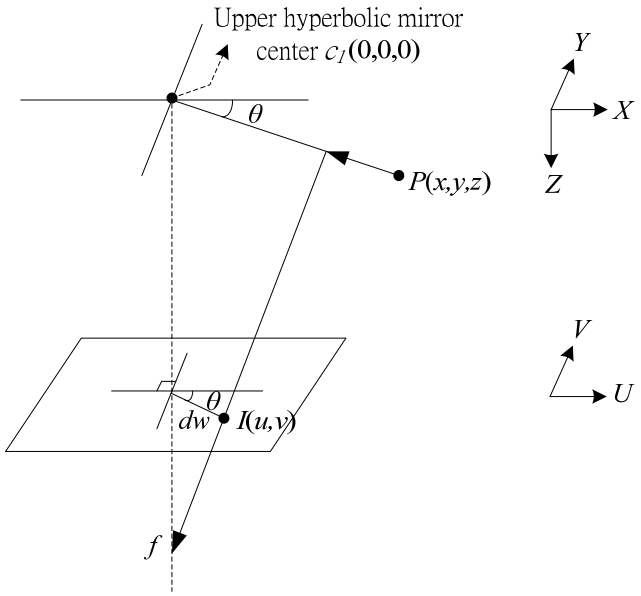


Figure 2.9 The system configuration of upper omni-camera with a hyperbolic mirror.

According to the characteristic that the axis of the camera is aligned with the axis of the hyperbolic mirror, the azimuth angle θ of point P in the WCS and the azimuth θ angle of point I in the ICS are the same one (according to the rotation-invariant property of the omni-camera). Therefore, the parameters x, y in the GCS can be estimated as follows:

$$\begin{aligned} x &= dw \times \cos\theta = \frac{b}{\tan\alpha_1 - \tan\alpha_2} \times \cos\theta \\ y &= dw \times \sin\theta = \frac{b}{\tan\alpha_1 - \tan\alpha_2} \times \sin\theta \end{aligned} \tag{2.11}$$

As a result, if an azimuth θ and a pair of elevation angles α_1 and α_2 are given, the unique position of a scene point P can be found. The method we use to transform each pixel in an omni-image to an azimuth angle and an elevation angle in the WCS will be described in Chapter 3. Therefore, if a pair of matching points (one is in an omni-image taken by the upper omni-camera, and the other is in an omni-image taken by the lower omni-camera) is known, a relevant 3D data is also obtained.

2.4 System Process

For the purpose of learning all the information that the proposed system can process a video surveillance with the two pairs of two-camera omni-directional imaging devices on the video surveillance car roof, we develop a learning interface for users. The entire learning process is shown in Figure 2.10.

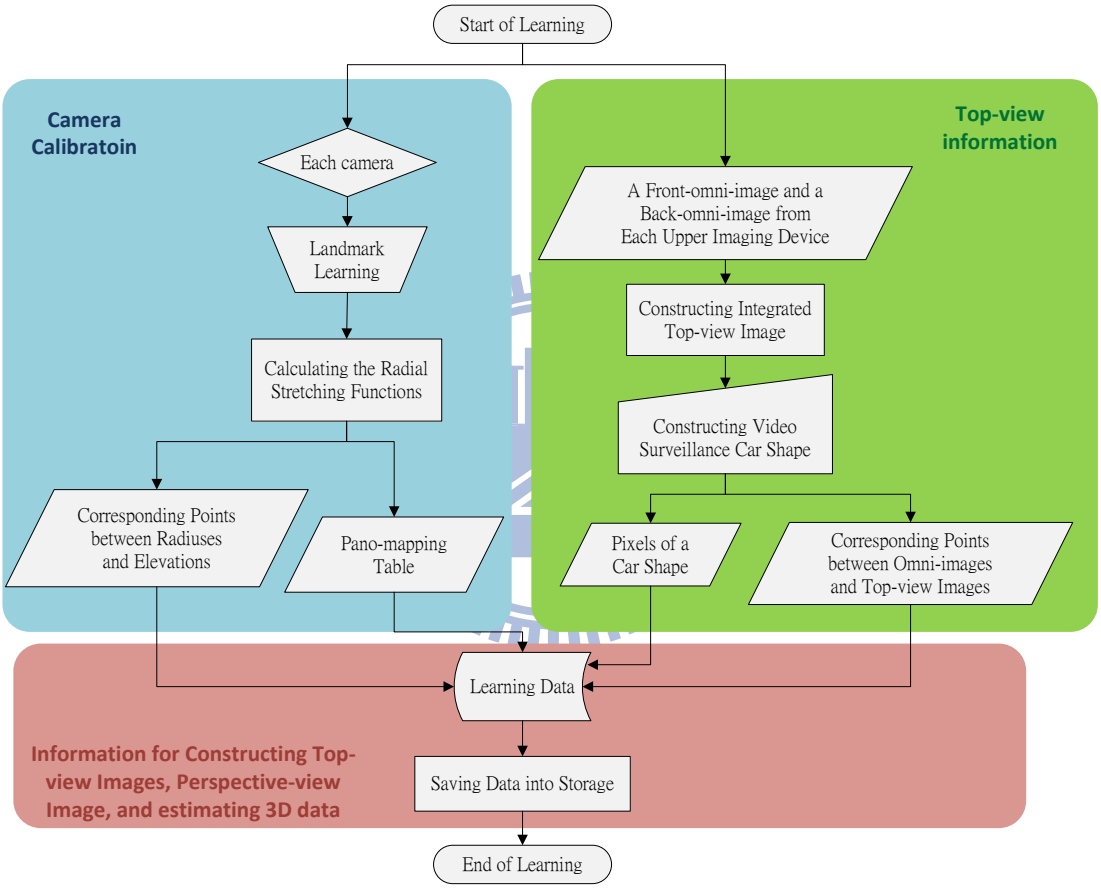


Figure 2.10 Flowchart of proposed learning process.

In this study, the recorded data are camera-related and object-related ones. The *camera-related data* are used in a transformation to estimate relevant 3D data and a transformation to construct top-view images. The former is obtained from the camera calibration processes which will be described in Chapter 3, and the latter is obtained from the transformation that transforming an omni-image to a top-view image which

will be described in Chapter 5. The *object-related data* is a shape of the video surveillance car in top-view image. It is used to construct a top-view image which is not affected by the height of the video surveillance car. The process will be described in detail in Chapter 5.

After all the data are obtained, they are saved into some text files. These files are then used in the video surveillance more than once, so this is also a method for improving the speed of calculations without computing the same data over and over. When the learning job has been done, the video surveillance system can start surrounding monitoring. The entire monitoring process of passers-by in suspect proposed in this study is shown in Figure 2.11.

As shown in Figure 2.11, we read the related table files at the beginning. Computer A is used to show the top-view image, and Computer B is used to show the perspective-view image. The communication of the two computers is described in Section 2.2.3. The construction of a perspective-view image will be described in Chapter 3. The passer-by detection, which yields a red mark in the image, will be described in Chapter 4. The construction of a top-view image and the integration of the two camera Systems will be described in Chapter 5. Finally, The passing-by car detection, which yields a yellow mark in the image, will be described in Chapter 6.

Because both the passer-by detection and passing-by car detection processes require heavy computations, the passing-by car detection process we propose is designed to be independent of the passers-by detection process. Such a compromise approach makes the execution of the two processes smoother.

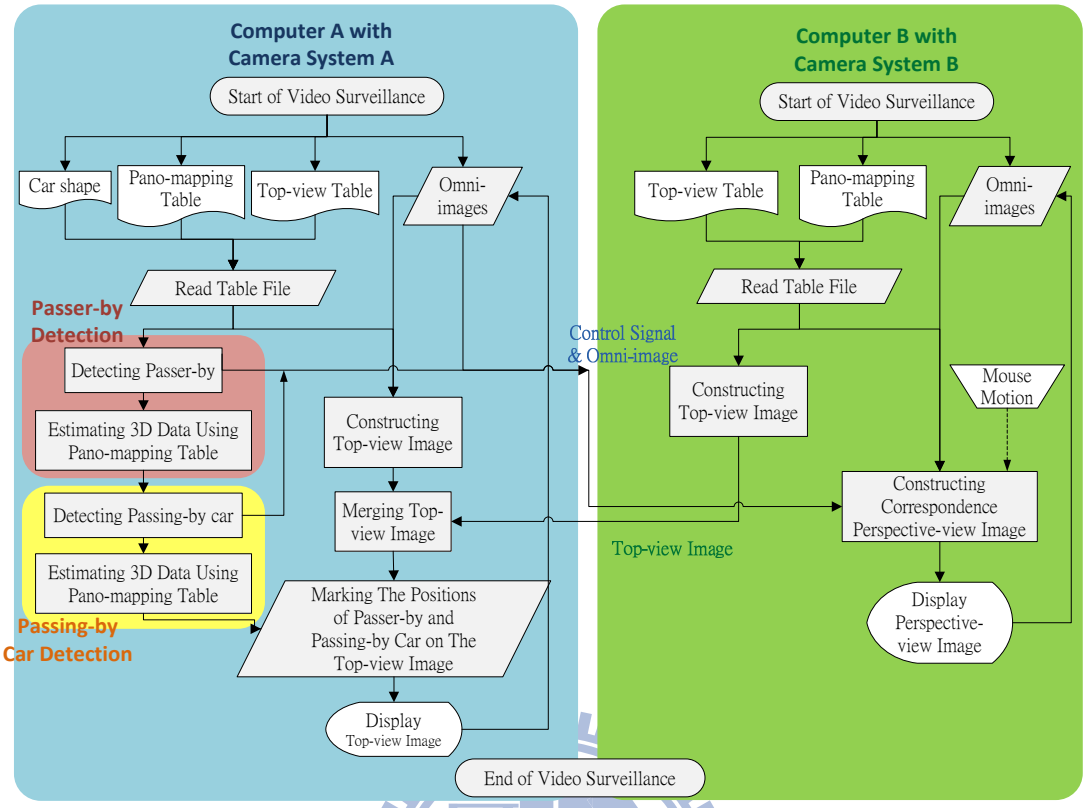


Figure 2.11 Flowchart of the proposed video surveillance system.

Chapter 3
Using Pano-mapping Tables for Unwarping Omni-images into Multi-perspective-view Images

3.1 Idea of Pano-mapping for Omni-image Unwarping

If a suspicious passing-by approaches the video surveillance car, the perspective-view image in the suspect's direction with respect to the car should be made available to the user in the car for a clearer inspection. This requires unwarping of the omni-images taken with the camera devices used in this study.

Conventional methods for unwarping omni-images require the knowledge of certain camera parameters, like the focal length of the lens, the coefficients of the mirror surface shape equation, etc., to calibrate the camera before omni-image unwarping. However, we cannot get the complete information of the omni-camera parameters in some situations. A solution to this problem is to use the space-mapping technique proposed by Jeng and Tsai [8], as mentioned previously. The technique is based on the use of a *pano-mapping table*, which may be regarded as a summary of the information conveyed by all the camera parameters. The pano-mapping table is created once forever for each omni-camera and not changed even when the camera is moved around. The table is created by a calibration process making use of certain selected points in the world space with known coordinates and their corresponding pixels in an omni-image. The detail will be described in Section 3.2. The table may be

used to create perspective-view images, as described in Section 3.3.

Another advantage of using the space-mapping technique is that the corresponding relationship of an omni-camera between a radial length r and an elevation angle ρ can also be obtained. The corresponding relationship is defined as a table, called *r-ρ Table* in this study. Like the method of 3D data extraction described in Section 2.3.3, if two corresponding pixels taken by a two-camera omni-directional imaging device are known, the corresponding elevation angles may derived by use of the r-ρ Table, and the azimuth θ also can be computed by the rotational-invariant property of the omni-camera. Then the unique position of a scene point can be found. This will be very useful in the following chapters for use in detections of passers-by and pass-by cars, for example.

The remainder of this chapter is organized as follows. In Section 3.2, we describe the technique we adopt for pano-mapping table creation in detail. In Section 3.3, we describe the method we use for imaging unwarping and perspective-view image generation based on the use of the pano-mapping table. We also propose a technique to change the view direction with mouse clicks for generation of a corresponding perspective-view image.

3.2 Construction of Pano-mapping Table

The method proposed by Jeng and Tsai [8] for pano-mapping table construction consists of three major stages: (1) landmark learning, (2) estimation of the coefficients of a radial stretching function describing the geometry of the mirror reflection in the omni-camera, and (3) pano-table creation.

3.2.1 Landmark learning

To construct a pano-mapping table, the first step is to pick out a number of pairs of world space points with known positions and their corresponding pixels in a taken omni-image. More specifically, in this study the coordinates of at least six points, called *landmark points* hereafter, are measured manually first with respect to a selected origin in the world space. Figure 3.1 shows the interface we have designed for acquiring the data of such landmark point pairs more easily. Especially, note that the origin defined in this study is a focal point of an omni-camera's hyperbolic mirror. The center O_m (in Figure 3.2) of the camera with known world coordinates (X_0, Y_0, Z_0) just appears to be the image center O_c (in Figure 3.1) with known image coordinates $p_k(u_0, v_0)$. After learning, assume generally that n sets of landmark point pair data are selected, each set including the image coordinates (u_k, v_k) with respect to the origin O_c of the image coordinate system (ICS) and the world coordinates $P_k(X_k, Y_k, Z_k)$ with respect to the origin O_m of the corresponding world coordinate system (GCS), respectively, where $k = 0, 1, \ldots, n-1$. Also, assume that the pixels of such landmark points in the taken omni-image are already segmented out after being selected on the user interface manually.

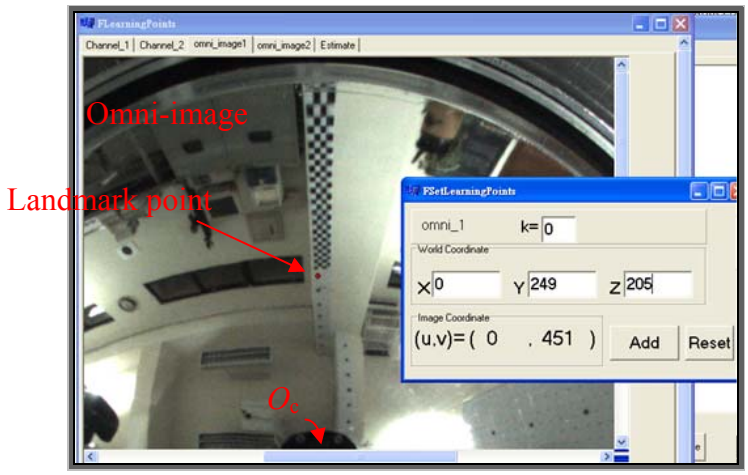


Figure 3.1 The interface for acquiring the data of the world space points

3.2.2 Estimation of coefficients of radial stretching function

Owing to the nonlinear property of the hyperbolic mirror surface shape, the radial-directional mapping should be specified by a nonlinear function f_r. As shown in Figure 3.2, we see that each of the elevation angles corresponds to a radial distance, or by notations and more specifically, that each elevation angle ρ of a scene point P corresponds to the radius r of its corresponding image point p.

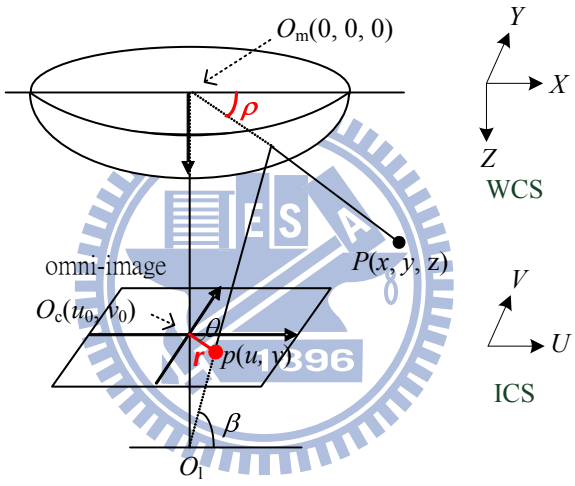


Figure 3.2 Nonlinear property of an omni-camera with mirror surface shape

Therefore, the radial distance r from each image pixel p with image coordinates (u, v) in the omni-image to the image center O_c at image coordinates (u_0, v_0) may be computed by $r = f_r(\rho)$. In this study, the function f_r, called a *radial stretching function*, is approximated by the following 5th-degree polynomial function:

$$r = f_r(\rho) = a_0 + a_1 \times \rho^1 + a_2 \times \rho^2 + a_3 \times \rho^3 + a_4 \times \rho^4 + a_5 \times \rho^5, \qquad (3.1)$$

where a_0 through a_5 are six coefficients to be estimated using the data of the landmark point pairs, as described in the following algorithm [8].

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

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

$$\rho_k = \tan^{-1}(\frac{Z_k}{D_k}); \quad r_k = \sqrt{u_k^2 + v_k^2}, \tag{3.2}$$

where D_k, is the distance from the landmark point P_k to the mirror center O_m in the X-Y plane of the WCS, computed by $D_k = \sqrt{X_k^{\,2} + Y_k^{\,2}}$.

