Indoor Security Patrolling with Intruding Person Detection and Following Capabilities by Vision-Based Autonomous Vehicle Navigation

Student: Yu-Tzu Wang　Advisor: Dr. Wen-Hsiang Tsai

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

A vision-based vehicle system for security patrolling by human detection and tracking in indoor environments is proposed. A vehicle with wireless control and a web camera is used as a test bed. A robot arm is equipped on the vehicle to hold the camera at a higher position and is used to change the orientation of the camera. First, a camera calibration method is proposed by use of a technique of angular mapping, which is based on the concept of spherical coordinate system. Next, a human detection module and a human tracking module are proposed, which use a color feature of the face and that of the rough shape of the human body to recognize human beings. To track a target person, a cloth region intersection method is proposed to predict the motion of the person. In addition, a vehicle escape function is proposed, which is designed for the vehicle to move away from offensive strangers by a technique of *safe-distance keeping*. Good experimental results show the flexibility and feasibility of the proposed methods for the application of indoor security patrolling.

ACKNOWLEDGEMENTS

The author is in hearty appreciation of the continuous guidance, discussions, support, and encouragement received from his 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. Tsung-Yuan Liu, Mr. Chih-Jen Wu, Mr. Kuo-Feng Chien, Mr. Yen-Long Chen, Mr. Kai-Li Chiang, Miss Pei-Pei Chen, Miss Chia-Yu Hsu, and Miss Yi-Lin Wang for their valuable discussions, suggestions, and encouragement. Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during her thesis study.

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

CONTENTS

ABSTRACT i

ACKNOWLEDGEMENTS ii

CONTENTS iii

LIST OF FIGURES v

Chapter 1 1

1.1 Motivation 1

1.2 Survey of Relative Studies 2

1.3 Overview of Proposed Approach 3

1.4 Contributions 5

1.5 Thesis Organization 6

Chapter 2 7

2.1 Introduction 7

2.2 System Configuration 8

2.2.1 Hardware Configuration 8

2.2.2 Software Configuration 11

2.3 Human Detection Principle and Major Steps in Proposed Process 11

2.4 Human Tracking Principle and Major Steps in Proposed Process 13

Chapter 3 15

3.1 Introduction 15

3.1.1 Coordinate Systems 16

3.1.2 Viewing Angles of Camera 18

3.1.3 Directional Angles of Camera 20

3.2 Review of Perspective Projection 21

3.3 Calibration of Camera by Angular Mapping 24

3.3.1 Linear Angular Transformation 24

3.3.2 Nonlinear Angular Mapping 26

3.4 Vehicle Location Techniques Using Angular Mapping 31

3.4.1 2D to 3D Distance Transformation 32

3.4.2 Angle Transformation between Coordinate Systems 33

Chapter 4 35

4.1 Overview of Human Detection 35

4.2 Proposed Process of Human Detection 36

4.3 Human Detection by Faces 38

4.3.1 Skin Region Segmentation Using Color Classification................38
4.3.1.1 YC_bC_r Color Space 38
4.3.1.2 Adopted Skin Color Model..41
4.3.2 Detection of Human Face by Ellipse Shape Fitting.......................42
4.4 Human Body Detection by Motion Analysis...45
4.4.1 Motion Detection by Shift Tolerance Blockwise Frame Differencing 45
4.4.2 Human Body Detection...49
4.4.3 Some Experimental Results 51
Chapter 5...53
5.1 Basic Idea of Human Tracking 53
5.2 Proposed Process of Human Tracking 54
5.3 Extraction of Colors of Human Clothes...56
5.4 Human Tracking by Motion Analysis of Human Clothes.......................59
5.4.1 Recording of Human Motion 60
5.4.2 Motion Detection by Cloth Region Intersection...........................60
5.5 Applications to Stranger Tracking and Person Following 61
5.5.1 Stranger Tracking...61
5.5.2 Person Following 61
5.5.3 Experimental Results 62
Chapter 6...65
6.1 Overview...65
6.2 Principle of Escape 66
6.3 Distance Computation from The Vehicle to A Stranger..........................68
6.4 Method for Adjustment of Camera Orientation for Human Monitoring .71
Chapter 7...73
7.1 Experimental Results 73
7.2 Discussions 77
Chapter 8...78
8.1 Conclusions..78
8.2 Suggestions for Future Works...79
References...81