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

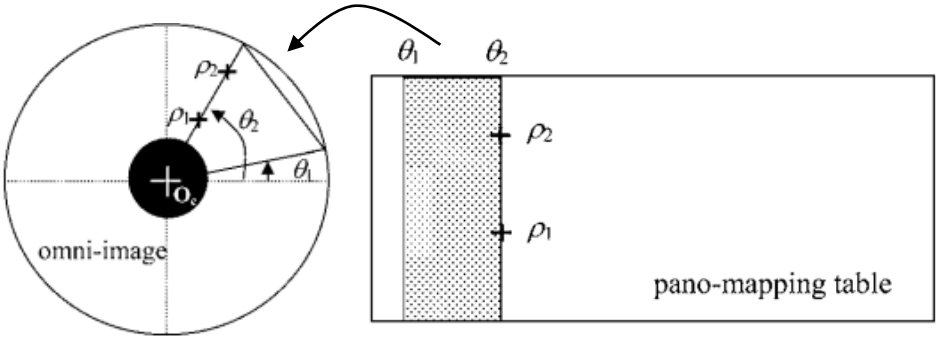
Substitute all the data $\rho_0, \rho_1, \ldots, \rho_{n-1}$ and $r_0, r_1, \ldots, r_{n-1}$ computed in Eq. (3.2) into Eq. (3.1) to get n homogeneous equations as follows:

$$\begin{aligned}
r_0 &= f_r(\rho_0) = a_0 + a_1 \times \rho_0^1 + a_2 \times \rho_0^2 + a_3 \times \rho_0^3 + a_4 \times \rho_0^4 + a_5 \times \rho_0^5, \\
r_1 &= f_r(\rho_1) = a_0 + a_1 \times \rho_1^1 + a_2 \times \rho_1^2 + a_3 \times \rho_1^3 + a_4 \times \rho_1^4 + a_5 \times \rho_1^5, \\
&\vdots \\
r_{n-1} &= f_r(\rho_{n-1}) = a_0 + a_1 \times \rho_{n-1}^1 + a_2 \times \rho_{n-1}^2 + a_3 \times \rho_{n-1}^3 + a_4 \times \rho_{n-1}^4 + a_5 \times \rho_{n-1}^5.
\end{aligned} \tag{3.3}$$

Finally, solving the above functions to get the desired coefficients (a_0, a_1, a_2, a_3, a_4, a_5) of the radial stretching function f_r by a numerical analysis method [16].

3.2.3 Filling of pano-mapping table entries

The procedure to build a pano-mapping table using the coordinate data of the landmark point pairs is described here. The table is a 2-dimensional one with the horizontal and vertical axes being the azimuth angle θ and the elevation angle ρ, respectively, as illustrated in Figure 3.3.



(a) (b)

Figure 3.3 Mapping between pano-mapping table and omni-image [8].

An example of the pano-mapping table of size $M \times N$ is shown in Table 3.1. Each entry E_{ij} with indices (i, j) in the pano-mapping table specifies an azimuth-elevation angle pair (θ_i, ρ_j), which represents an infinite set of points in the WCS passing through by the light ray with azimuth angle θ_i and elevation angle ρ_j for the reason that these world space points in S_{ij} are all projected onto an identical pixel P_{ij} in any omni-image taken by the camera, forming a pano-mapping f_{pm} from S_{ij} to P_{ij} as shown in Figure 3.3.

Table 3.1 Example of pano-mapping table of size $M \times N$ [8]

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

This mapping is shown in the Table 3.1 by filling entry E_{ij} with the coordinates (u_{ij}, v_{ij}) of pixel P_{ij} in the omni-image. We divide the range 2π (= 360°) of the azimuth

angles into M units, and the range of the elevation angles from ρ_s to ρ_e into N units, to create a table T_{pm} of $M \times N$ entries. Owing to the rotation-invariant property of the omni-camera, the azimuth angle θ of a space point P with respect to the x-axis in the GCS which the light ray passes is essentially identical to the angle ϕ of the corresponding pixel p with respect to the u-axis in the input image I. Hence, $\theta = \phi$. Each image point p in the ICS with respect to image center O_c may be computed by Eq. (3.1). Accordingly, the entries of table T_{pm} may be filled by the following algorithm [8].

Step 1. Divide the range 2π of the azimuth angles into M intervals, and the ith azimuth angle θ_i can be described by

$$\theta_i = i \times (2\pi / M), \text{ for } i = 0, 1, ..., M-1. \tag{3.4}$$

Step 2. Divide the range $(\rho_e - \rho_s)$ of the elevation angles into N intervals, and describe the jth elevation angle ρ_j by

$$\rho_j = j \times [(\rho_e - \rho_s)/N] + \rho_s, \text{ for } j = 0, 1, ..., N-1. \tag{3.5}$$

Step 3. Regard the pairs $(r, \phi) = (f_r(\rho), \theta)$ of all the image pixels to form a polar coordinate system with the image coordinates (u, v) specified by

$$u_{ij} = r_j \times \cos\phi_i = f_r(\rho) \times \cos\theta_i; \quad v_{ij} = r_j \times \sin\phi_i = f_r(\rho) \times \sin\theta_i. \tag{3.6}$$

Step 4. Based on Eqs. (3.1) and (3.6), fill the entry E_{ij} with corresponding image coordinates as follows:

$$u_{ij} = r_j \times \cos\theta_i; \quad v_{ij} = r_j \times \sin\theta_i, \tag{3.7}$$

where r_j is computed by

$$r_j = f_r(\rho_j) = a_0 + a_1 \times \rho_j^1 + a_2 \times \rho_j^2 + a_3 \times \rho_j^3 + a_4 \times \rho_j^4 + a_5 \times \rho_j^5, \qquad (3.8)$$

with the coefficients (a_0, a_1, a_2, a_3, a_4, a_5) computed in the way as described in Section 3.2.2.

3.3 Image Unwarping and Generation of Perspective-view Images

3.3.1 Generation of a perspective view

According to above-mentioned information, the procedure to construct a perspective-view image from an omni-image with the aid of a pano mapping table is described in the following.

Input: an omni-image G, a pano-mapping table T_{pm} (with $M \times N$ entries), and a planar rectangular region ($W \times H$) at a distance D with respect to the mirror center O_m.

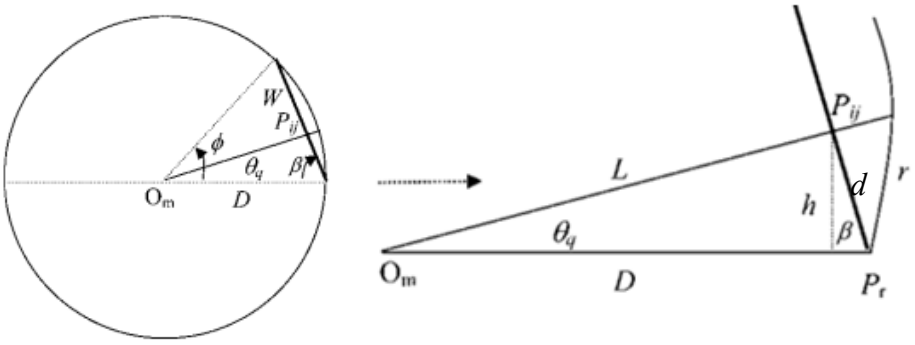
Output: a perspective-view image Q of any size $M_Q \times N_Q$.

Idea:

(A) Map each image pixel q_{kl} in Q at coordinates (k, l) to an entry E_{ij} in T_{pm} filled with coordinates pixel (u_{ij}, v_{ij}).

(B) Assign the color value of the pixel p_{ij} of G at (u_{ij}, v_{ij}) to q_{kl}.

A top view of the configuration for such an image generation process is shown in Figure. 3.4, and the above idea will be accomplished by the following two steps [8], computing the azimuth angles θ_q and elevation angle ρ_q associated with E_{ij} and corresponding to q_{kl}.



(a) (b)

Figure 3.4 A Top-view configuration for generating a perspective-view image [8].

Steps:

Step 1. Computing θ_q associated with E_{ij} and corresponding to q_{kl} ---

Referring to Figure 3.4, the angle ϕ can be derived by trigonometry to be as follows:

$$W^2 = D^2 + D^2 - 2 \times D \times D \times \cos\phi;$$
$$\phi = \cos^{-1}[1 - \frac{W^2}{2 \times D^2}]. \tag{3.9}$$

Also, β in the figure may be derived to be:

$$\beta = \frac{\pi - \phi}{2}. \tag{3.10}$$

Next, we compute the index i of entry E_{ij} of table T_{pm} corresponding to pixel q_{kl} in image Q. First, let P_{ij} denote the intersection point of the light ray R_q projected onto q_{kl} and the planar projection region A_p. Note that each entry E_{ij} has a corresponding P_{ij}. Then, we compute the distance d between point P_{ij} and the border point P_r shown in Figure 3.4 by linear proportionality as

$$d = k \times \frac{W}{M_Q}, \tag{3.11}$$

where the projection region A_p has a width of W, the image Q has a width of M_Q pixels, and pixel q_{kl} has an index of k in the horizontal direction.

In Figure 3.4(b), we also can compute the distance L between point P_{ij} and the mirror center O_m by trigonometry as follows:

$$L = \sqrt{D^2 + d^2 - 2 \times d \times D \times \cos\beta}, \tag{3.12}$$

and then the distance h from point P_{ij} to the line segment $\overline{O_m P_r}$ connecting O_m and p_r as:

$$h = d \times \sin\beta. \tag{3.13}$$

Therefore, the azimuth θ_q of point P_{ij} with respect to $\overline{O_m P_r}$ satisfies

$$\sin\theta_q = \frac{h}{L} = \frac{d \times \sin\beta}{\sqrt{D^2 + d^2 - 2 \times d \times D \times \cos\beta}}, \tag{3.14}$$

which leads to

$$\theta_q = \sin^{-1}\frac{h}{L} = \sin^{-1}[\frac{d \times \sin\beta}{\sqrt{D^2 + d^2 - 2 \times d \times D \times \cos\beta}}]. \tag{3.15}$$

Finally, the index i of entry E_{ij} may computed by linear proportionality as

$$i = \frac{\theta_q}{2\pi} \times M\,. \tag{3.16}$$

Step 2. Computing ρ_q associated with E_{ij} and correspond to q_{kl} ---

An illustration of the involved imaging configuration from a lateral view is shown in Figure 3.5. The height of the projection region A_p is H and the image Q is divided into N_Q intervals, and by linear proportionality again, the

height of P_{ij} can be computed, similarly to the derivation of Eq. (3.11), to be:

$$H_q = l \times \frac{H}{N_Q}. \tag{3.17}$$

Then, by trigonometry, the elevation angle ρ_q is derived as:

$$\rho_q = \tan^{-1}\left(\frac{H_q}{L}\right). \tag{3.18}$$

Therefore, we can compute the index j of E_{ij} by proportionality again, similarly to the derivation of Eq. (3.16), to be:

$$j = \frac{(\rho_q - \rho_s) \times N}{(\rho_e - \rho_s)}. \tag{3.19}$$

With the indices (i, j) of E_{ij} ready, finally, we can obtain the co-ordinates (u_{ij}, v_{ij}) in E_{ij} and assign the color value of the image pixel p_{ij} of G at co-ordinates (u_{ij}, v_{ij}) to pixel q_{kl} of Q at co-ordinates (k, l). After all pixels of Q are processed, the final content of Q is the desired perspective-view image.

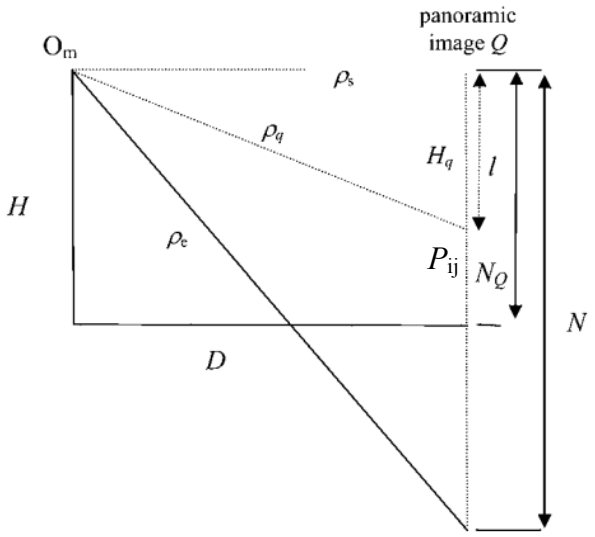


Figure 3.5 A Lateral-view configuration for generating a perspective-view image [8].

3.3.2 Generation of specified perspective-view images with mouse clicks

In this section, a method proposed to make the user interface friendly for a user to change the view direction of the perspective-view image with moving a mouse is described. Figure 3.6 shows an experimental result of a perspective-view image and a corresponding omni-image. According to the previously-mentioned derivation and observation of the following perspective-view image and omni-image, we can get some relations with a mouse motion and the direction of a perspective view. Specifically, the horizontal motion (M_x) of a mouse is just dependent on the azimuth angle θ, and the vertical motion (M_y) is dependent on the elevation angle ρ.

As a result, we reconstruct Eq. (3.15) as follows:

$$\theta_q = \sin^{-1}\frac{h}{L} + \theta_{mouse}, \tag{3.20}$$

where θ_{mouse} is an index to record the azimuth angle of the view. It will increase as the mouse moves to the left. On the other hand, it will decrease as the mouse moves to the right. Eq. (3.18) is also reconstructed as follows:

$$\rho_q = \tan^{-1}\left(\frac{H_q}{L}\right) + \rho_{mouse}, \tag{3.21}$$

where ρ_{mouse} is an index to record the elevation angle of the view. Similar to the θ_{mouse}, it will increase as the mouse moves down. On the other hand, it will decrease as the mouse moves up. Hence, a user can use a mouse to choose any view direction conveniently to observe the scene which he/she is concerned with. The user interface of the program becomes friendly after adding the two variables θ_{mouse} and ρ_{mouse}.

(a)

(b)

Figure 3.6 Corresponding omni-image and perspective-view image. (a) A perspective-view image. (b) Omni-image from which (a) was generated.

Chapter 4
Automatic Detection of a Suspicious Passer-by with a Two-camera Omni-directional Imaging Device

4.1 Introduction

To detect a suspicious passing-by person approaching the video surveillance car, an automatic human detection process should be available. In this chapter, the method proposed to detect a suspicious passer-by near a surveillance car and estimate his/her relevant 3D data using a two-camera omni-directional imaging device is described. Using the result of this method, a user in a video surveillance car can get the passer-by's position and height for use in various security applications. Also, according to the information of a passer-by's position, a corresponding perspective view also can be computed by the proposed method.

The remainder of this chapter is organized as follows. In Section 4.2, we introduce the concepts behind the proposed system. In Section 4.3, we describe the techniques we propose for detection of a passer-by's distance and height using two omni-images which are taken by a two-camera omni-directional imaging device.

4.2 Review of Related Concepts in Proposed System

In this study, we use a pair of two-camera omni-directional imaging devices to perform a video surveillance task around a video surveillance car. To segment the

shape of a moving object out of an omni-image accurately, we use the moment-preserving threshold method proposed by Tsai [17] and a so-called *dynamic offsetting* scheme for image intensity adjustment in the proposed passer-by and passing-by car detection methods. The concepts behind the two methods are described in following two sections.

4.2.1 Moment-preserving thresholding for object segmentation

An approach to automatic threshold selection for segmenting meaningful objects out of a given image is adopted in this study and is reviewed here [17]. The approach can automatically and deterministically select multiple thresholds without iterations or searches. A bi-level thresholding solution is obtained. The details are described in the following.

Given an image f with n pixels whose gray value at pixel (x, y) is denoted by $f(x, y)$, the ith moment m_i of f is defined as

$$m_i = \frac{1}{n}\sum_x \sum_y f^i(x, y), \quad i = 1, 2, 3. \tag{4.1}$$

Moments can also be computed from the histogram of f in the following way:

$$m_i' = \frac{1}{n}\sum_{j=0}^{1} n_j (z_j)^i = \sum_{j=0}^{1} p_j (z_j)^i, \quad i = 1, 2, 3 \tag{4.2}$$

where n_j is the total number of pixels in f with gray value z_j and $p_j = n_j/n$. The approach also defines m_0 to be 1. Image f can be considered as a blurred version of an ideal bi-level image which consists of pixels with only two gray values z_0 and z_1, where $z_0 < z_1$. The adopted moment-preserving thresholding scheme is to select a threshold value such that if all below-threshold gray values in f are replaced by z_0 and

all above-threshold gray values replaced by z_1, then the first three moments of image f are preserved in the resulting bi-level image g.

To find the desired threshold value t, we can solve two equations described by Eqs. (4.1) and (4.2) above to obtain p_0 and p_1, as described in the following [17]:

$$c_d = \begin{vmatrix} m_0 & m_1 \\ m_1 & m_2 \end{vmatrix}; \quad c_0 = \frac{1}{c_d}\begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix}; \quad c_1 = \frac{1}{c_d}\begin{vmatrix} m_0 & -m_2 \\ m_1 & -m_3 \end{vmatrix};$$

$$z_0 = \frac{1}{2}\left[-c_1 - (c_1^2 - 4c_0)^{\frac{1}{2}}\right]; \quad z_1 = \frac{1}{2}\left[-c_1 + (c_1^2 - 4c_0)^{\frac{1}{2}}\right]; \tag{4.3}$$

$$p_d = \begin{vmatrix} 1 & 1 \\ z_0 & z_1 \end{vmatrix}; \quad p_0 = \frac{1}{p_d}\begin{vmatrix} 1 & 1 \\ m_1 & z_1 \end{vmatrix}; \quad p_1 = 1 - p_0,$$

and then choose t as the p_0-tile of the histogram of f, i.e., choose t such that

$$p_0 = \left(\frac{1}{n}\right)\sum_{z_j \le t} n_j. \tag{4.4}$$

Because a high-resolution omni-camera is used in this study, it is a heavy computation load to acquire the histogram of an omni-image. Note that p_0 and p_1 denote the fractions of the below-threshold pixels and the above-threshold pixels in f, respectively; $p_0 + p_1 = 1$; and z_0 and z_1 can be regarded as the *representative gray values* for the below-threshold and the above-threshold pixels, respectively. We use an alternative method to obtain the threshold value t in this study. An illustration of a histogram with parameters z_0, z_1, t is shown in Figure 4.1.