LIST OF FIGURES

Figure 1.1 A flowchart of proposed system 5
Figure 2.1 The vehicle Pioneer3 used in this study. (a) The front of the vehicle. (b) The flank of the vehicle..9
Figure 2.2 The structure of proposed system..10
Figure 2.3 An illustration of Human Detection Process 12
Figure 2.4 An illustration of the human tracking process...14
Figure 3.1 The coordinate systems used in this study. (a) The image coordinate system. (b) The vehicle coordinate system. (c) The spherical coordinate system. (d) The polar coordinate system. 17
Figure 3.2 The viewing angles of the camera. (a) The horizontal viewing angle *α*. (b) The vertical viewing angle *β*..19
Figure 3.3 An illustration of the image coordinate system confined by the ranges of the longitude and the latitude values...20
Figure 3.4 The pan angle of the camera. (a) $0 \le \theta_c \le \pi$ (b) $0 \ge \theta_c \ge -\pi$ 21
Figure 3.5 The tilt angle of the camera. (a) $0 \le \varphi_c \le \frac{\pi}{2}$ (b) $0 \ge \varphi_c \ge -\frac{\pi}{2}$ 21
Figure 3.6 Concept of perspective projection. 23
Figure 3.7 Image coordinate system mapping into real world....................................23
Figure 3.8 The relationship of image resolution and the viewing angles. (a) The horizontal direction. (b) The vertical direction. 25
Figure 3.9 An illustration of the image coordinate system confined by the ranges of image resolution...26
Figure 3.10 An illustrate of Attaching the lines on the wall. 29
Figure 3.11 A method of finding image coordinates of tessellated points in the grabbed image. (a) A grabbed image with tessellated points. (b) The tessellated points marked by yellow points...30
Figure 3.12 The points on the wall corresponding to of yellow points in Figure 3.11(b). 30
Figure 3.13 An illustrate of the interpolation method that a region contains the point *I* in the ICS. 31
Figure 3.14 The distance between the object and the vehicle. (a) The camera has no tilting, i.e. $\varphi_c = 0$. (b) The camera has a tilt angle of φ_c. 33
Figure 3.15 The rotation angle of the vehicle to obtain the tracking of the object......34

Figure 4.1 The proposed process of human detection. 37

Figure 4.2 YC_bC_r color model with Y = 126.......... 40

Figure 4.3 YC_bC_r color models with different Y values. 40

Figure 4.4 3D YC_bC_r color model in [23][24]. 41

Figure 4.5 Distribution of conditional probability density function of skin color in C_b-C_r plane [20]. 42

Figure 4.6 The elliptic skin model used in this study.......... 42

Figure 4.7 The detection of human face by ellipse shape fitting. (a) Input image. (b) Skin Segmentation. (c) Rectangular and elliptic mask. (d) Detected face region. 44

Figure 4.8 The image is segmented into blocks.......... 47

Figure 4.9 The searching window. 48

Figure 4.10 The blocks within in the searching window in I_r.......... 48

Figure 4.11 An example of blockwise frame difference images.......... 49

Figure 4.12 The order of the images: the current image, the reference image and the difference image using the proposed blockwise frame difference. (a) The person regions are close. (b) The person regions are far. (c) The person only exists in one of the images (d) No person exists in both images. 52

Figure 5.1 A cycle of the human tracking process. 54

Figure 5.2 The drawing of Vitruvian Man by Leonardo da Vinci. 58

Figure 5.3 The body proportion according to Vitruvius. 59

Figure 5.4 The application for stranger tracking. 62

Figure 5.5 The application for person following. 62

Figure 5.6 The consecutive images which were taken by the camera equipped on the vehicle when the vehicle tracked the target person. The order of the images is (a) through (j). 63

Figure 6.1 The process of escape for the vehicle.......... 67

Figure 6.2 The *safe-distance* for the vehicle. The vehicle is in an *unsafe state* when a person detected within the green circle. Else if the person is in the yellow area, the vehicle is in a *buffer state*. Else, it is in a *safe state*.......... 68

Figure 6.3 The illustration of the distance between the person and the vehicle. (a) Distance computing using length of the face. (b) Distance computing using length of the clothes. 69

Figure 6.4 The experimental result that the vehicle moves backward to the last position when the vehicle detected the fact that the person is too close. (a) The images grabbed by the camera equipped on the vehicle. (b) The images captured from a third person's viewpoint.......... 72

Figure 7.1 An interface of the experiment. The green box shows the image stream and

the blue box shows the input image at this moment. The yellow box shows the difference image and the red box shows the output image. 74
Figure 7.2 An experimental result of human body detection in the proposed system.(a) The input image. (b) The difference image. (c) The output image. 75
Figure 7.3 An experimental result of face detection and the extraction of the cloth. (a) The output image with a detected face region and the extracted cloth region by region growing. (b) The image of the extracted cloth. 75
Figure 7.4 An experimental result of human tracking using the intersection of the cloth images. (a) The input image. (b) The output image. 76

Chapter 1 Introduction

1.1 Motivation

In recent years, autonomous vehicle guidance by computer vision techniques in both indoor and outdoor environments has been widely and intensively studied. The vision-based autonomous vehicle has been used in numerous applications, including security patrolling. For the application of security patrolling, not only vehicle learning and navigation but also stranger intrusion detection and tracking are important research topics. Use of an autonomous vehicle equipped with a video camera is more "active" to accomplish the task of human tracking than use of traditional stationary cameras. The activeness nature comes from the dynamic movement capability of the vehicle, which makes monitoring of any corner in an environment possible.

Vision-based object detection and tracking is one of the most challenging issues in computer vision. Compared to other sensing devices, visual sensors like video cameras provide more information and higher-level intelligence to make logical decisions for vehicle control. However, along with the advantages of the vision-based system come additional difficulties. One of the difficulties is detection of moving objects in changing backgrounds. Detection and tracking of irregular-shaped targets, especially human beings, demands complicated solutions by pattern recognition and motion detection techniques.