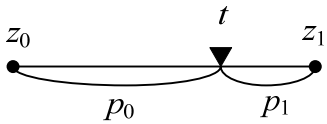


Figure 4.1 A conceptual illustration of a histogram with parameters z_0, z_1, and t.

Accordingly, we may use the following equation to approximate the value t to

speed up the computation:

$$t = z_0 + (z_1 - z_0) \times p_0. \tag{4.3}$$

Our experimental results also support the validity of using the above equation to compute the value of t.

4.2.2 Dynamic offsetting

Dynamic offsetting is an approach used in this study to solve the problem of object detection in images caused by varying light intensities in different weather conditions. In order to make the intensity of two omni-images similar by dynamic offsetting, we use the following algorithm where a pixel of an image C at image coordinates (u, v) is denoted as $p_C(u, v)$.

Algorithm 4.1 Intensity normalization by dynamic offsetting.

Input: two grayscale omni-images A and B both with size n (the number of pixels).

Output: a modified version A' of grayscale omni-image A, whose intensity is normalized based on B.

Steps:

Step 1. Calculate the mean values M_A and M_B of A and B, respectively, where

$$M_A = \frac{1}{n} \sum_v \sum_u p_A(u, v) \tag{4.4}$$

and M_B is similarly defined.

Step 2. Subtract M_A from M_B to get an offset denoted as M_{offset} as follows:

$$M_{offset} = M_A - M_B. \tag{4.5}$$

Step 3. Subtract all the pixels in A, $p_A(u, v)$, by the value M_{offset} to obtained a modified version A' of A as follows:

$$p_{A'}(u,v) = p_A(u,v) - M_{offset}. \quad (4.6)$$

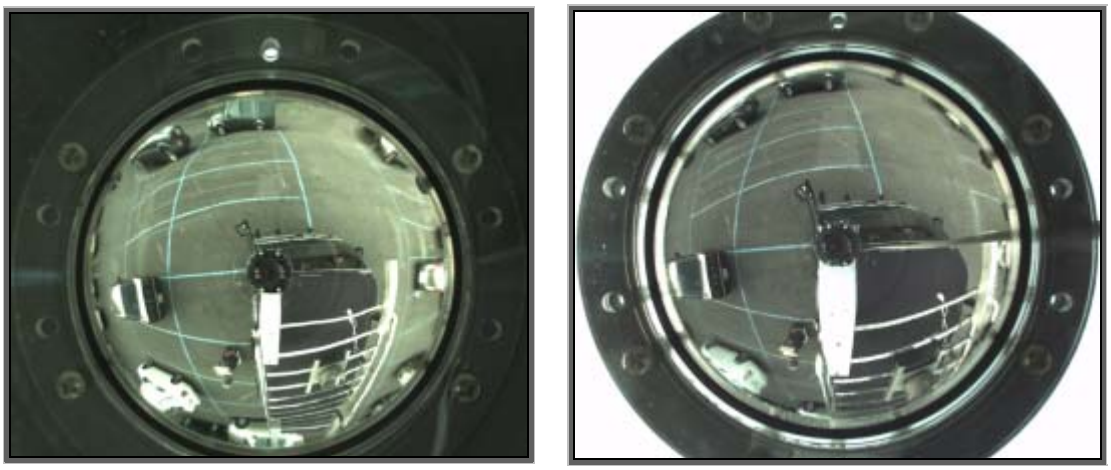
After the completion of the above steps, the intensity of the two omni-images A' and B will not be much different.

4.3 Estimation of a Passer-by's Distance and Height Information

The proposed method for extraction of a passer-by's information consists of three major stages: (1) moving object extraction (2) acquisition of the passer-by's head; and (3) estimation of the passer-by's 3D data. In the first stage, we used the dynamic threshold technique proposed by Tsai [17] and dynamic offsetting, as described in the last section, to acquire moving objects from an omni-image. The detail will be described in Section 4.3.1. In the second stage, we propose an algorithm to detect a passer-by's head in the omni-image using a specific property in an omni-camera, as described in Section 4.3.2. In the final stage, we use the technique described in Section 4.3.2 to estimate the relevant 3D data of the passer-by. The detail will be described in Section 4.3.3.

4.3.1 Detection of moving objects in an omni-image

Before extracting moving objects around a video surveillance car, omni-images without unnecessary objects should be captured, and each of such images is called a *background image*. An example of background images is shown in Figure 4.2. To detect moving objects, each omni-camera must capture the current image. Each of the current images is called a *foreground image*.



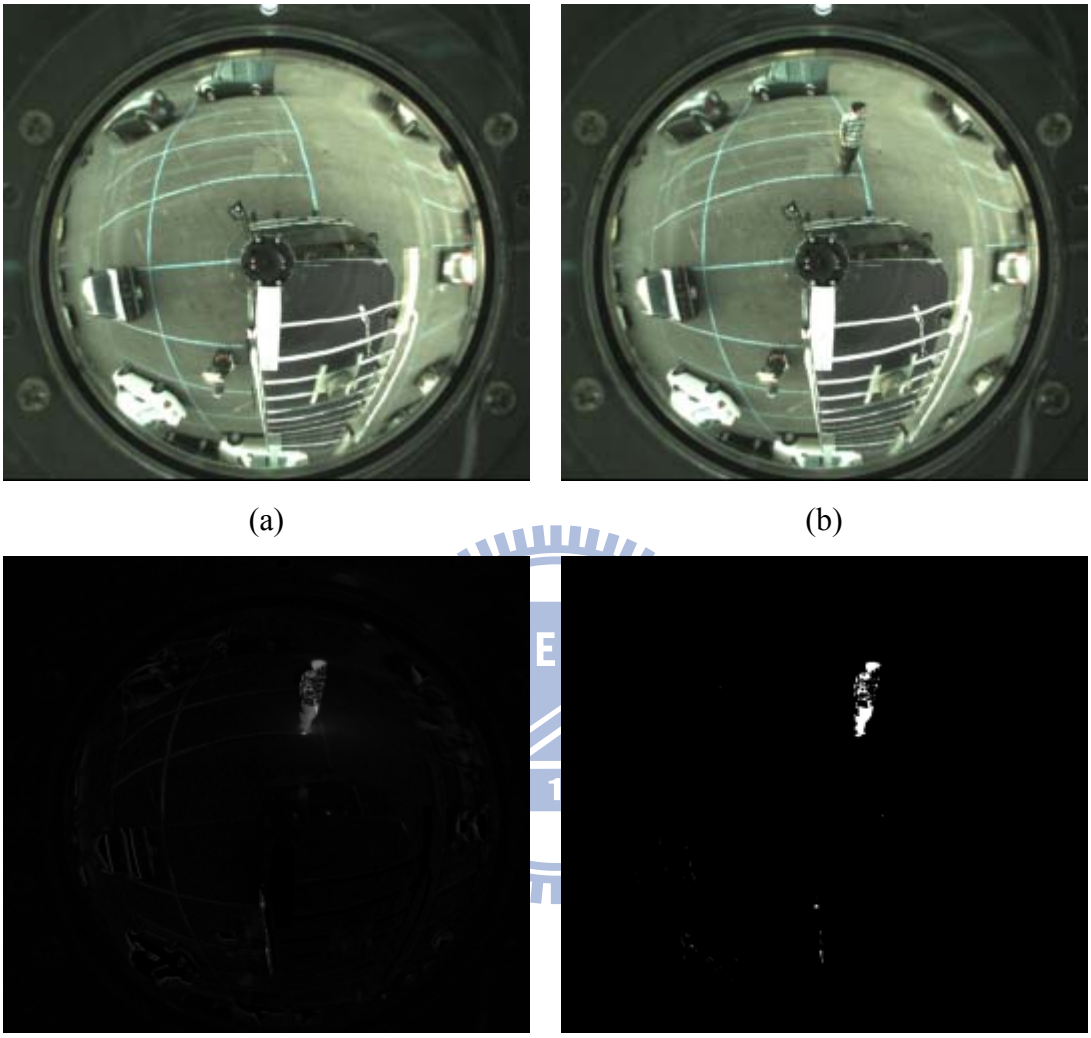
(a) (b)

Figure 4.2 Background images of a two-camera omni-directional imaging device. (a) A *background* taken by an upper omni-camera. (b) A *background* taken by a lower omni-camera.

First, we transform the *background* and *foreground* images, which are color ones, into two grayscale omni-images. However, to avoid the varying light intensities that affect the accuracy of the object detection, we used the *dynamic offsetting* technique described Section 4.2.2 to make the intensity of the two omni-images similar. Second, by subtracting the background image from the foreground one, we obtain all differences between the two images. Third, because there is a lot of noise in the surveillance area, such as those caused by light variations, we set an appropriate threshold parameter *TH* to threshold the difference image to eliminate noise. The value *TH* we use in this study is a dynamic threshold value yielded by using moment-preserving thresholding proposed by Tsai [17] and described in the last section (Section 4.2.1). Finally, if the difference value of a pixel is larger than the parameter *TH*, it is recorded as "1"; else, as "0". This process is the so-called *bi-level thresholding*. At the end, we will obtain a binary image I_{BI} with detected moving objects labeled as "1.".

A sequence of images yielded as intermediate results of the above-mentioned

object detection process is shown in Figure 4.3. Figure 4.3(d) is the resulting binary image I_{BI}. It is also shown that Eq. (4.3) we proposed is feasible to get the threshold t.



(a) (b)

(c) (d)

Figure 4.3 Related images of passers-by detection. (a) Background image. (b) Foreground image. (c) The difference image obtained after a subtracting process. (d) The binary image obtained by moment-preserving thresholding [17].

4.3.2 Detection of a passer-by's head by component labeling

Before introducing how to detect a passer-by's head, we prove a specific

property of an omni-camera, as shown in Figure 4.4. The property is: each line L, which is horizontal to the Z-axis in the WCS, will be projected onto the image as a line I_L which goes through the omni-image center (O_c).

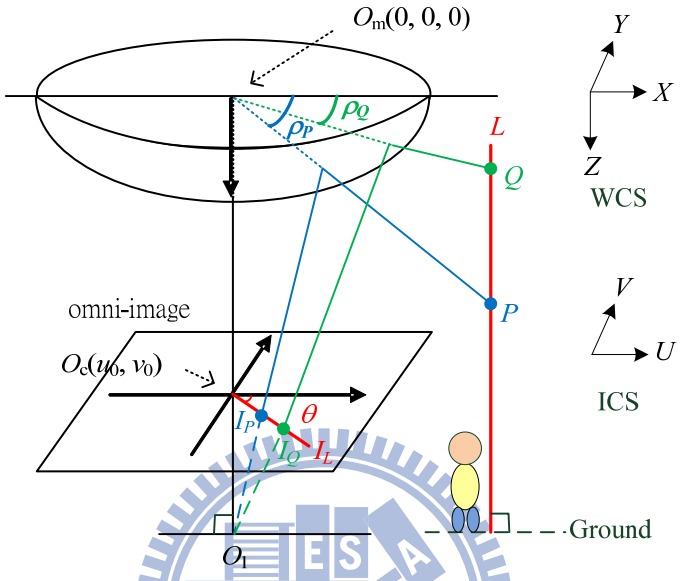


Figure 4.4 A specific property of an omni-camera.

An explanation is in the following. Owning to the *rotation-invariant* property of the omni-image, the azimuth angle which is formed by the light ray with respect to the X-axis in the WCS is essentially identical to the azimuth angle θ of the corresponding pixel with respect to the U-axis in the omni-image. Because the Z-axis is vertical to the ground, all the points (e.g., P and Q) on L have the same azimuth angle with respect to the X-axis in the WCS. Therefore, these points are projected onto the points (e.g., I_P and I_Q) with an identical azimuth angle θ with respect to the U-axis in the ICS, forming a line I_L on the omni-images. And this line extends to go through the center of the omni-image, O_c.

A more theoretical verification of the above fact is in the following. By using minimum distance estimation with ρ_P and ρ_Q in the corresponding r-ρ Table

described in Section 4.3.3, the corresponding radial lengths r_P and r_Q can be derived, respectively. Because the azimuth angles θ are the same, the corresponding pixels in the omni-images can be described as follows:

$$\begin{aligned} I_P &= (r_P \cos\theta,\ r_P \sin\theta); \\ I_Q &= (r_Q \cos\theta,\ r_Q \sin\theta). \end{aligned} \tag{4.7}$$

The line including the two point I_P and I_Q in the ICS is expressed as follows:

$$v = \frac{(r_Q - r_P)\sin\theta}{(r_Q - r_P)\cos\theta} u + b = \tan\theta \times u + b. \tag{4.8}$$

Because the line passes through the point I_P, substituting the coordinates of I_P into Eq. (4.8) leads to the following equation:

$$r_P \sin\theta = \frac{\sin\theta}{\cos\theta} r_P \cos\theta + b = r_P \sin\theta + b. \tag{4.9}$$

Thus, $b = 0$ can be obtained, and the line of (4.8) can be expressed in the following:

$$v = \tan\theta \times u. \tag{4.10}$$

As a result, it is easy to see that the line goes through the center of the omni-image, O_c, with coordinates $(u, v) = (0, 0)$.

However, every passer-by around of the video surveillance car stands on the ground, so it is also means that everyone is vertical to the ground. As illustrated in Figure 4.5, according to the above-proved property, the *midline* I_L of a passer-by (i.e., the *axis* of his/her body) goes through the center of the omni-image O_c As a result, the process of finding the top of a passer-by's head is simplified, which is described as Algorithm 4.2. And the result of passer-by's head detection in Figure 4.3 is shown in Figure 4.6.

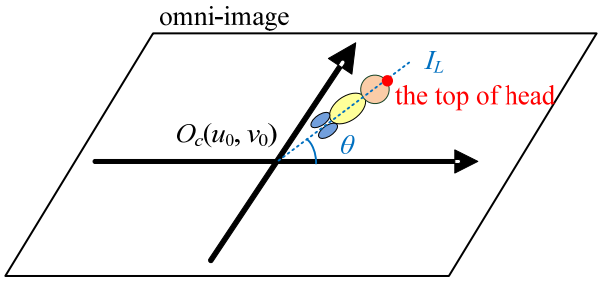


Figure 4.5 The midline of a passer-by through the center of the omni-image.

Algorithm 4.2 Finding the top of a passer-by's head.

Input: A binary image I_{BI} which is obtained as described in Section 4.3.1.

Output: The position of the passer-by's head.

***Steps*:**

Step 1. Initialize the position of the passer-by's head to be 0 (the minimum distance).

Step 2. Scan each line in the radial direction through the image center based on the polar coordinates (θ, r), where θ is the azimuth angle and r the radial length. If the whole omni-image has scanned, output the position of the passer-by's head and finish the process.

Step 3. If there is a segment with fifteen continuous points or more, called a *continuous segment*, in the line, continue; else, go to *Step 2.*

Step 4. Find the farthest point with respect to the image center of each *continuous segments* in the line.

Step 5. If the farthest point is farther than the position of the passer-by's head, assign it as the new position of the original record of the head position and Go to *Step 2.*

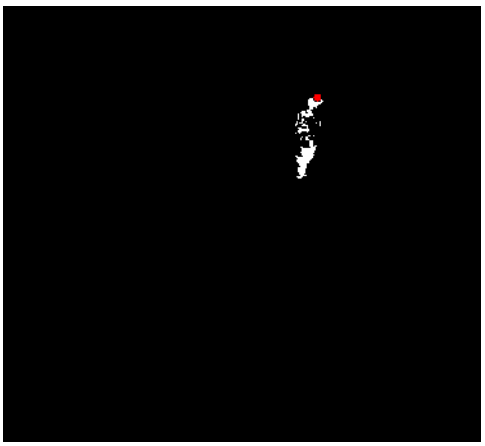


Figure 4.6 A result of passer-by's head detection. (The top of the passer-by's head is marked in red.)

4.3.3 Calculation of a passer-by's distance and height in 3D space

At the beginning, note that we construct an r-ρ Table using the *radial stretching function* in Section 3.2 in the learning process. An r-ρ-Table records the corresponding relationships of an omni-camera between radial lengths r and elevation angles ρ, as shown in Table 4.1.

Table 4.1 r-ρ-Table

$\boldsymbol{r}$	r_1	r_2	r_3	…	r_n
$\boldsymbol{\rho}$	ρ_1	ρ_2	ρ_3	…	ρ_n

To calculate a passer-by's relevant 3D data, we do the same process as described in Section 4.3.2 using each omni-camera of a two-camera omni-directional imaging device. Then, we get the two positions of a passer-by's head (P_{up}, P_{down}) in the two omni-images taken by a two-camera omni-directional imaging device. When

the two positions $P_{up}(x_{up}, y_{up})$ and $P_{down}(x_{down}, y_{down})$ are obtained, we can derive the radial distance in the following:

$$r_{up} = \sqrt{x_{up}^2 + y_{up}^2}; \quad r_{down} = \sqrt{x_{down}^2 + y_{down}^2}. \tag{4.11}$$

By using minimum distance estimation with r in the corresponding r-ρ Table, the corresponding elevation angles ρ_{up}, ρ_{down} can be derived.