Stationary cameras, usually fixed in houses or open-space environments, can only record scenes into videos passively, and abnormal situations in videos are usually observed by humans. Instead of inspecting videos by manpower, some existing researches of indoor security monitoring with stationary cameras focus on automatic detection of intruding humans in images or videos. It takes a lot of cameras to monitor every corner in a house. Hence, it is a good choice to use a vision-based autonomous vehicle, which has mobile ability, to reduce the use of cameras. Moreover, the vehicle can repeat identical steps, patrols all day, and needs only electric power.

Besides security patrolling, it is desired to design a navigation method by which a person can lead the vehicle to any desired place. Such a method creates more applications for the vehicle. The system can be used as an autonomous handcart or a shopping car to help humans carrying heavy stuff. Many systems of person following use a special mark for detecting the person's location, which puts more restrictions on person identification.

In this study, it is desired to develop a vision-based vehicle system for indoor security patrolling. We propose a system to be capable of security monitoring, including detection and tracking of intruding persons. Moreover, the system also provides a function for person following without the need of special marks.

1.2 Survey of Relative Studies

In ordinary video surveillances, human detection is performed by the use of fixed cameras with stationary backgrounds, which focuses on analysis of moving objects by frame differences and background establishment [1][2]. However, it is easier to detect moving objects in image sequences with stationary backgrounds. When moving cameras are used, these conventional frame difference-based techniques cannot be applied to detect moving objects since the scenes taken by the camera are unstable.

In the case of human detection by mobile cameras, many approaches have been proposed using various sensors and many kinds of features. To detect humans with non-stationary background, existing methods employ optical flow [3][4], direct camera motion parameter estimation [5], geometric transformation [6][7], dense stereo and motion measurements [8][9], or thermal infrared sensors for detection [10]. Unfortunately, the equipments of dense stereo and thermal infrared sensors are expensive. In some applications, direct camera motion parameters cannot be estimated. And geometric transformation is applied to detect motions under the assumption of uniform background.

Furthermore, many features have been used to recognize human beings, such as skin color [11], motion [12], depth [13], contour, and texture [14], etc. Since it is difficult to recognize humans in images using only a single feature, many systems use multiple features to distinguish human beings from other objects. Some systems extract skin color to detect human body parts, and track each part by motions [15][16]. Some systems detect moving parts first and analyze the shapes of the moving parts to detect human beings [17]. Heisele & Wohier proposed a system for detecting and tracking pedestrians by clustering of color image segmentation results [18]. By analyzing the cluster shape, they can find the regions of human legs. The system proposed by Papageorgiou extracts a set of wavelet features and applies a support vector machine (SVM) classifier [19].

1.3 Overview of Proposed Approach

In this study, we want to design a vision-based vehicle system for security patrolling by human detection and tracking in indoor environments. To achieve the goal, to recognize human beings and know their locations are necessary. The chief tools are the image captured by the camera equipped on the vehicle. An overall framework of the proposed system is illustrated in Figure 1.1.

The proposed approach of the vision-based vehicle system for security patrolling includes four major parts. The first part is camera calibration. In this study, we calibrate the camera by a technique of *angular mapping*, which uses the concept of spherical coordinate system. After fixing a camera on the vehicle, the angular mapping calibration technique using image analysis is used to compute the direction between the vehicle and a target used for calibration. Since each point in the image represents a unique light ray from the viewpoint into the camera, the proposed calibration method is to define the view angle of each point. According to these angles and the height of the camera, we can know the relative locations of targets in images.

The second and the third parts are a human detection module and a human tracking module. Before the vehicle tracks a target person, how to detect human beings from the image is essential. Identifying human beings in images a conventional but sophisticated topic. We use a color feature of the face and that of rough shape of human body to recognize human beings. To recognize faces in images, the color and the shape of a pattern are the main features to make decision. In addition, if a human body shape is found in motion detection, the system will pay attention to the humanlike object to verify it.

After an intruding person is detected, the system will remember his/her clothing and track him/her. To track the target person, we propose a cloth region intersection method to predict the motion of the person. Also, we record all the motions of the target person, and compute accordingly a parameter for the motion prediction of the target person. Unless the face of the target person is captured by the camera clearly, the system will come back to the detection mode again and again.

The last part is a vehicle escape function. Sometimes, the intruding strangers might try to attack the vehicle. We designed a function for the vehicle to escape from offensive strangers by a technique of *safe-distance keeping*. From the coordinates of the detected human face in the image, we can calculate the distance of a stranger. If the distance is shorter than the safe-distance we define in advance, the vehicle will be commanded to go to the last position in its path for escape.

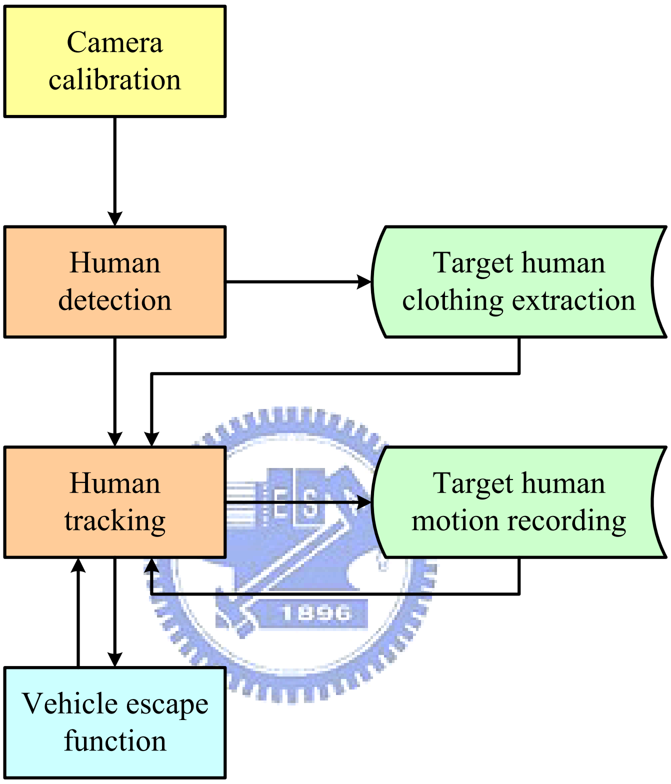


Figure 1.1 A flowchart of proposed system

1.4 Contributions

The major contributions of this study are summarized as follows.

(1) An angular mapping method for camera calibration is designed.

(2) A method of angular transformation from images to real world coordinates is proposed.

(3) A motion detection method for use in changing backgrounds by block-wised frame

differencing is proposed.

(4) A method of face detection by image analysis is proposed.

(5) A technique for prediction of target object movement is proposed.

(6) A technique for real-time human tracking is proposed.

(7) A method for escaping from a human being during vehicle navigation is proposed.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. The system configuration of the vehicle and the principles of human detection and tracking are described in Chapter 2. In Chapter 3, the proposed method of angular mapping calibration and the technique for vehicle location are described. In Chapter 4, the proposed methods for human detection by image analysis are described. The proposed techniques for human tracking will be described in Chapter 5. The function for escape from humans by safe-distance keeping is described in Chapter 6. Some satisfactory experiments results are shown in Chapter 7. Finally, some conclusions and suggestions for future works are given in Chapter 8.

Chapter 2
System Configuration and Navigation Principles

2.1 Introduction

A security patrolling system in an indoor environment is always required to be aware of stranger intrusion. Rather than waiting an intruding person to pass through the field of view of a monitoring system passively, tracking the intruding person actively by the use of an autonomous vehicle is more helpful.

To achieve this goal, a vehicle with wireless control and a web camera is used as a test bed for our research in this study. Because the target of detection is the human being, the camera must be fixed at a sufficient height from the ground. If the camera is located at a low position, it is difficult to observe the whole human body by the grabbed image. Hence, a robot arm is equipped on the vehicle to hold the camera at a higher position in this study. The arm is also used to change the orientation of the camera. The entire hardware equipment and software used in this study are introduced in Section 2.2.

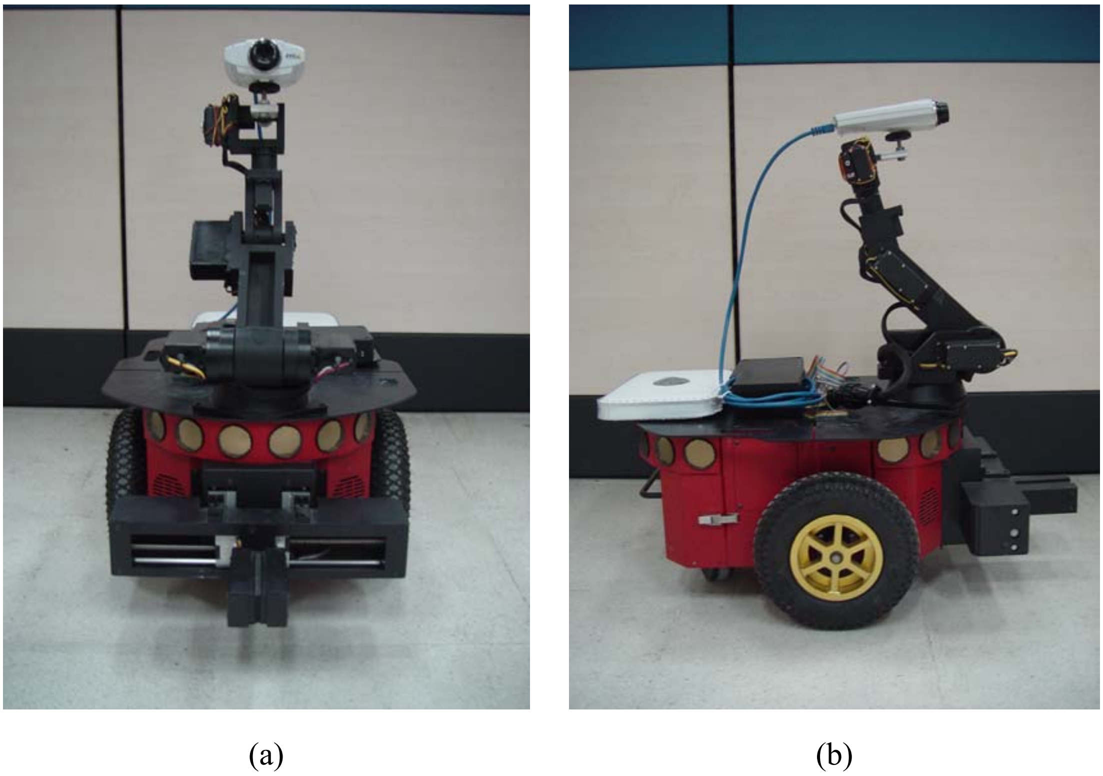
To conduct human detection in an unknown indoor environment, we have to define some features for human detection. Based on the features, we can analyze the grabbed images to detect whether a person is in the image. In Section 2.3, we will describe the human detection principle and process proposed in this study. After a person is detected using the information of his/her clothes and location, the system will enter the tracking mode. It will then compute the location and record the motion, of the target person, and moves the vehicle closer to the target person. The proposed principle and process will be introduced in Section 2.4.