Because the two omni-cameras are combined coaxially in the longitudinal direction and their camera coordinates (X, Y, Z) have the same axis directions, the two azimuth angles (θ_{up}, θ_{down}) of P_{up} and P_{down} are the same by the rotational-invariant property of the omni-camera. However, because of mechanical errors, it is accepted that the difference angle between θ_{up} and θ_{down} is less than 10 degrees. Thus, if θ_{up} and θ_{down} is matched within this tolerance of angular error, we can use the 3D data acquisition proposed in Chapter 2.3.3 to estimate a passer-by's position (X, Y, Z) with respect to the center of the upper omni-camera C_{up} in the WCS. In more detail, with $H_{C_{up}}$ denoting as the height of C_{up}, the passer-by's height $H_{passer\text{-}by}$ can be computed as follows:

$$H_{passer\text{-}by} = H_{C_{up}} - Z, \tag{4.12}$$

where $H_{C_{up}}$ in this study is 256cm as measured manually in our experiment..

To summarize, the flowchart in Figure 4.7 shows the overall processes used to detect a passers-by in this study.

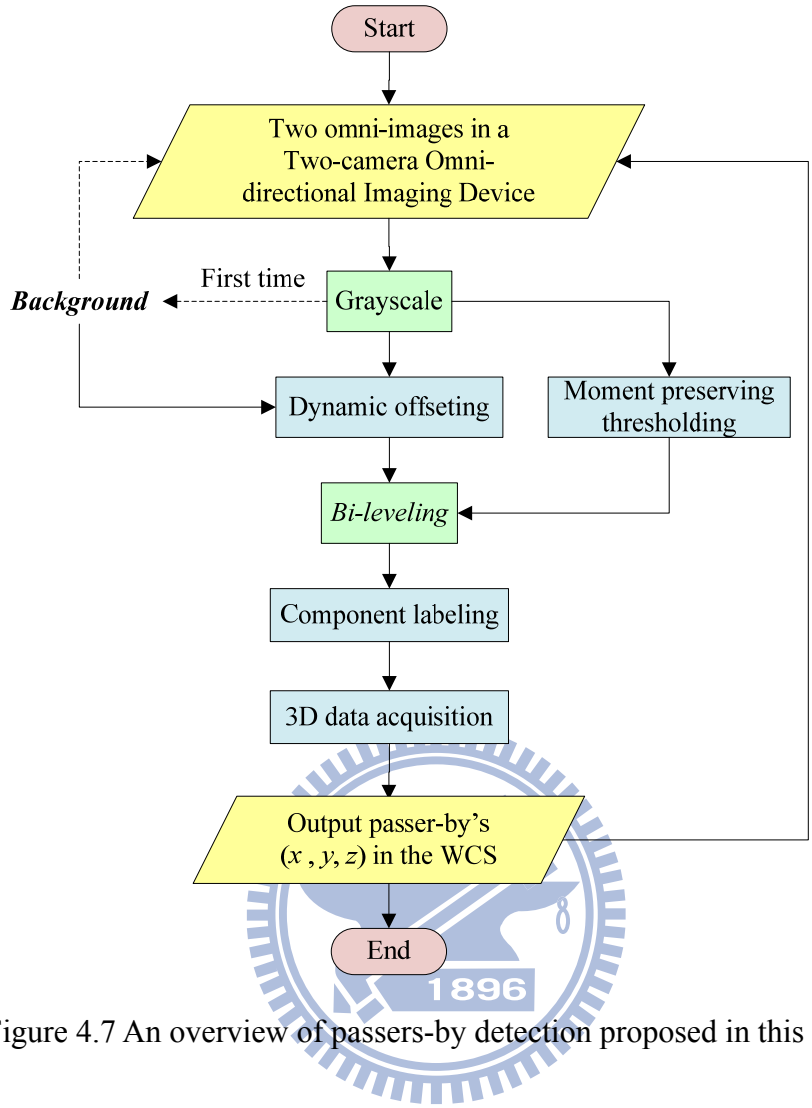


Figure 4.7 An overview of passers-by detection proposed in this study.

Chapter 5
Integration of Two Omni-images into a Top-view Image with a Pair of Two-camera Omni-directional Imaging Devices

5.1 Introduction

In order to expand the range of surveillance, two pairs of two-camera omni-directional imaging devices are used in this study. A top-view image around a video surveillance car which comes from merging the two omni-images taken by the upper cameras in the two pairs of two-camera omni-directional imaging devices is available for users in the car. In this chapter, we describe all the relevant methods which are used in computing such an integrated top-view image.

The remainder of this chapter is organized as follows. In Section 5.2, we introduce the techniques we propose for construction of such an integrated top-view image. In Section 5.3, we describe the techniques we propose for superimposition of the video surveillance car shape and filling of ground texture in the top-view image.

5.2 Construction of a Top-view Image

5.2.1 Construction of a top-view image with an omni-camera

Because the view range of the upper camera in a two-camera omni-directional imaging device is wider than the lower one, we use the upper omni-camera to construct a top-view image. The height of the omni-camera affixed on the video surveillance car roof is known in this study, so we can derive a simple top-view image by assuming all pixels in the omni-image are on the ground.

Figure 5.1 illustrates the geometry relationship between a point on the ground and the upper omni-camera. A straightforward derivation (forward mapping) is in the following. Note that, forward mapping means to map the pixels in an omni-image to those of the top-view image. Specifically, any pixel p at coordinates (u, v) in the ICS can be used to compute the radial distance r as follows:

$$r = \sqrt{u^2 + v^2}, \tag{5.1}$$

and the azimuth angle θ as follows:

$$\theta = \cos^{-1}\frac{u}{r} = \sin^{-1}\frac{u}{r}. \tag{5.2}$$

By using minimum-distance estimation with r in the corresponding r-ρ Table as described in Section 4.3.3, the corresponding elevation angle ρ can be derived. More specifically, we try to find the r_i in the r-ρ Table with the minimum distance to r, and regard the corresponding ρ_i of r_i in the table as the corresponding elevation angle ρ of r. Then, according to the known height of the mirror center dh and the known elevation angle ρ so computed, as shown in Figure 5.1, the horizontal distance between a scene point and the mirror base center, dw, can be computed as follows:

$$dw = dh \times \cot\rho. \tag{5.3}$$

Accordingly, by the rotational-invariant property of the omni-camera, the position (x, y) of a point P on the ground in the WCS can be obtained from Eqs. (5.2) and (5.3) as

follows:

$$x = dw\cos\theta;\ y = dw\sin\theta. \tag{5.4}$$

By mapping all pixels in the omni-image in this way, a top-view image can be obtained.

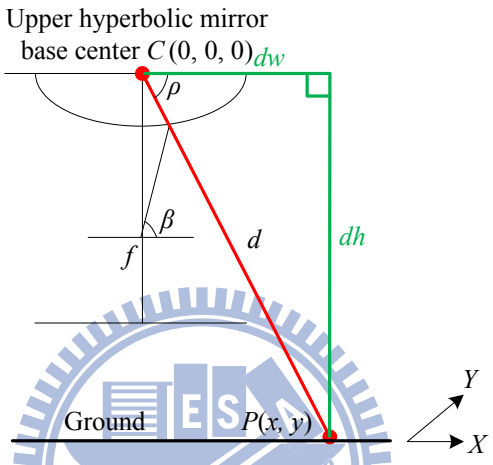


Figure 5.1 The ray tracing of a scene point P on the ground with a hyperbolic-shaped mirror.

However, a forward mapping will lead to a "broken" image with many unfilled points, as shown in Figure 5.2(b) (the black portion in the image). Thus, we attempt to derive a backward mapping for computing a "complete" top-view image. Note that, backward mapping is to map all pixels in a top-view image to those of an omni-image. First, we may compute dw for every pixel with coordinates (x, y) in the WCS as follows:

$$dw = \sqrt{x^2 + y^2}, \tag{5.5}$$

and the azimuth angle θ can be derived as follows:

$$\theta = \cos^{-1}\frac{x}{dw} = \sin^{-1}\frac{y}{dw}. \tag{5.6}$$

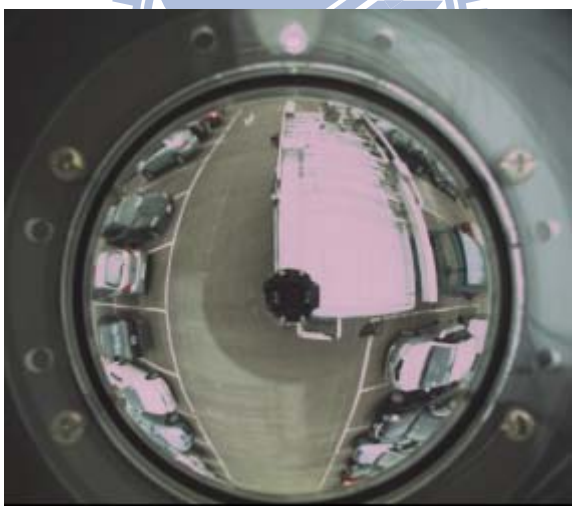
Eq. (5.3) can be rewritten to derive the elevation angle as follows:

$$\rho = \tan^{-1}\frac{dh}{dw}. \tag{5.7}$$

By minimum distance estimation with ρ in the corresponding r-ρ Table as described in Section 4.3.3, the corresponding radial distance r can be derived. Accordingly, by the rotational-invariant property of the omni-camera, the image coordinates (u, v) of the corresponding pixel p in the ICS can be obtained from Eq. (5.6) as follows:

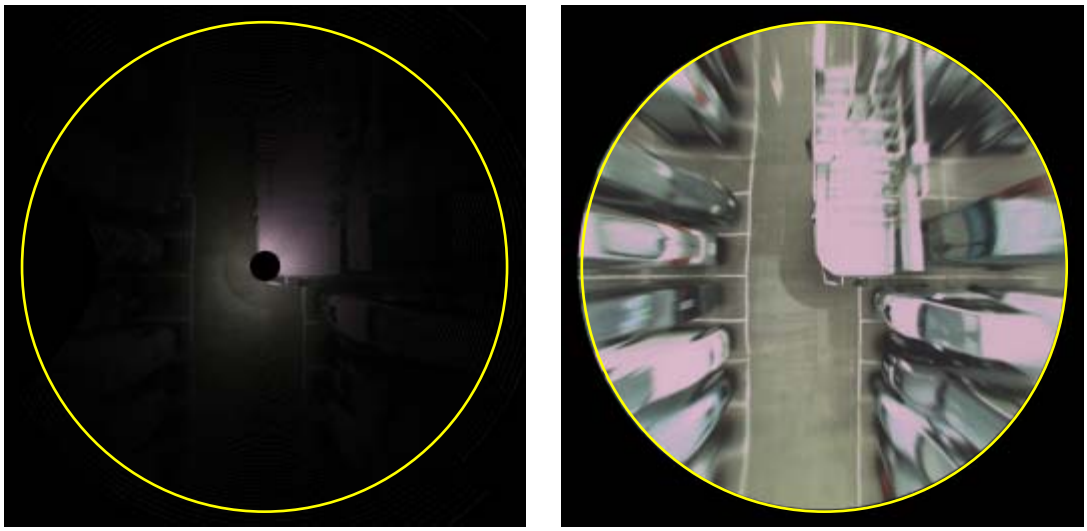
$$u = r\cos\theta;\ v = r\sin\theta. \tag{5.8}$$

As a result, a complete top-view image can be obtained, as shown in Figure 5.2(c).



(a)

Figure 5.2 An omni-image and its corresponding top-view images. (a) An omni-image. (b) A top-view image obtained from forward mapping. (c) A top-view image obtained from backward mapping.



(b) (c)

Figure 5.3 An omni-image and its corresponding top-view images. (a) An omni-image. (b) A top-view image obtained from forward mapping. (c) A top-view image obtained from backward mapping continued.

In order to improve the program speed, the corresponding relations of all pixels between an omni-image and a top-view image is stored into a table, called *TopviewTable*, in the learning process. Accordingly, constructing top-view images do not need to compute the above-mentioned equations again and again, but just to refer to the corresponding *TopviewTable*.

5.2.2 Calculation of relative position of two omni-cameras

Figure 5.3 is an illustration of the layout of the video surveillance car roof where the length and the width of the car roof are obtained by manual measurement. Accordingly, we can get the relative position between the pairs of two-camera omni-directional imaging devices easily with this layout we designed. The red circles in the layout are the positions of our devices. If the front device is assumed as the origin of the WCS, then the position of the back device is (–110, –330). That is, the

offset between the two devices is (−110, −330).

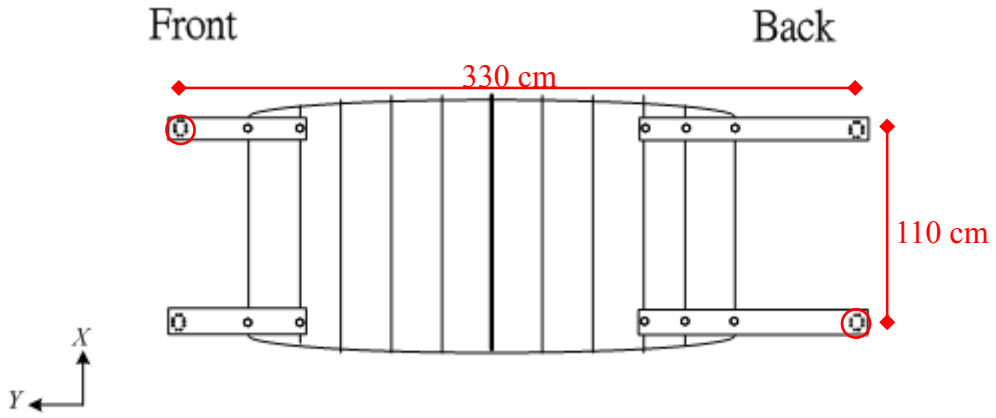


Figure 5.4 An illustration of the layout of the video surveillance car roof.

5.2.3 Merging of two top-view images into a single one

At the beginning, we construct two top-view images using the omni-images taken from the upper cameras in the pairs of two-camera omni-directional imaging devices, respectively. To make the merging construction simple and fast, we divide a top-view image around the video surveillance car into two parts. As shown in Figure 5.4, C_W is the width of the relative distance between the two upper omni-cameras, and C_L is the length of the relative distance between the two upper omni-cameras. One part is the front half top-view of the car surrounding (covering the image's upper part before the spot at $C_L / 2$) and the other is the rear half top-view of the car surrounding (covering the image's lower part behind at spot at $C_L / 2$). The front one is taken from the front upper omni-camera and the rear one is taken from the rear upper omni-camera. As described in Section 5.2.2, the relative position of the two upper

cameras is described by an offset (C_W, C_L) = (−110, −330) between the front omni-camera and the rear one, so all the pixels in the rear top-view image need to shift for such an offset. Thus, a preliminary integrated top-view image can be obtained as shown in Figure 5.5

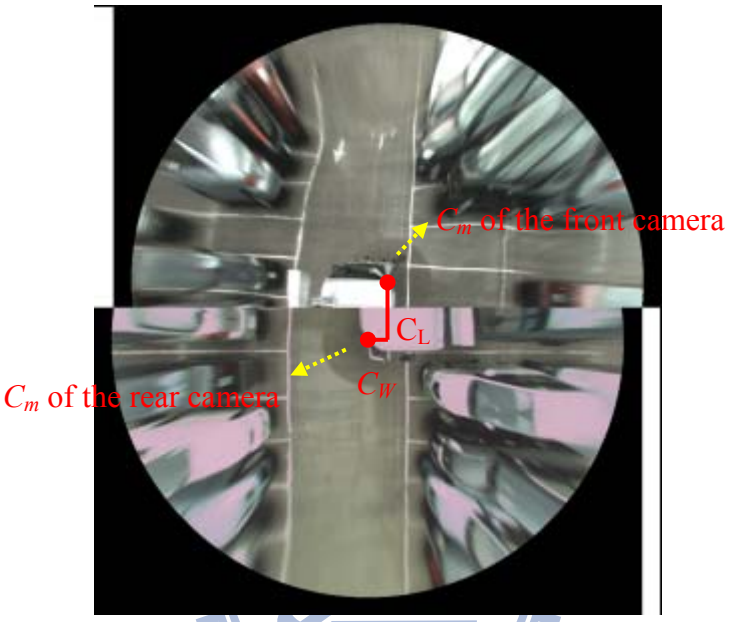


Figure 5.5 An integrated top-view image.

To obtain further a good-looking top-view image, an eclipse shape is used in the top-view image as the *viewing window*. The parameters a and b of the eclipse shape are described in the following:

$$a = r - C_W;\ b = r - C_L / 2, \tag{5.9}$$

where r is the radius of each top-view image taken from an omni-camera, C_W is 110 cm, and C_L is 330 cm, as mentioned previously. We discard the pixels with image coordinates (x, y) which satisfy the following equation in the integrated top-view image:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}>1 \tag{5.10}$$

Thus, an integrated top-view image can be obtained as shown in Figure 5.6.

Figure 5.6 An integrated top-view image with a eclipse shape.

5.3 Video Surveillance Car Shape Superimposition and Ground Texture Filling in Top-view Image

In the previous section, we can derive a simple top-view image by assuming all pixels in the omni-image are on the ground. However, not all the things are *flat* on the ground obviously, e.g., the video surveillance car, as shown in Figure 5.5. In this section, some techniques will be proposed to solve this problem of drawing the video surveillance car in the integrated omni-image.

5.3.1 Construction of car shape

Because the two pairs of two-camera omni-directional imaging devices are affixed on the video surveillance car roof, the car shape in the top-view image is always fixed. By this property, we can construct a car shape with respect to a top-view image in the learning process. In this learning stage, the video surveillance car shape is selected manually. As shown in Figure 5.6, the yellow mark portions are the video surveillance car shape appearing originally in the non-integrated omni-images taken by the two upper omni-cameras.