2.2 System Configuration

In this study, we use the Pioneer 3, a rugged vehicle made by ActiveMedia Robotics Technologies Inc., as a test bed, on which an optional robotic arm is equipped, as shown in Figure 2.1. The arm can reach up to 50 cm (measured from the center of its base to the tip of its closed fingers). The tip of the arm is enabled to hold a digital web camera, AXIS210. The camera is IP-based with a build-in web server. We get the image captured and control the vehicle via wireless communication through computer networks.

2.2.1 Hardware Configuration

The entire navigation system is composed of three parts, as shown in Figure 2.2. The first part is a vehicle with a build-in wireless device and an embedded control system. The vehicle has an aluminum body of the size of 44cm×38cm×22cm with three wheels of the same diameter of 16.5cm. The vehicle can reach a forward speed of 160cm per second and a rotation speed of 300 degrees per second. There are three 12V batteries in the vehicle which supply the power. The vehicle can run 18-24 hours with the three fully charged batteries. By a user’s command, the embedded control system can control the vehicle to move forward or backward or to turn around. The system is also able to return some status parameters of the vehicle to the user. The robot arm has five degrees-of-freedom, residing on the top platform of the vehicle.



(a) (b)

Figure 2.1 The vehicle Pioneer3 used in this study. (a) The front of the vehicle. (b) The flank of the vehicle.

The second part is a digital web camera. To increase the height of the viewpoint, the camera is held by the robot arm. Since the arm has a carry weight limit, we have to choose a camera which meets to the limit. However, the lighter the camera is, the less the capability of the camera is. The camera we adopt has no panning, tilting, and zooming functions. Still we can change the direction of the camera by controlling the robot arm. Because the signal issued by the camera is digital, the grabbed image has no noise. Moreover, the camera is directly connected to an access point by a network cable for transmission of the captured image. The resolution of the image grabbed in our experiment is 320×240 pixels for the reason of raising image processing efficiency.

The third part is a remote control system in a desktop computer or a notebook PC. A kernel program can be executed on the remote control system to issue commands and get the status information from the vehicle and the robot arm. All commands transmitted to the vehicle or to the camera are through the wireless network. There is an access point in our test environment which meets the IEEE 208.11b standard to offer a bandwidth for the remote control system to communicate with the vehicle and the camera. Both the vehicle and the remote control system own wireless devices to connect to the access point, and the camera connects to the access point via a network cable. In other words, we use the access point as a medium to connect the three parts of the proposed navigation system.

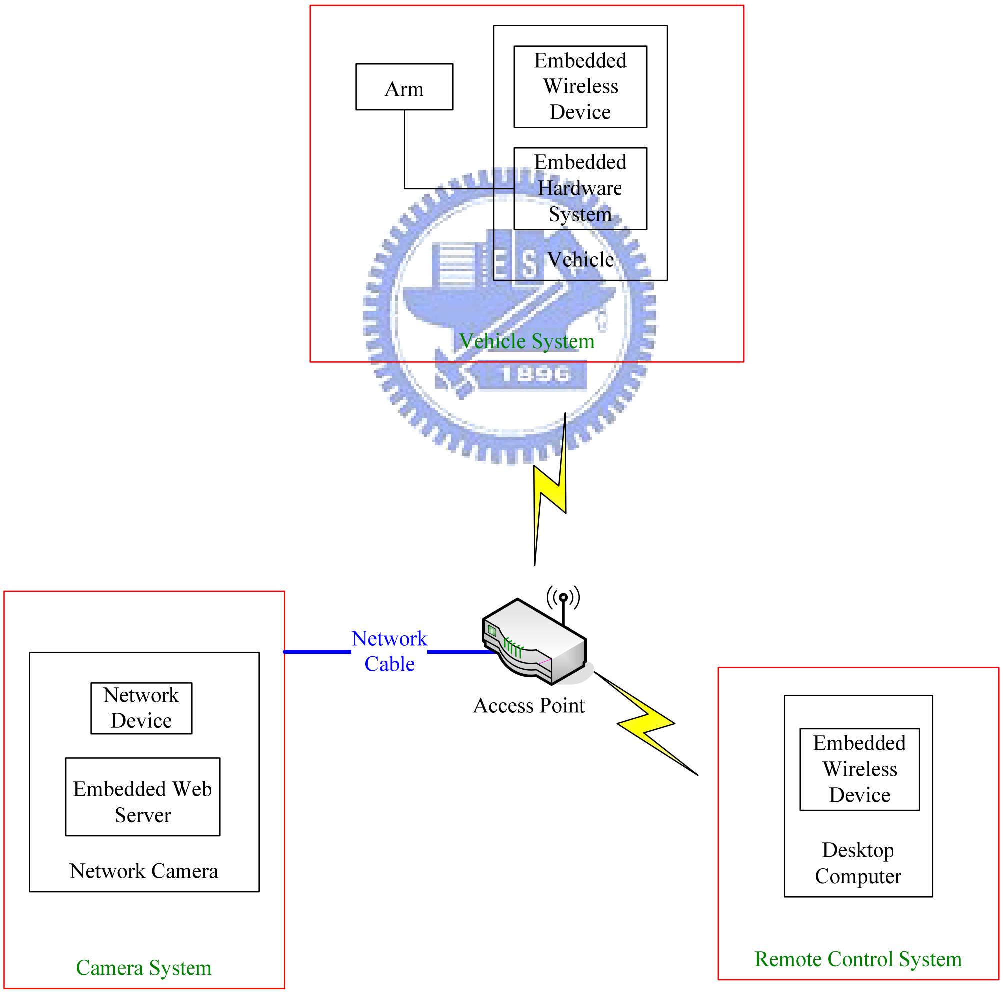


Figure 2.2 The structure of proposed system

2.2.2 Software Configuration

The ActiveMedia Robotics provides an application interface ARIA to control the mobile robot. ARIA is an objected oriented interface which is useable under Linux or Win32 in C++ language. We use the ARIA to communicate with the embedded system of the vehicle. And we use the Borland C++ Builder as a development tool in our experiments.

2.3 Human Detection Principle and Major Steps in Proposed Process

To conduct human tracking, we have to detect the existence of a person first. After a person is detected, the vehicle is moved to get close to the target. Since the web camera is the only sensor of the proposed system, we have to recognize human beings by image analysis.

In this study, we detect a person by face identification. There are three features, color, shape and motion, which we use for this purpose. When we get an image with the camera, we have to conduct two kinds of detections first at the same time. One is detection of skin-colored ellipses and the other is motion detection. Combining the results of these two kinds of detections, we define a moving skin color ellipse as a face of a person. Sometimes, the distance from the vehicle to the person is far, and the face cannot be detected in the grabbed image. Then we apply a human body detection process to the moving part to confirm if a person exists in the image or not. When an object similar to a human body is detected, the vehicle would be ordered to get closer to the object to take a clearer image of it.

If a face is detected by the system, the system will extract the information of the clothing and the system mode will be changed to the tracking mode and the vehicle would be commanded to do the tracking process. Otherwise, the system will stay in the detection mode. An illustration of the human detection process is shown in Figure 2.3.