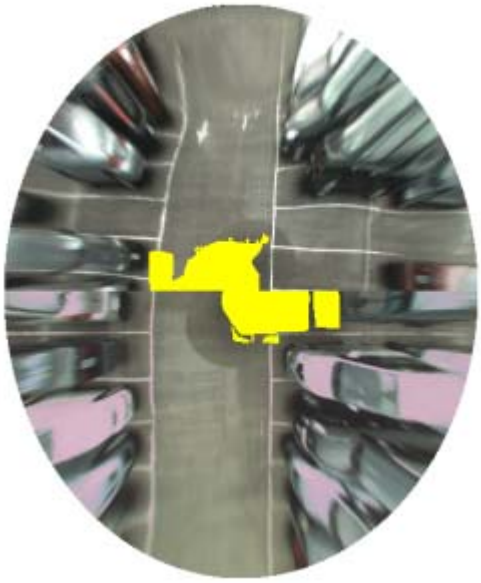
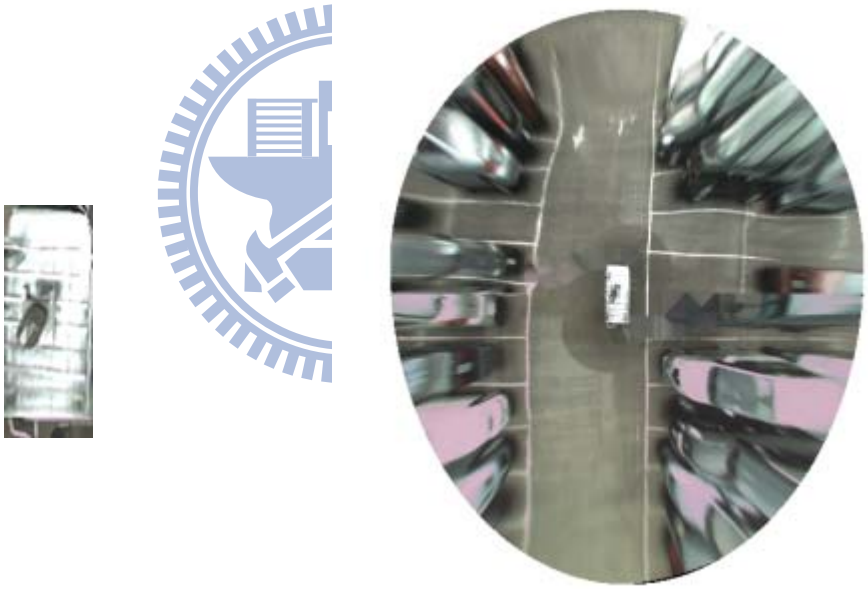


Figure 5.7 An illustration of the video surveillance car shape.

5.3.2 Video surveillance car shape superimposition and ground texture filling

A simple texture synthesis scheme is adopted in this study to fill ground texture into the shape area of the video surveillance car, which is described in this section.

Specifically, to fill the yellow mark portions (the car shape) in Figure 5.6, for each car shape pixel p_c, we find the closest non-car shape pixel p_n to p_c, and use the color of p_n to fill p_c. This simulates ground text within the car shape area. Furthermore, the corresponding relations of all pixels within the car shape with those outside the car shape in an integrated top-view image is stored into a table called *CarPatternMatch* in the learning process to improve the program speed. After filling the ground texture, a real top-view image portion of a car as shown in Figure 5.7(a) is superimposed on the corresponding position in the integrated top-view image, with a result as shown in Figure 5.7(b).



(a) (b)

Figure 5.8 A top view of a car and an integrated top-view image. (a) A top-view image of video surveillance car. (b) An integrated top-view image with video surveillance car shape superimposition and ground texture filling.

Chapter 6
Automatic Detection of a Passing-by Car with a Two-camera Omni-directional Imaging Device

6.1 Proposed Idea of Automatic Detection of a Passing-by Car

In Chapter 5, we derive a simple top-view image by assuming all pixels in the omni-image are on the ground. However, if a passing-by car is in the surveillance area, the constructed top-view image will be incorrect, as shown in Figure 6.1. In this study, we propose a method to detect a passing-by car using a two-camera omni-directional imaging device.

Figure 6.1 A top-view image with a passing-by car.

A flowchart of the proposed method is shown in Figure 6.2. It consists of three

major stages: (1) detection of a passing-by car in an omni-image, (2) detection of a passing-by car in the real world, and (3) car shape superimposition and ground filling. In the first stage, the ground is eliminated first, and then moment preserving thresholding is used to select the non-ground area. Finally, region growing, erosion, and dilation are used to capture the passing-by car. In the second stage, the information of the median point obtained from the process of region growing is used to construct a car model. After the car model is used to conduct template matching, 3D data acquisition is applied to obtain the position of the passing-by car. In the final stage, because the car shape in the omni-image and the position in the real world are known, we can eliminate the car shape in the omni-image, use ground filling, and superimpose a top-view shape of a car on to the top-view image. As a result, a correct and understandable top-view image can be obtained. These techniques are proposed to detect the passing-by car and solve the problems caused by the top-view image error in this study, and each method will be described in the following sections.

The remainder of this chapter is organized as follows. In Section 6.2, we introduce the detection of a car region in an omni-image. In Section 6.3, we describe the detection of a car position in the real world. Finally, the proposed techniques for passing-by car shape superimposition and ground texture filling in the top-view image are described in Section 6.4.

6.2 Detection of Car Region in an Omni-image

6.2.1 Detection of non-ground region

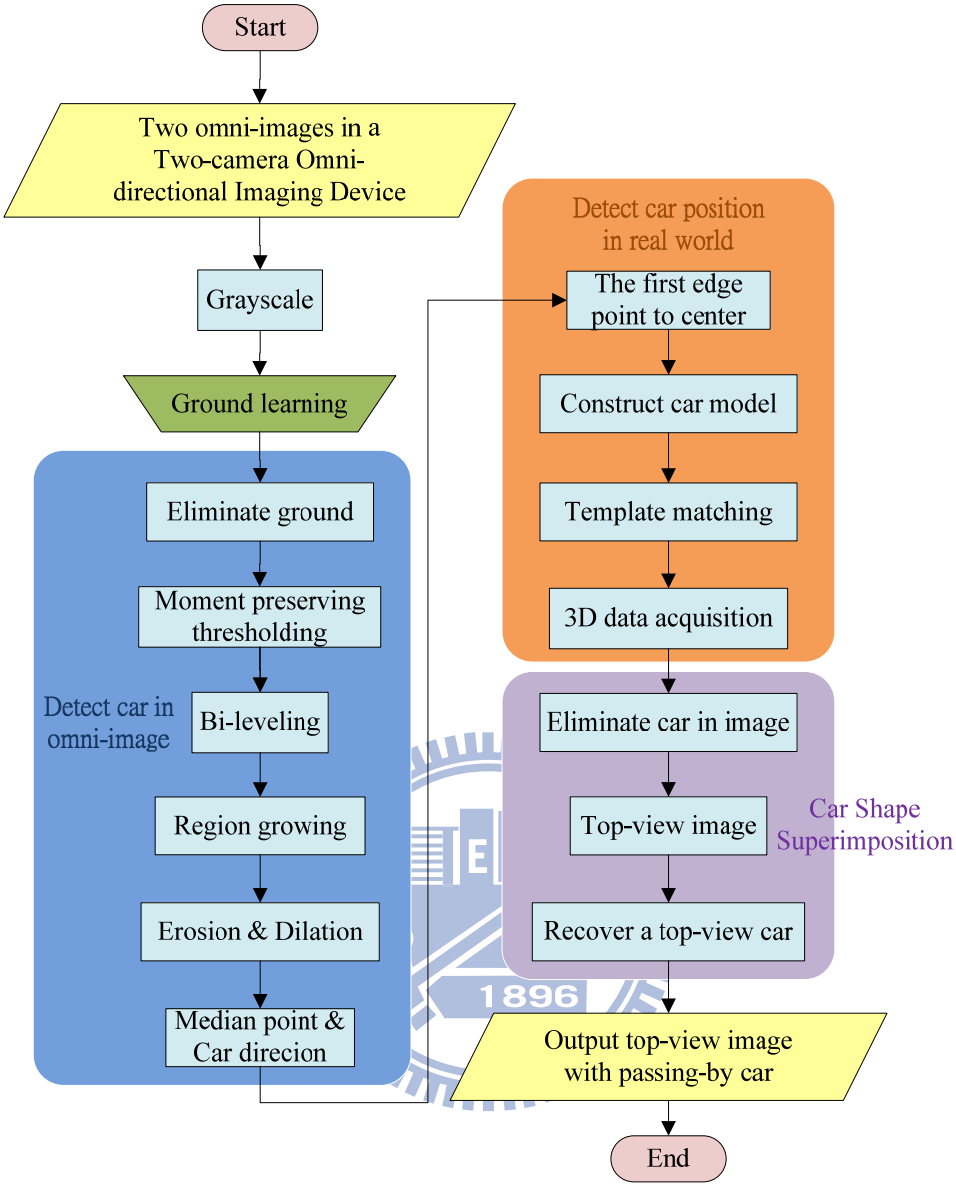


Figure 6.2 Flowchart of a passing-by car detection

At the beginning of non-ground region detection, we transform the two omni-images taken from a two-camera omni-directional imaging device into two grayscale omni-images, and each of the two images is processed in the following way, respectively. In order to discard the ground in the omni-image I, the ground needs to be learned. More specifically, a ground area is selected manually first. Let the number of total pixels in the area be n, and the gray value at pixel (u, v) in I be denoted by $I(u, v)$. The mean value of the ground can be computed as follows:

$$mean = \frac{1}{n}\sum_{v}\sum_{u} I(u,v). \tag{6.1}$$

Figure 6.3 shows the interface we have designed for acquiring the data of such ground points more easily.

(a) (b)

Figure 6.3 The interface for ground learning. (a) The omni-image taken by the upper omni-camera. (b) The omni-image taken by the lower one.

To eliminate the ground area in I, the gray values of all the pixels of I are reduced by the mean value of the ground to get a difference image I_d. Then, the image I_d is used to compute a threshold TH by moment-preserving thresholding introduced in Section 4.2.1[17]. Finally, if the difference value of a pixel is larger than the parameter TH, it is recorded as "1"; else, as "0". This process is the so-called *bi-level thresholding*. At the end, we obtain a binary image I_{BI} with non-ground object pixels all labeled as "1," as shown in Figure 6.4.

6.2.2 Detection of car region by region growing and component labeling

(a) (b)

Figure 6.4 The non-ground omni-images. (a) The omni-image taken by the upper omni-camera. (b) The omni-image taken by the lower one.

A region growing method [18] is used in this study to cluster the related regions. First, a seed is selected as the start point, and the eight neighboring points of this start point are examined to check whether they belong to the region or not. The proposed scheme for this *connected-component check* will be described later. Then, each of the points decided to belong to the region is used as a seed again, and the connected-component check is repeated, until no more region points can be found. A flowchart describing the concept is shown in Figure 6.5. More details of the method are described as an algorithm in the following.

Algorithm 6.1: *Finding the connected region in an omni-image*.

Input: A seed point and a binary image I_{BI}, as described in Section 6.2.1.

Output: An image with the connected region of the seed.

Steps:

1 Select a seed P in the image as the start point.

2 Check the eight neighboring points T_i of P to see if they belong to the region or not.

2.1 Find all the neighboring points N_i of T_i which belong to the similar region

for each T_i.

2.2 Calculate the value of the similarity degree between T_i and each N_i.

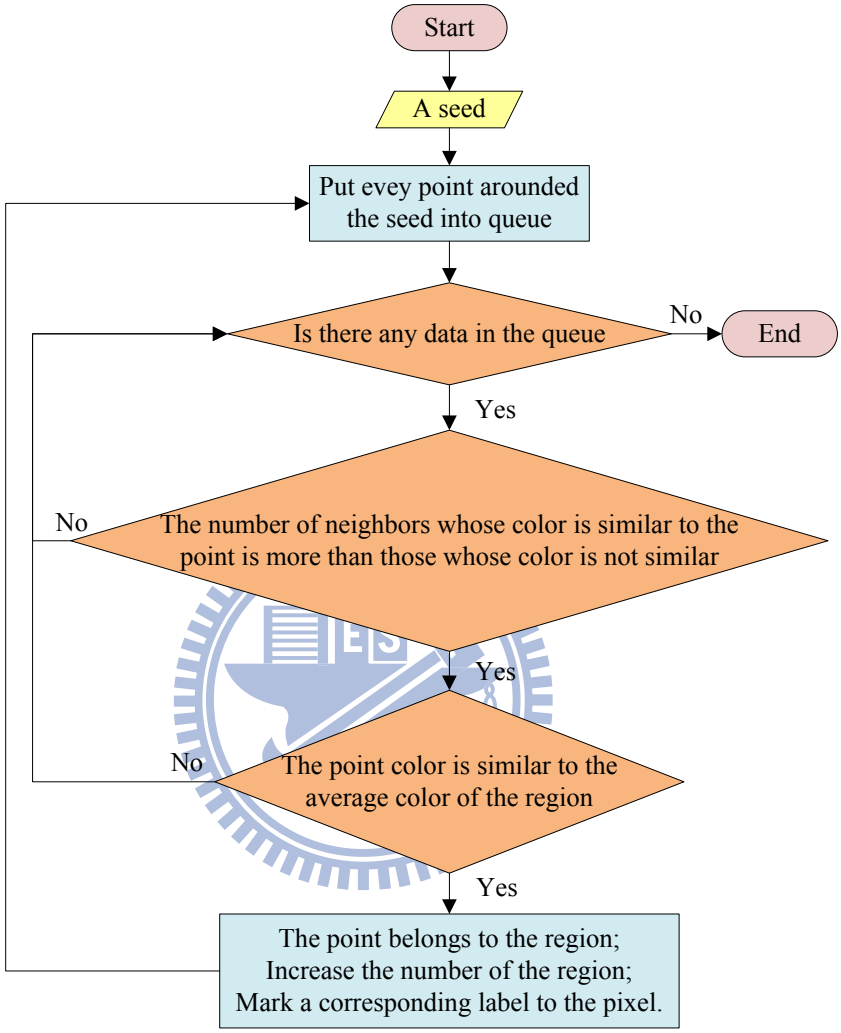


Figure 6.5 A flowchart of the region growing we used.

2.3 Decide whether T_i belongs to the similar region according to the similarity values by the following steps.

2.3.1 Compare the values of similarity calculated in Step 2.2 with a threshold TH_1 separately (the detail will be described later).

2.3.2 Calculate the number p of similarity values which are larger than TH_1.

2.3.3 Calculate the number q of similarity values which are smaller than or

equal to TH_1.

2.3.4 Compare p with q, and if the value of p is larger than q, then mark the point T_i as not belonging to the region and go to Step 2 to process the next T_i; else, continue.

2.3.5 Calculate the similarity degree d between T_i and the average RGB values of all pixels in the region of the ground (the detail will be described later).

2.3.6 Compare the similarity degree d with another threshold TH_2, and if d is smaller than TH_2, then mark T_i as belonging to the ground region; else, mark T_i as not.

2.4 Gather the points B_i which belong to T_i and belong to the region of the ground.

3 If there are some points of B_i which are not examined yet, then regard each B_i as a seed P and go to Step 2 again to check if they belong to the ground region or not.

In Steps 2.1 and 2.2, when a point T_i is examined, all the neighboring points N_i of T_i which have already been decided to belong to the ground region are found out first, and a similarity degree between T_i and each of its eight neighboring points, as shown in Figure 6.6, is computed. The similarity degree between two points A and B is computed in the following way:

$$\text{similarity between } A \text{ and } B = \left| r_A - r_B \right| + \left| g_A - g_B \right| + \left| b_A - b_B \right| \quad (6.1)$$

where r_C, g_C, and b_C are the color values of point C with $C = A$ or B here.

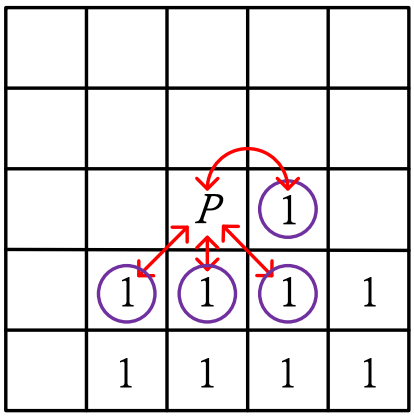


Figure 6.6 Illustration of calculation of the similarity degree between two image points.

In Step 2.3.1, after the similarity degree is calculated, the degree is compared with a threshold TH_1, whose value may be adjusted by a user. If the value is large, then the scope of the ground region which is found will be enlarged; else, reduced.

In Steps 2.3.2 and 2.3.3, the two introduced values p and q are set to zero at first. The value of p represents the number of points whose similarity degree is larger than TH_1, and the value of q represents the number of points whose similarity degree is not so. Hence, if a degree is larger than TH_1, then we add one to p, and if the degree is not so, then we add one to q. Afterward, in Step 2.3.4, if the value of p is larger than q, the point T_i is marked as not belonging to the region, and then go to Step 2 again to check the next T_i. If the value of p is not so, then an additional iterative process is conducted to examine T_i.

Sometimes the boundary between the connective region and obstacles is not very clear in images. So in Step 2.3.5, an average values AVR is calculated first, which contains 3 values, namely, the average values R_{avr}, G_{avr} and B_{avr} of the red, green, and blue values, respectively, of all the pixels in the ground region. We use AVR to decide whether a pixel T_i belongs to the ground region or not. The similarity degree d between T_i and AVR, as mentioned in the step, is then calculated according to a similar version of Equation 6.1. In Step 2.3.6, d is compared with another threshold TH_2. If d

is smaller than TH_2, then the point T_i is marked as belonging to the region; else, as not.

In Step 2.4, the points B_i which belong to T_i and belong to the region of the ground are found, and in Step 3, these points are regarded as seeds and Step 2 is repeated again to check if these points belong to the region or not. No matter whether the point T_i belongs to the region or not, the point will be marked as *scanned*.

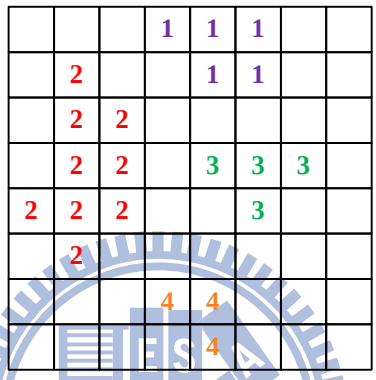


Figure 6.7 The concept of the component labeling we used.

A component labeling technique is also used here. Every pixel which has not been *scanned* in the non-ground image is regarded as a seed and processed in Algorithm 6.1 until all pixels in the non-ground image part are marked as *scanned*. Every first seed we put into the algorithm has a unique label, and all pixels in the non-ground part which is regarded as the same connected region in the algorithm are marked with the same label. The concept is shown in Figure 6.7. In the meantime, each number of those clusters can be counted. Thus, the labels of which the number is the largest can be known, and the region which is marked with this label is regarded as a passing-by car.

Furthermore, we apply consecutively two operations of morphology to decrease the interference of noise. One is *erosion* and the other is *dilation*. An erosion

operation can eliminate the noise in the shape, and a dilation operation can mend unconnected areas to make the shape smooth. The equations of them are described in the following where A and B are sets in Z^2 (2D image area) and all elements of them are zero (false) or one (true); $\hat{B}$ is the complement of the set B and its origin is z:

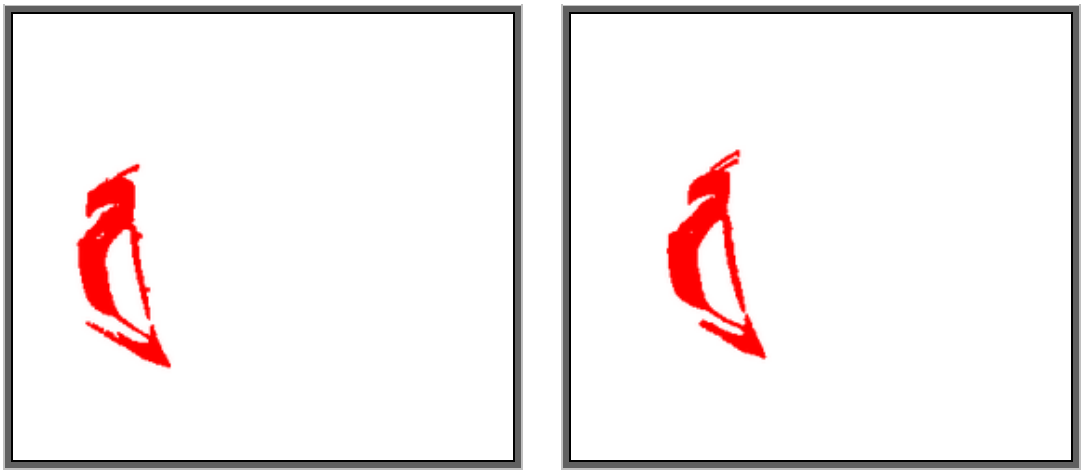
$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}; \tag{6.2}$$

$$A \Theta B = \{z \mid (B)_z \subseteq A\}. \tag{6.3}$$

We denote the *dilation* of A by B by $A \oplus B$ for which if there is at least one element overlap between $\hat{B}$ and A, then z is set true. The result of dilation is using B as a mask translated by z over the set A. The *erosion* of A by B is denoted $A \Theta B$ for which if B is contained in A, then z is set true. The result of erosion is also using B as a mask translated by z over the set A.

After we obtain a binary image of passing-by car shape, the noise which is much smaller or bigger than the car can be eliminated by the method of *erosion* using a certain square shape as the mask. For example, given a binary image A composed of the square D of size 1 pixel on the side and a mask B also being a square of size 3 pixels on the side, erosion of A by B results in the square D being eliminated. After erosion, we can perform the method of *dilation* to repair some holes in the range of the shape of the car. By the way, if the shape of the car can be more complete, then the detection of the passing-by car will also be more precise.

As a result, the shape of a passing-by car in the omni-image is obtained. An example is shown in Figure 6.8, and the red regions are the passing-by car regions found by the above algorithm from Figure 6.4.



(a) (b)

Figure 6.8 An example of finding the region of the passing-by car. (a) Result from the omni-image taken by the upper omni-camera. (b) Result from the omni-image taken by the lower one.

6.3 Detection of Car Position in Real World

6.3.1 Transformation of a car model in real world into an omni-image

After the shapes of the passing-by car in the omni-images are known, we attempt to simulate the car shape in an omni-image in this study. We use a cuboid with a general size of a car model of TOYOTA Inc., whose size has the dimension of (C_L, C_W, C_H) = 482.5cm×182cm×147cm, where the parameters are the length, width, height of the car. To simplify the computation, we use the major three surfaces to construct the car shape in an omni-image. More specifically, the passing-by car is assumed to be seen from two directions: one is lateral in front of the video surveillance car, and the other is parallel aside the video surveillance car. Thus, the three surfaces which are the top, the right, and the behind of the cuboid are used to construct the car shape, as

shown in Figure 6.9.

(a) (b)

Figure 6.9 The cuboids we used. (a) A lateral passing-by car. (b) A parallel passing-by car.

If the position of the passing-by car in the WCS is known, we can use Equations (5.5) through (5.8) of the backward mapping method proposed in Section 5.2.1 to get the corresponding car position in the ICS. An example of a cuboid shape placement in an omni-image using these equations is shown in Figure 6.10.

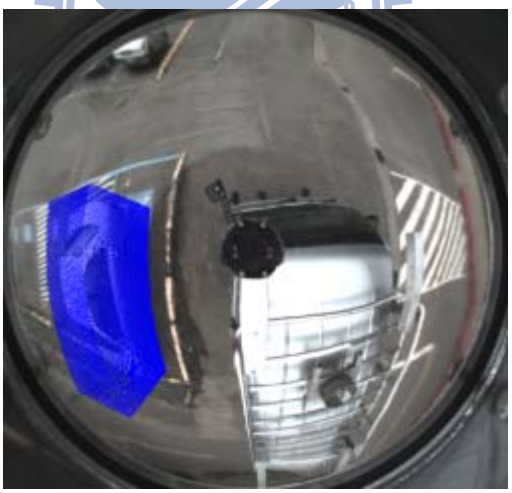


Figure 6.10 An example of cuboid shape placement in an omni-image using forward mapping.

However, by observing the experimental data, the upper half car body is susceptible to light intensity and its shape is always apparent to detect. On the other hand, the lower half car body is sometimes in the shadow, so its shape with respect to the upper half one is comparatively harder to detect. Accordingly, we use a cuboid (C_L,

C_W, $C_H/2$) which simulates the upper half car body. As shown in Figure 6.11(a), the cuboid shape is fragile. We can use erosion and dilation described in Section 6.2.2 to solve this problem and let the template matching described in the next section to work correctly, as shown in Figure 6.11(b).

(a) (b)

Figure 6.11 A cuboid shape in an omni-image. (a) Without erosion and dilation. (b) With erosion and dilation.

6.3.2 Detection of car position by template matching

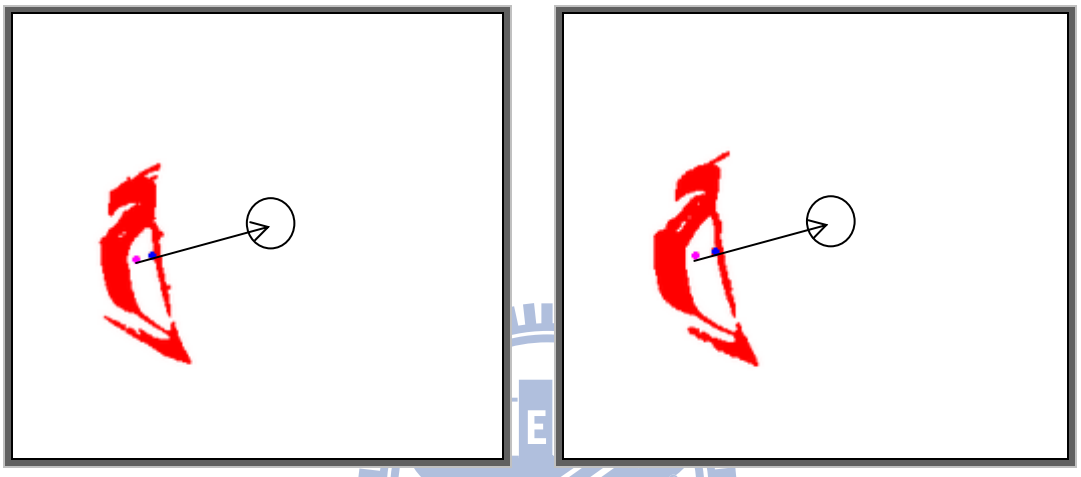
An approximate position of the passing-by car is needed to know for constructing the corresponding car shape in the omni-image. As shown in Figure 6.12, the centroid (m_x, m_y) of the passing-by car region in the omni-image can be computed by Eq. (6.4), which is marked as the pink point in the figure:

$$m_u = \frac{1}{n}\sum_v \sum_u p_u(u,v); \qquad m_v = \frac{1}{n}\sum_v \sum_u p_v(u,v), \tag{6.4}$$

where $p(u, v)$ is a marked point at coordinates (u, v), $p_u(u, v)$ returns its u-axis value,

$p_v(u, v)$ returns its v-axis value, n is the total number of $p(u, v)$.

In a general case, we assume that the centroid is a point on the middle horizontal plane of the car body. If the centroid does not belong to the marked car body, we find the first edge point from the midpoint to the center of the omni-image, marked as the blue point in Figure 6.12.



(a) (b)

Figure 6.12 An approximate position of the passing-by car. (a) The omni-image taken by the upper omni-camera. (b) The omni-image taken by the lower one.

The edge point may be regarded an approximate position of the passing-by car we use, and the height of the edge point is assumed to be a half of the height of a general car. Then, we can use Equations (5.1) through (5.4) of the forward image mapping method proposed in Section 5.2.1 to get the corresponding position $P(X, Y, Z)$ of the edge point in the WCS, where $Z = C_H / 2$ and C_H is the height of a general car. According to the resulting image after region growing, shown as the red region in Figure 6.13, if the following equation, Eq. (6.5), is satisfied, the orientation of the passing-by car is decided to be parallel to the video surveillance car; otherwise, it is decided to be lateral:

$$|\mathrm{Max_J} - \mathrm{min_J}| >= |\mathrm{Max_I} - \mathrm{min_I}| \tag{6.5}$$

where $\mathrm{Max_I}$, $\mathrm{min_I}$, $\mathrm{Max_J}$, and $\mathrm{min_J}$ are the extreme corner coordinates of the car shape as illustrated in Figure 6.13. By using the position P and the orientation of the passing-by car, we can construct a cuboid shape on the corresponding position. Note that, we also record the center of the passing-by car (cu_1, cv_1) in the omni-image which is mapped from the center of the cuboid model.

Figure 6.13 Decision of the passing-by car orientation.

After obtaining the cuboid shape in the omni-image, we superimpose the cuboid shape on the passing-by car shape in the image and perform an AND operation. More specifically, the cuboid shape shifts within a region R of 100×100 pixels located at relative coordinates from (–50, –50) to (50, 50), and the number of the overlapping pixels between the cuboid shape and the passing-by car shape is counted. Then, the two shapes are decided to match at a spot where this number of overlapping is the maximum with a shift (S_u, S_v) with respect to the origin of the region R. After this way of template matching, we can obtain the center of the passing-by car by the following equation:

$$cu_1 = cu_1 + S_u;\ cv_1 = cv_1 + S_v. \tag{6.6}$$

And this position becomes more accurate in general. An example of the two shape

matching is shown in Figure 6.14.

The two cameras in a two-camera omni-directional imaging device perform the same process and the two corresponding centers of the passing-by car can be obtained. Then, we can derive the position of the passing-by car in the WCS using the method described in Section 4.3.3.

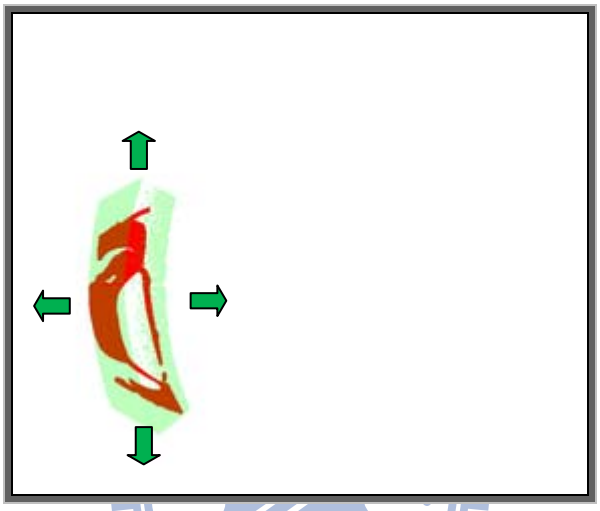


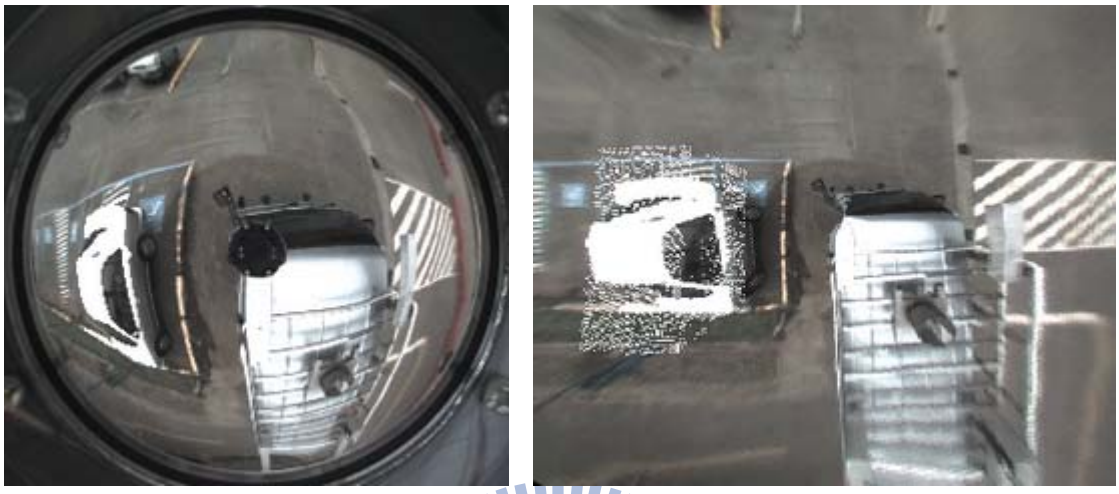
Figure 6.14 An illustration of template matching.

6.4 Passing-by Car Shape Superimposition and Ground Texture Filling in Top-view Image

6.4.1 Ground Texture Filling

A simple texture synthesis scheme is adopted in this study to fill ground texture into the shape area of a passing-by car, which is described in this section. In the above-mentioned section, the passing-by car shape in the omni-image can be obtained. First, we fill the white mark portions within the shape in the original omni-image

taken by the upper omni-camera, as shown in Figure 6.15. Because the top-view image is not so distorted as the omni-image, we fill the passing-by car shape with the ground texture after constructing the top-view image as described in Section 5.2.



(a) (b)

Figure 6.15 An illustration of ground filling for the passing-by car shape. (a) The omni-image. (b) The top-view image.

Specifically, to fill the white mark portions (the passing-by car shape) in Figure 6.15(b), for each passing-by car shape pixel p_c, we find the closest non-car shape pixel p_n from the left side to p_c, and use the patch (e.g., 5×5 pixels) of p_n to fill p_c. The concept is illustrated in Figure 6.16. This is a simple patch-based texture synthesis, and it is faster and smoother than a pixel-based method. This is the way we adopt to simulate ground texture and fill it within the car shape area.

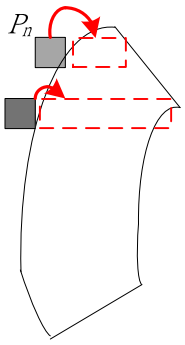


Figure 6.16 An illustration of ground filling in a patch-based texture synthesis.

6.4.2 Passing-by car shape superimposition

According to the direction of the passing-by car and the car position in the WCS which are given by the method in Section 6.3.2, after filling the ground texture, a top-view image of a car can be superimposed on the corresponding position in the top-view image, as shown in Figure 6.17.