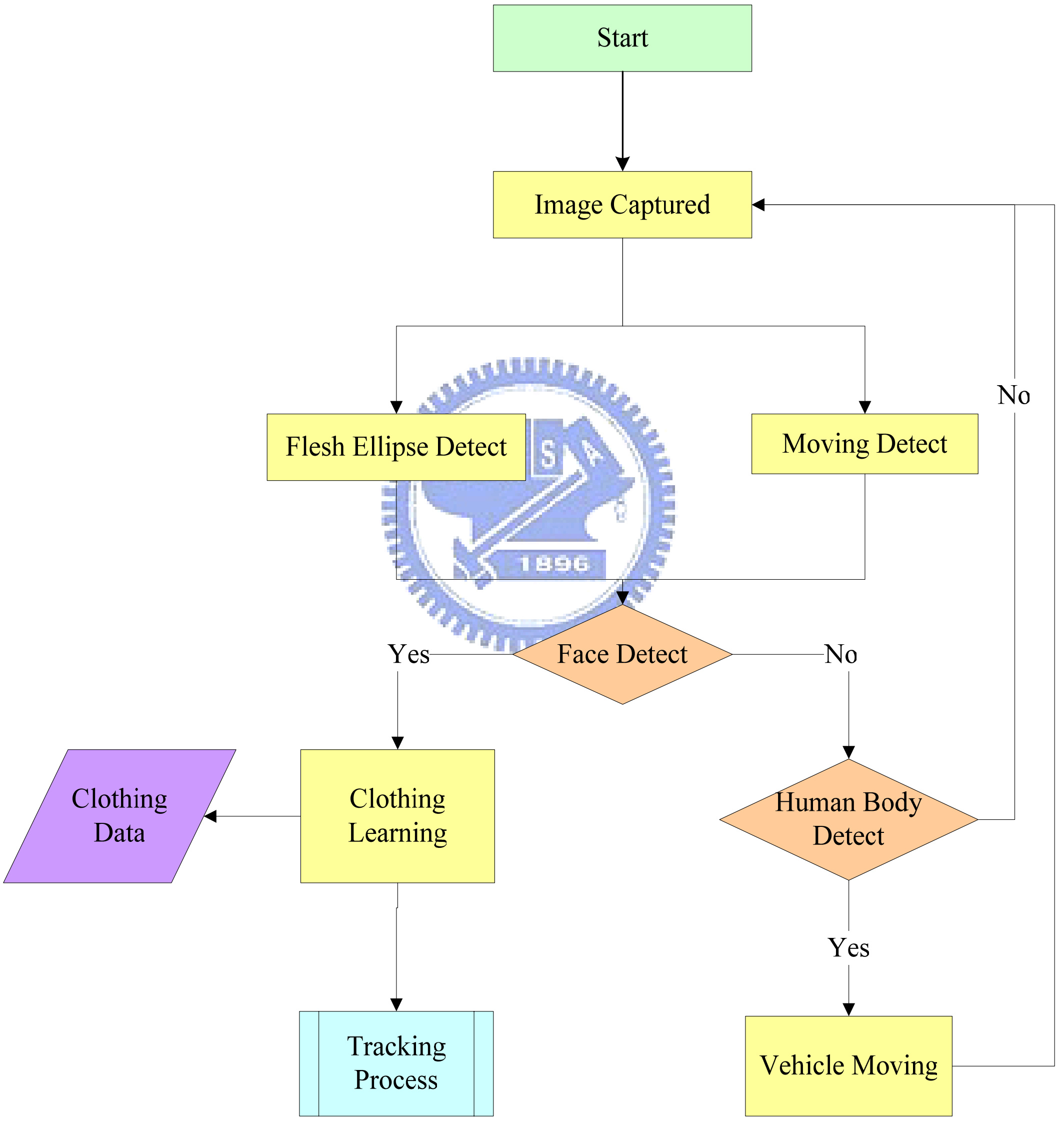


Figure 2.3 An illustration of Human Detection Process

2.4 Human Tracking Principle and Major Steps in Proposed Process

While the system is in the tracking mode, it means that we have the information about the person's clothing and the location of the person. We can search the new location of the person by the information of his clothing using of a prediction window.

Sometimes, a stranger might try to get close to the vehicle and attack it. For the reasons, we conduct face detection first to inference that the person faces the vehicle or turns his/her back to the vehicle. When a person faces the vehicle, a face will be detected, and we will compute the distance from the person to the vehicle. If the distance is shorter than a safe distance we define in advance, we order the vehicle to move backward to avoid possible attacks from the intruding person. Otherwise, we compute the moving direction of the person by information of the clothing and record the motion. Moreover, we propose a method of image intersection to predict the motion of the person. Combining the prediction and the record of the last motion of the person, we compute the possible movement of the person and command the vehicle to move to the person. The vehicle will then turn its head to aim at the person's face and go forward for a distance. When the system loses the cloth region for region intersection or when the person disappears in the grabbed image, the system will finish the tracking process of the person and enter the detection mode to continue security patrolling. An illustration of the human tracking process is shown in Figure 2.4.