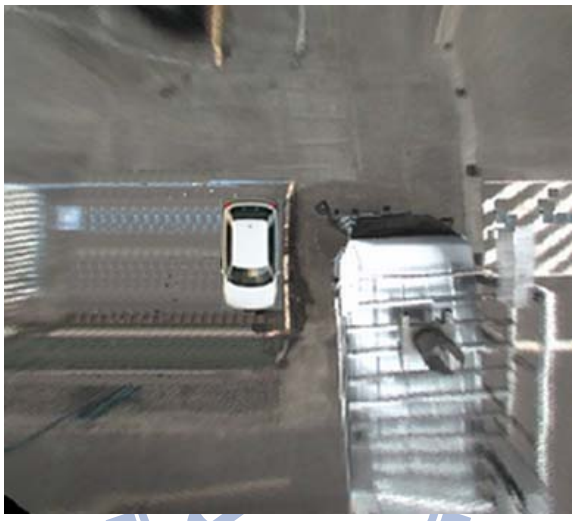


Figure 6.17 Passing-by car shape superimposition.

Chapter 7
Experimental Results and Discussions

In this chapter, we show some experimental results of the proposed video surveillance system using the pair of two-camera omni-directional imaging devices. The first are the results of calculating the positions P_{real} of real-world points when calibrating the omni-cameras. We compare the values of P_{real} which are calculated by the method proposed in Chapter 3 with those obtained by manual measurement.

The second are the results of constructing a top-view and the corresponding perspective-view images in an actual environment in National Chiao Tung University using the method proposed in Chapters 3 and 5.

The third are the results of calculating the positions of a passer-by, detected by the proposed method described in Chapter 4, in an actual environment in a parking area in National Chiao Tung University, and the computing results are compared with the manually-measured real positions of the passer-by.

The fourth are the results of calculating the positions of a passing-by car, detected by the proposed method described in Chapter 6, in an actual environment in a parking area in National Chiao Tung University, and the computing results are compared with the manually-measured real positions of the passing-by car.

The last are the results of an integrated system including all the above-mentioned techniques.

7.1 Experimental Results of Pano-mapping Process

In this experiment, a two-camera omni-directional imaging device was affixed on the ground and images were taken by it of the calibration pattern as shown in Figure 7.1. We obtained the values of the real-world point positions P_{real} by the method mentioned in Chapter 3 and by manual measurement simultaneously.

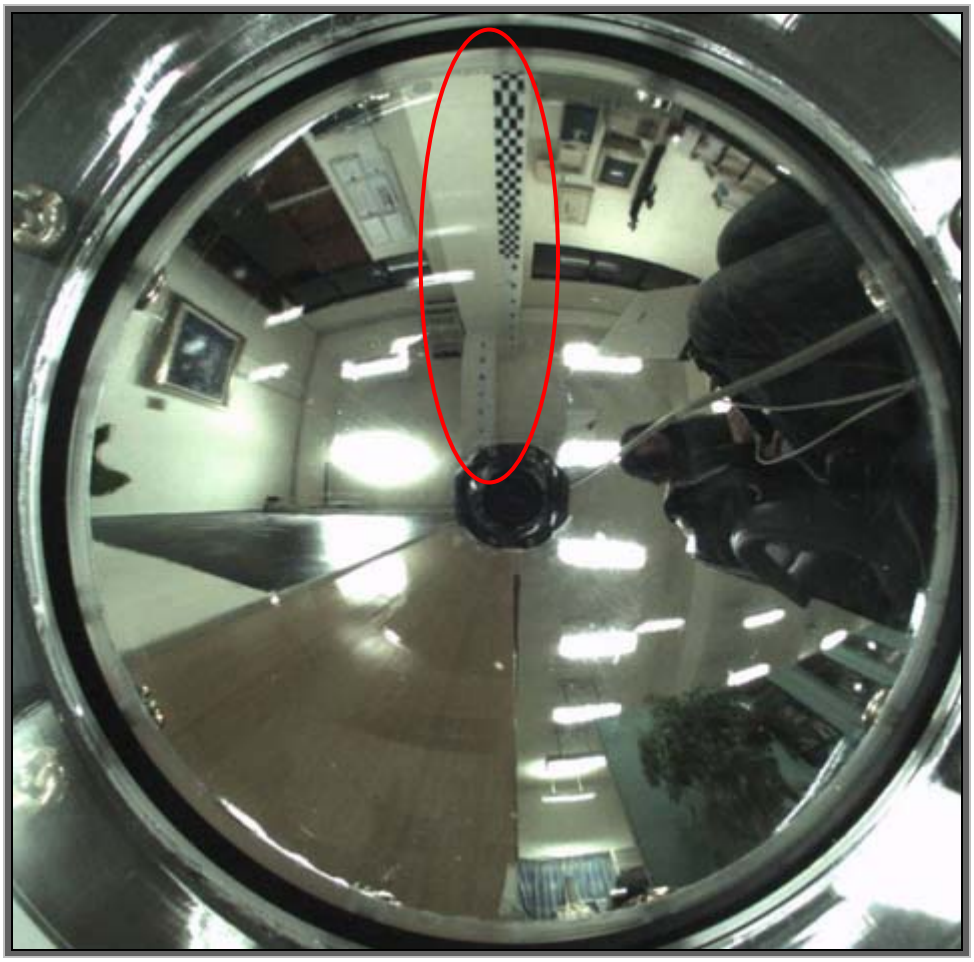
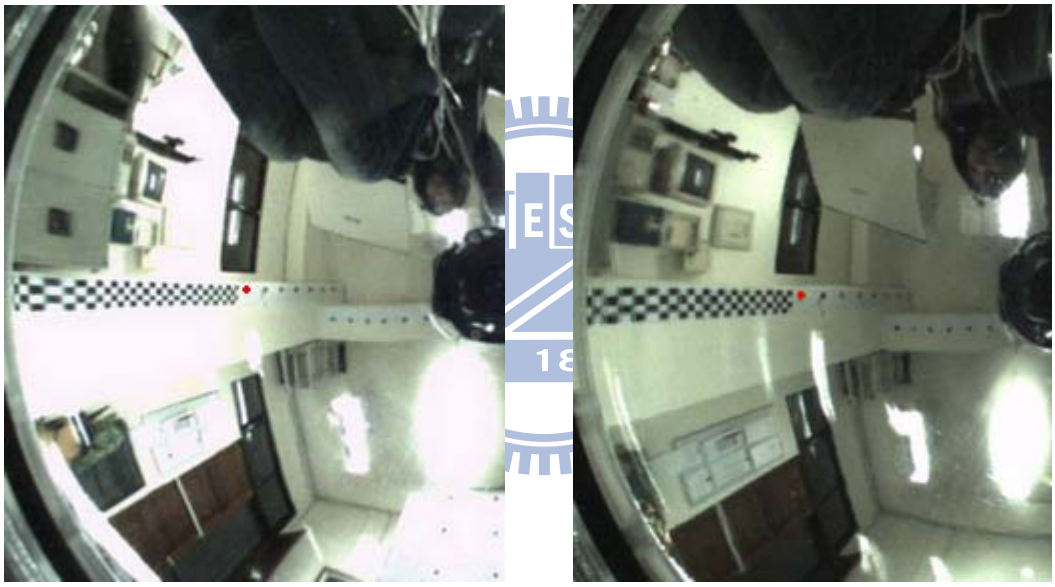


Figure 7.1 An omni-image of a calibration pattern taken by an omni-camera affixed on the ground.

The real-world width between every two consecutive intersection points on the patterns we used is 5cm obtained by manual measurement, and the bigger width

between every two black points on the patterns is 25 cm. Then, we picked out pairs of the corresponding pixels in two omni-images taken by a two-camera omni-directional imaging device, and an example is shown in Figure 7.2. We then calculated the horizontal distance *dw* and the vertical one *dh* in the WCS with respect to the center of the mirror by the method proposed in Chapter3. To increase the accuracy of the mapping method, we used four *radial stretching functions* to simulate four regions of omni-images, as shown in Figure 7.3. Note that the radial stretching function generated for Figure 7.1 corresponding to Table 1.



(a) (b)

Figure 7.2 An illustration of picking out pairs of corresponding pixels in two omni-images.

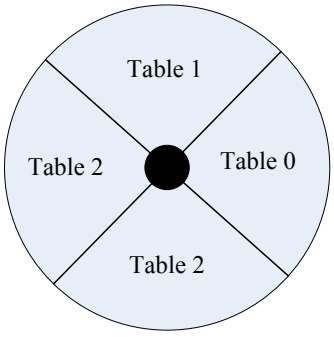


Figure 7.3 The regions corresponding to four radial stretching functions.

The results of calculation with the *ARTCAM-200SO* cameras which are within the two-camera omni-directional imaging device affixed on the right-front of the video surveillance car roof are shown in Tables 7.1 and 7.2. Because the cameras are facing up, the taken omni-images are opposite. More specifically, the left-hand side in the omni-image is just the right-hand side in the WCS, and vice versa. Therefore, the left-hand and front sides of the *ARTCAM-200SO* cameras are important, and it means that Table 1 and Table 2 are important to the imaging device. Note that the results of Table 1 and Table 2 are shown in Tables 7.1-2 and 7.3-4, respectively.

Table 7.1 The results of calculating the values of W_{real} by two ways (corresponding to Table 1 with camera 200SO).

No.	(1)Measured W_{real}		(2)Calculated W_{real}		Error (|(1)-(2)|/(1) %)	
	dw	dh	dw	dh	dw	dh
1	249	355	250.2	355.4	0.481928	0.112676
2	249	330	251.4	331.4	0.963855	0.424242
3	249	305	255	310.2	2.409639	1.704918
4	249	280	260.6	290.7	4.658635	3.821429
5	249	255	247.6	253.6	0.562249	0.54902
6	249	230	257.7	237.2	3.493976	3.130435
7	249	205	236.6	197	4.97992	3.902439
8	249	190	227.7	175	8.554217	7.894737
9	249	170	223	154.1	10.44177	9.352941
10	249	150	231.8	140.5	6.907631	6.333333
11	249	130	238	124.2	4.417671	4.461538
12	249	100	301.9	114.9	21.24498	14.9
13	227.959	319	219.5	300.2	3.71074	5.893417

Table 7.2 The results of calculating the values of W_{real} by two ways (corresponding to Table 1 with camera 200SO) (continued).

14	203.446	319	185.5	286.7	8.821031	10.12539
15	179.0678	319	148	266.5	17.34976	16.45768
16	154.888	319	156.1	323.3	0.782506	1.347962
17	131.0164	319	130.5	321.2	0.394129	0.689655
				AVG.	5.892625	5.35893

Table 7.3 The results of calculating the values of W_{real} by two ways (corresponding to Table 2 with camera 200SO).

No	(1)Measured W_{real}		(2)Calculated W_{real}		Error (|(1)-(2)|/ (1) %)	
.	dw	dh	dw	dh	dw	dh
1	249	355	249	355.2	0	0.056338
2	249	330	251.4	331.4	0.963855	0.424242
3	249	305	255	310.2	2.409639	1.704918
4	249	280	260.6	290.7	4.658635	3.821429
5	249	255	247.6	253.6	0.562249	0.54902
6	249	230	257.7	237.2	3.493976	3.130435
7	249	205	236.6	197	4.97992	3.902439
8	249	190	231.3	178.4	7.108434	6.105263
9	249	170	232.8	160.8	6.506024	5.411765
10	249	150	237.3	143.2	4.698795	4.533333
11	249	130	238	124.2	4.417671	4.461538
12	249	100	252.5	120	1.405622	20
13	227.959	319	219.7	303.9	3.623005	4.733542
14	203.446	319	185.5	286.7	8.821031	10.12539

Table 7.4 The results of calculating the values of W_{real} by two ways (corresponding to Table 2 with camera 200SO) (continued).

15	179.0678	319	165.2	292.8	7.74446	8.213166
16	154.888	319	156.1	323.3	0.782506	1.347962
17	131.0164	319	130.5	321.2	0.394129	0.689655
				AVG.	3.680585	4.659438

7.2 Experimental Results of Perspective-view Image Generation

In this experiment, we use an omni-image taken by the upper omni-camera of a two-camera omni-directional imaging device to construct perspective-view images of various directions by the method proposed in Chapter 3. The result is shown in Figure 7.4.

(a)

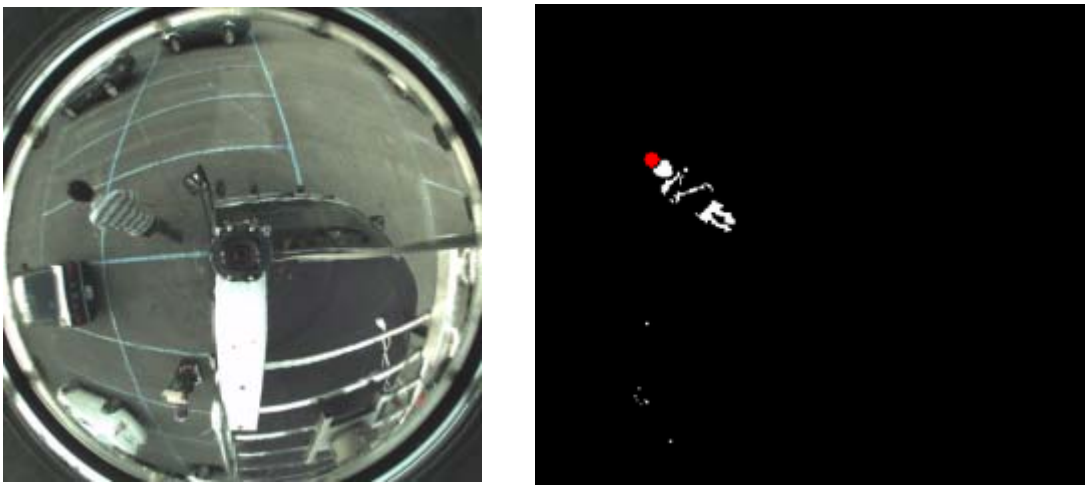
Figure 7.4 An experimental result of perspective-view image generation. (a) An omni-image. (b) The perspective-view image of the left direction. (c) The perspective-view image of the front direction. (d) The perspective-view image of the right direction.

(b) (c) (d)

Figure 7.5 An experimental result of perspective-view image generation. (a) An omni-image. (b) The perspective-view image of the left direction. (c) The perspective-view image of the front direction. (d) The perspective-view image of the right direction continued.

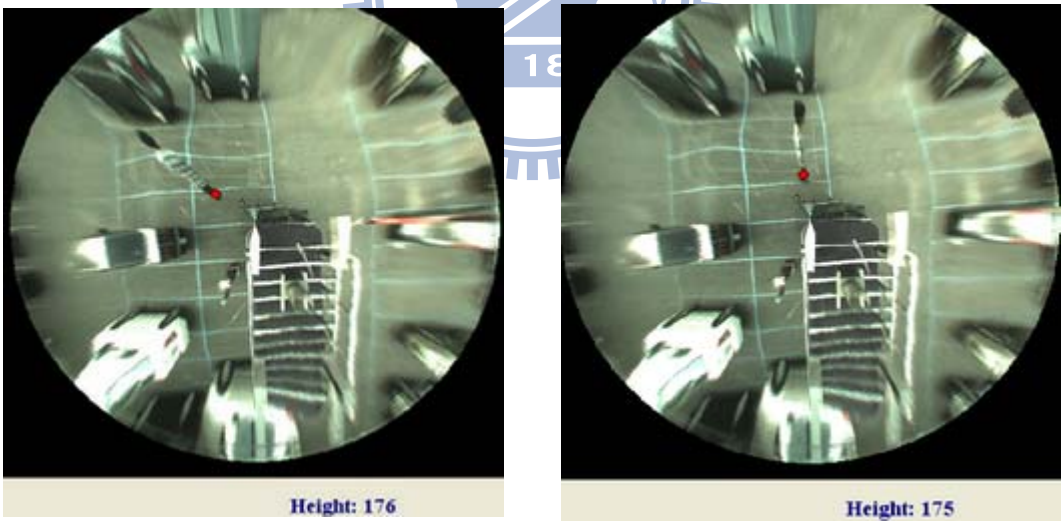
7.3 Experimental Results of Top-view Image Generation and Passer-by Detection

The environment for this experiment is an open space area in a parking area in National Chiao Tung University. Because of the property of imaging projection, after the region of a passer-by is found in the image, the body point which is farthest to the center of the image is the position of the passer-by's head, as shown in Figure 7.5. We constructed several top-view images by the method described in Chapter 5, calculated several positions of a passer-by simultaneously by the method described in Chapter 4, and marked a red point on each top-view image. A real example of detecting a passer-by around of the video surveillance car is shown in Figure 7.6.



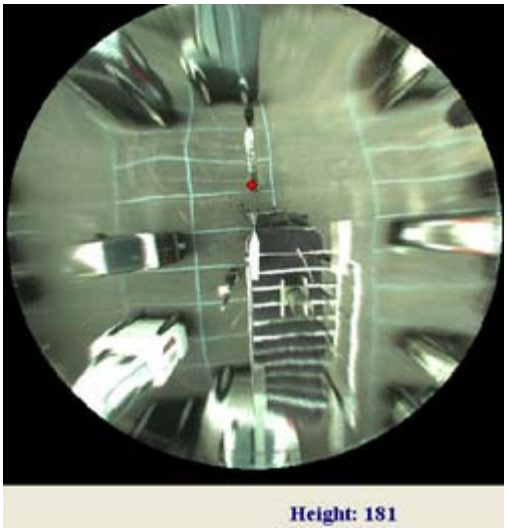
(a) (b)

Figure 7.6 Finding the position of a passer-by's head in the omni-image. (a) An omni-image. (b) Result of detection of passer-by's head.

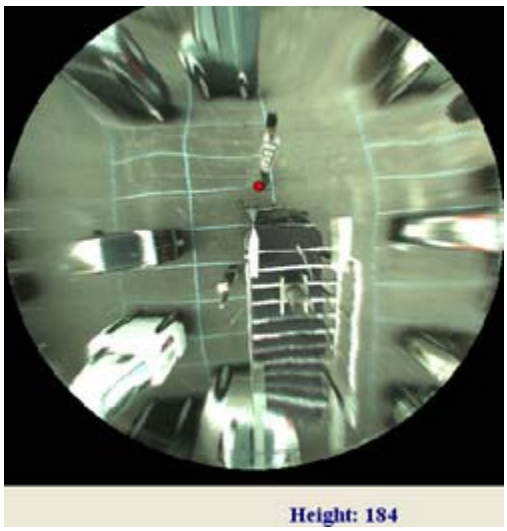


(a) (b)

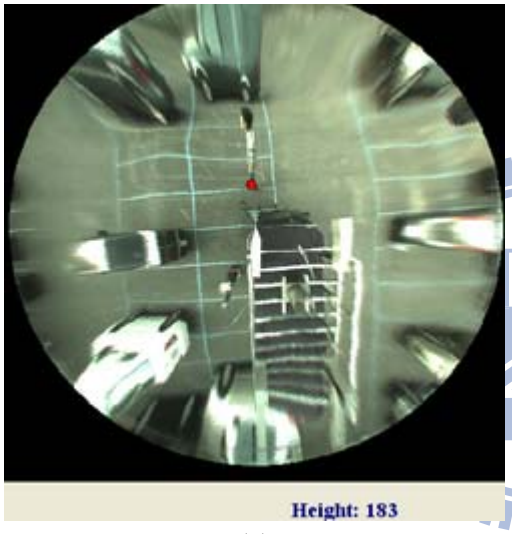
Figure 7.7 A real example of detecting a passer-by. (a)~(f) Detection results with red points marking the feet of the detected person.



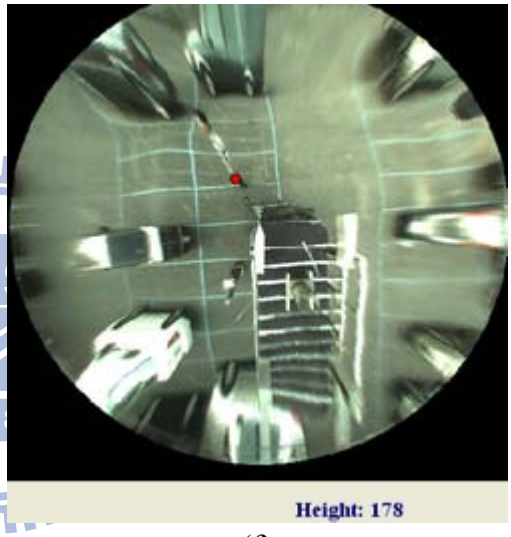
(c)



(d)



(e)



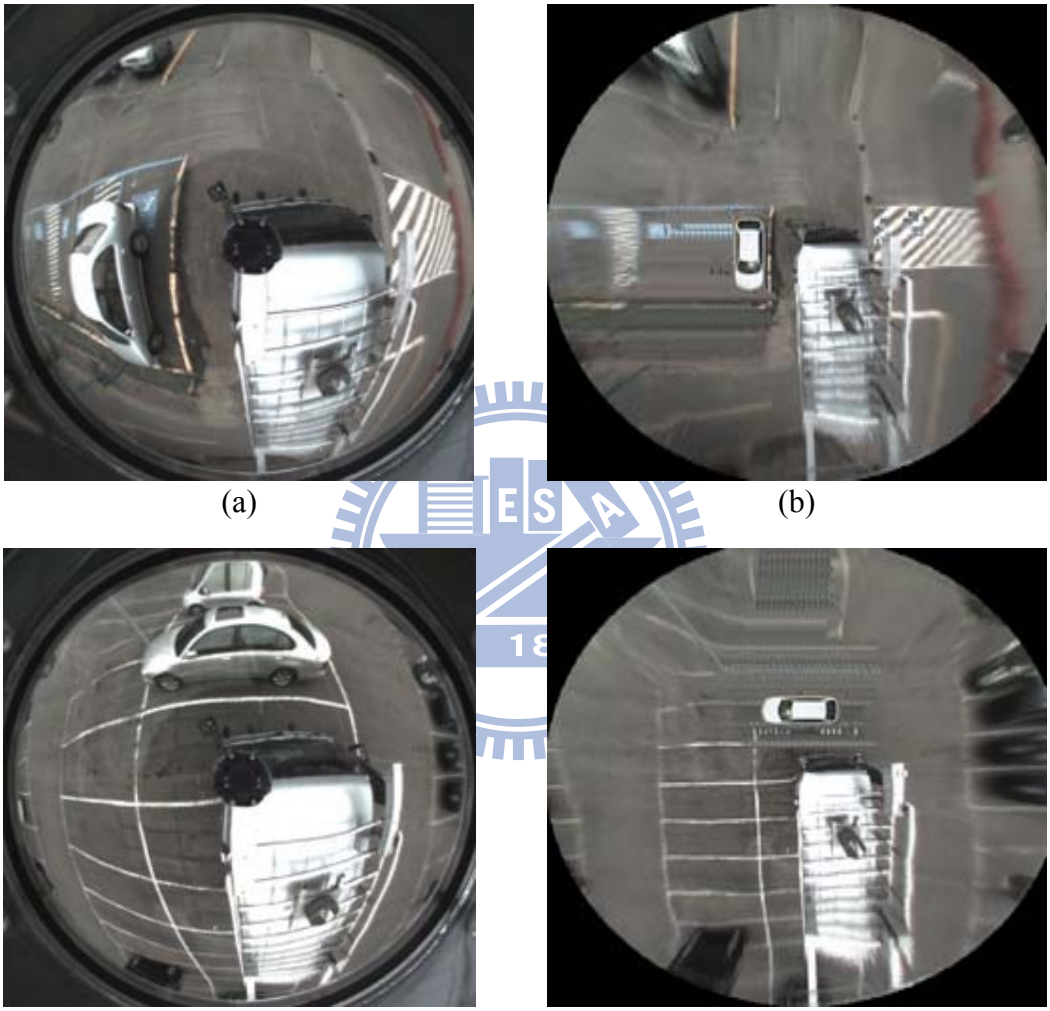
(f)

Figure 7.8 A real example of detecting a passer-by. (a)~(f) Detection results with red points marking the feet of the detected person continued.

7.4 Experimental Results of Passing-by Car Detection

The environment for this experiment is the same open space area mentioned previously in a parking area in National Chiao Tung University. We constructed several top-view images by the method described in Chapter 5, calculated several

positions of a passing-by car simultaneously by the method described in Chapter 6, and superimposed a top-view shape of a car on the corresponding position in the top-view image. A real example of detecting a passing-by car around of the video surveillance car is shown in Figure 7.7.



(a) (b)

(c) (d)

Figure 7.9 A real example of detecting a passing-by car. (a)~(d) Detection results with a top-view image of a car marking the position of the detected passing-by car.

7.5 Experimental Results of Integrated System

The environment for this experiment is an open space area in front of the Microelectronics and Information Systems Building in National Chiao Tung University. The integrated system used in the experiment included the capability to construct an integrated top-view image and a perspective-view image whose view direction is movable, and the capability of detecting a passer-by. More specifically, the imaging devices, after being affixed, were used to estimate relevant 3D data of objects (the detail was described in Chapter 2.3). Then, an integrated top-view image was obtained to show the surrounding environment of the video surveillance car from the top (the detail was described in Chapter 5). Also, any passing-by car were detected automatically and marked on the top-view image (the detail was described in Chapter 4). A corresponding perspective-view image may be generated for inspection (the detail was described in Chapter 3). Two real examples of images generated by the integrated system of the video surveillance car are shown in Figure 7.8.

(a)

(b)

Figure 7.10 Real examples of images generated by the integrated system. (a) Example 1. (b) Example 2.

7.6 Discussion

The proposed system utilizes a pair of two-camera omni-directional imaging device equipped on the video surveillance car roof to perform video surveillance. The system increases the mobility of the surveillance area. It is also utilized to detect suspicious people and vehicles around, and estimate the distance and height information of each approaching passer-by in suspect without walking out the car to measure relevant information manually. Top-view images and perspective-view images may be generated and utilized to see expanded environments outside the car. There are more applications of these techniques, such as providing various services of monitoring environments. If the environment is a scene of a car accident, we can take some pictures easily by the video surveillance car to obtain the relevant information of the scene, such as the position of a damaged car and the relative positions of all objects in the accident. If the environment is a critical security-alert region, the system can be used to detect any suspicious passers-by and obtain his/her position and height.

However, there are still some problems in the proposed system. Because high-resolution cameras are used in this study to get a good-looking perspective-view image, the heavy computation load makes the system operation and image generation not so smooth. To solve the problem, it might be necessary to improve the performance of the processes in the used laptop PCs or add more parallel computations. Furthermore, when tracking a person in the environment, if another person walks close to the person who is being tracked, we may not be able to calculate the right position of the person. To solve the problem, it might be necessary to add information of color and sample models of the person into this system. These problems are worth future researches.

Chapter 8
Conclusions and Suggestions for Future Works

8.1 Conclusions

In this study, we utilize a pair of two-camera omni-directional imaging devices equipped on a video surveillance car to develop a video surveillance system for monitoring the surrounding environment of a mobile video surveillance car. We have proposed several techniques and designed some algorithms to implement them, which are summarized in the following.

(1) *A 3D data acquisition method based on a space mapping approach* [17] has been proposed. In this study, we do not use the traditional projection-based transformation for camera calibration and image warping. Instead, we can construct the so-called pano-mapping tables [17] of the two-camera omni-directional imaging device we used in this study. By the pano-mapping tables, we can convert the coordinates of the points between the image coordinate system and the world coordinate system. Hence we can calculate the real-world position of a passer-by or a passing-by car.

(2) *A method for generation of specified perspective views by mouse clicks* has been proposed, by which it is proposed to make the user interface friendly for a user to change the view direction of the perspective-view image by moving and clicking a mouse. Thus, the user in the surveillance car can watch interesting views

without turning around his/her body to look at the outside. It makes the omni-camera seem as a rotatable PTZ camera.

(3) *A passer-by detection method based on the rotational invariance property of the omni-image* has been proposed, by which we can calculate the position and the height of a passer-by via the omni-images taken by the two-camera omni-directional imaging device on the video surveillance car roof. This method detects a passer-by's head using component labeling and dynamic grayscale offsetting which can solve image thresholding problems caused varying light intensities that affect the accuracy of the object detection result.

(4) *A method for top-view image construction with a backward mapping scheme* has been proposed, by which we can construct a "complete" top-view image using an omni-image and a corresponding r-ρ Table. The top-view image is generated by merging two top-view images taken from the upper omni-cameras of the pair of two-camera omni-directional imaging devices into a single one, so that the user can see the environment around the video surveillance car easily.

(5) *A network architecture between two laptop PCs* has been proposed, by which data can be communicated between the two laptop PCs. Both the proposed techniques of merging two top-view images into a single one and controlling the view direction of the generated perspective-view image need to exchange relevant data, and such tasks are completed through this architecture between the two laptop PCs.

(6) *A method for detection of a passing-by car in an omni-image* has been proposed, by which we can eliminate the ground region in an acquired omni-image and capture any passing-by car shape in the image by techniques like region growing, component labeling, etc. The approximate position and the direction of the passing-by car also can be obtained by the method. These data are important for

calculating the car position in the WCS and superimposing a top-view shape of a car on a top-view image.

(7) *A method for calculating the position of a passing-by car in the WCS* has been proposed, by which we can construct a car model in the real world and map it to a car shape in the omni-image, and then use template matching with the car shape and detect a passing-by car shape to get the accurate position of the car in an omni-image. Finally, we can estimate the position of the passing-by car in the WCS using a two-camera omni-directional imaging device.

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

8.2 Suggestions for Future Works

The proposed strategies and methods, as mentioned previously, have been implemented on a video surveillance car with a pair of two-camera omni-directional imaging device equipped on the car roof. According to experience obtained this study, in the following we make several suggestions and point out some related interesting issues, which are worth further investigation in the future:

(1) improve the design of the two-camera omni-directional imaging device to get a larger-sized expanded view and reduce the mechanical error more effectively;

(2) increase the speed of computation to achieve real-time detection, e.g., by parallel computing;

(3) improve the method of texture synthesis to get a smoother top-view image after the cars are replaced by their car model shapes but remember that the computation load should not be too heavy;

(4) increase the capability to detect more dangerous conditions;

(5) increase the capability of warning users immediately through cell phones or electronic mails;

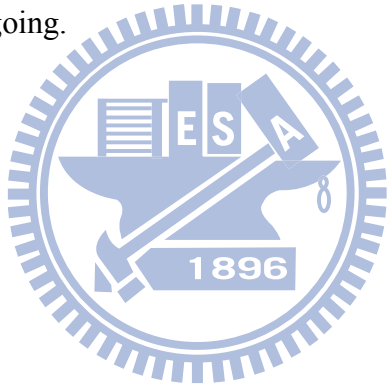
(6) increase the capability of constructing the adaptive mapping table automatically for cameras whose FOV's is not vertical to the ground;

(7) increase the capability of tracking multiple passers-by in the environment simultaneously;

(8) increase the capability of tracking multiple passing-by cars in the environment simultaneously;

(9) improve the accuracy of finding the head of the passer-by; and

(10) increase the capability of tracking passers-by and passing-by cars when the video surveillance car is on going.



References

[1] Y. C. Liu, K. Y. Lin, and Y. S. Chen, “Bird’s-eye View Vision System for Vehicle Surrounding Monitoring,” *Proceedings of Conference on Robot Vision*, pp.207-218, Berlin, Germany, Feb. 20, 2008.

[2] H. C. Chen and W. H. Tsai, "Optimal Security Patrolling by Multiple Vision-based Autonomous Vehicles with Omni-monitoring from The Ceiling," *Proceedings of 2008 International Computer Symposium*, vol. 2, pp.196-201, Nov. 2008.

[3] S. W. Jeng and W. H. Tsai, *A Study on Camera Calibration and Image Transformation Techniques and Their Application*, Ph.D. Dissertation, Institute of Information Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2007.

[4] H. Koyasu, J. Miura and Y. Shirai, “Real-time Omnidirectional Stereo for Obstacle Detection and Tracking in Dynamic Environments,” *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp.31-36, Maui, Hawaii, U. S. A., Oct. 29-Nov. 03, 2001.

[5] H. Ukida, N. Yamato, Y Tanimoto, T Sano, and H. Yamamoto, “Omni-directional 3D Measurement by Hyperbolic Mirror Cameras and Pattern Projection,” *Proceedings of 2008 IEEE Conference on Instrumentation & Measurement Technology*, pp.365-370, Victoria, BC, Canada, May 12-15, 2008.

[6] J. Gluckman, S. K. Nayar, and K. J. Thoresz, “Real-time Omnidirectional and Panoramic Stereo,” *Proceedings of DARPA98*, pp.299–303, 1998.

[7] J. Salvi, X. Armangué, and J. Batle, “A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation,” *Proceedings of Pattern*

Recognition, Vol. 35, No. 7, pp.1617-1635, July 2002.

[8] 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.

[9] C. J. Wu and W. H. Tsai, "Unwarping of Images Taken by Misaligned Omni-cameras without Camera Calibration by Curved Quadrilateral Morphing Using Quadratic pattern Classifiers," *Optical Engineering*, Vol. 48, No. 8, Aug. 2009.

[10] T. Mashita, Y. Iwai, and M. Yachida, "Calibration Method for Misaligned Catadioptric Camera," *Proceedings of IEICE Transactions on Information & Systems*, Vol. E89-D, No. 7, pp.1984-1993, July 2006.

[11] Y. Onoe, N. Yokoya, K Yamazawa, and H. Takemura, "Visual Surveillance and Monitoring System Using an Omnidirectional Video Camera," *Proceedings of ICPR98*, Vol. 1, pp.588-592, Sep. 1998.

[12] T. Mituyosi, Y. Yagi, and M. Yachida, "Real-time Human Feature Acquisition and Human Tracking by Omnidirectional mage Sensor," *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp.258-263, Sep. 2003.

[13] S. Morita, K. Yamazawa, and N. Yokoya, "Networked Video Surveillance Using Multiple Omnidirectional Cameras," *Proceedings of 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Vol. 3, pp.1245-1250, July 16-20, 2003.

[14] L. Matuszyk and A. Zelinsky, "Stereo Panoramic Vision for Monitoring Vehicle Blind-spots," *Proceedings of 2004 IEEE Intelligent Vehicles Symposium*, pp.31-36, June 14-17, 2004

[15] J. I. Meguro, J, I, Takiguchi, and Y, A, T, Hashizume, "3D Reconstruction Using

Multi-baseline Omni-directional Motion Stereo Based on GPS/DR Compound Navigation System," *International Journal of Robotics Research*, Vol. 26, No. 6, pp.625-636, June 2007..

[16] L.R. Burden, and J.D. Faires, "Numerical analysis," Brooks Cole, Belmont, CA, 2000. 7th edition, ISBN. 0534382169.

[17] W. H. Tsai, "Moment-preserving Thresholding: A New Approach," *Journal of Computer Vision, Graphics, and Image Processing*, Vol. 29, No. 3, pp.377-393, 1985.

[18] J. Y. Wang, "A Study on Indoor Security Surveillance by Vision-based Autonomous Vehicles with Omni-cameras on House Ceilings," M.S. Thesis, Institute of Multimedia Engineering, National Chiao Tung University, Hsinchu, Taiwan, June 2009.