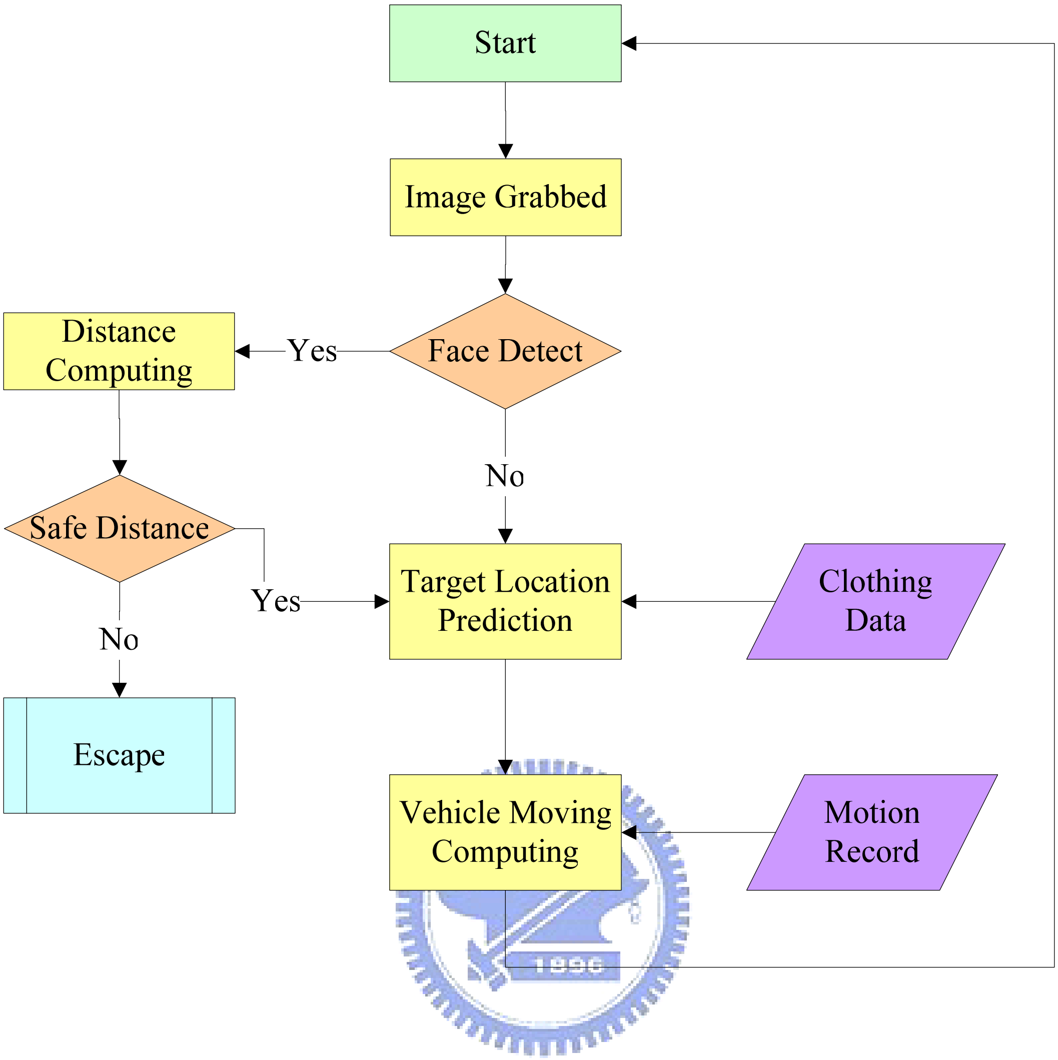


Figure 2.4 An illustration of the human tracking process

Chapter 3
Camera Calibration by Viewing Angles

3.1 Introduction

While a vehicle is tracking a person, the relative position and distance of the target person are important information for the tracking process. Since the camera is the only sensor of the proposed system and the techniques of human detection and tracking are based on visual perception, camera calibration and image analysis techniques for 2D images are indispensable in this study. Through imaging with the camera, 3D world coordinate systems are mapped into 2D image coordinate systems. However, there is ambiguity in the inverse mapping from 2D image coordinates to the 3D world coordinates. Each point in the image is the projection result of a light ray onto the image sensor. The light ray can be described by a longitude angle and a latitude angle of the ray in the 3D world space. To define the corresponding longitude and latitude angles (or simply called *longitude* and *latitude* in the sequel) of each point in images, we propose a method of *angular-mapping camera calibration*. Unfortunately, we cannot define the longitude and the latitude independently because of the existence of nonlinear camera distortion. In this study, we propose to use a 2D mapping method to achieve the goal of angular-mapping camera calibration.

We will review the principle of perspective projection which is involved in the basic idea of the proposed calibration method in Section 3.2. And we will discuss the detail of the proposed angular-mapping calibration method in Section 3.3. However, the position of the target object in the real world is what we are concerned with. We will propose a method to compute the location of the vehicle using angular mapping in Section 3.4.

Before describing the above-mentioned method, we first introduce the definitions of coordinate systems, and the viewing angles and directional angle of the camera for use in the study. We introduce the coordinate systems in Section 3.1.2 and the viewing angles of the camera in Section 3.1.1. Then, the directional angle of the camera will be introduced in Section 3.1.3.

3.1.1 Coordinate Systems

A few coordinate systems are used in this study, which describe the relative locations between the vehicle and objects. The coordinate systems are shown in Figure 3.1. The definitions of all the coordinate systems are stated in the following.

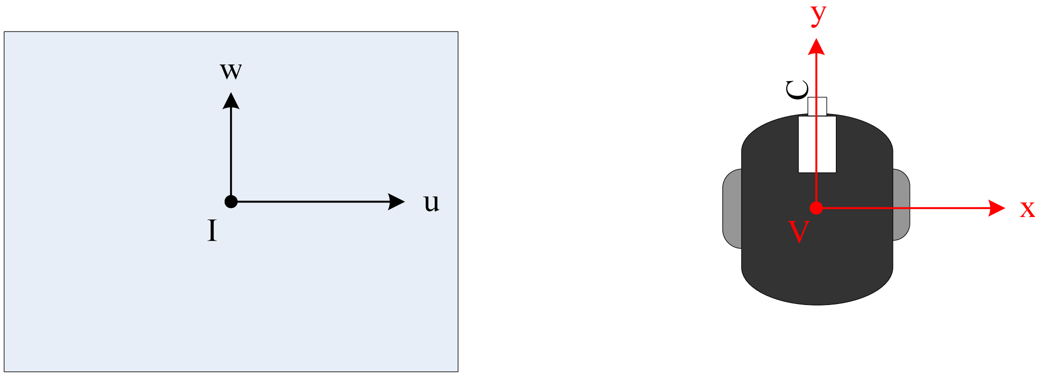
(1) Image coordinate system (ICS): denoted as (u, v). The uv-plane of the system is coincident with the image plane, and the image center, assumed to be the origin I of the ICS, will be described in detail in Section 3.2.

(2) Vehicle coordinate system (VCS): denoted as (x, y). The xy-plane is coincident with the ground, and the center of the VCS, the origin V, is taken to be the rotation center of the vehicle, which is the middle of the line segment connecting the two driving wheels. The x-axis of the system is parallel to the line segment of the two driving wheels and through the origin V. The y-axis is perpendicular to the x-axis and through V.

(3) Spherical coordinate system (SCS): denoted as (ρ, θ, φ). It is a 3D polar coordinate system. For convenience, we explain this system in terms of the 3D Cartesian coordinate system with coordinates (i, j, k). The ij-plane of the Cartesian

system is parallel to the *uv*-plane in the ICS. The origin *S* of the spherical system, which is also the origin of the Cartesian system, is the optical center of the camera. A point *P* at coordinates (i, j, k) in the Cartesian space is represented by a 3-tuple (ρ, θ, φ) in the spherical space. The value ρ with $\rho \geq 0$ is the distance between the point *P* and the origin *S*. The longitude θ is the angle between the positive *k*-axis and the line from the origin *S* to the point *P* projected onto the *ik*-plane. The latitude φ is the angle between the *ik*-plane and the line from the origin *S* to the point *P*. The range of θ is from $-\pi/2$ to $\pi/2$ and the range of φ is the same.

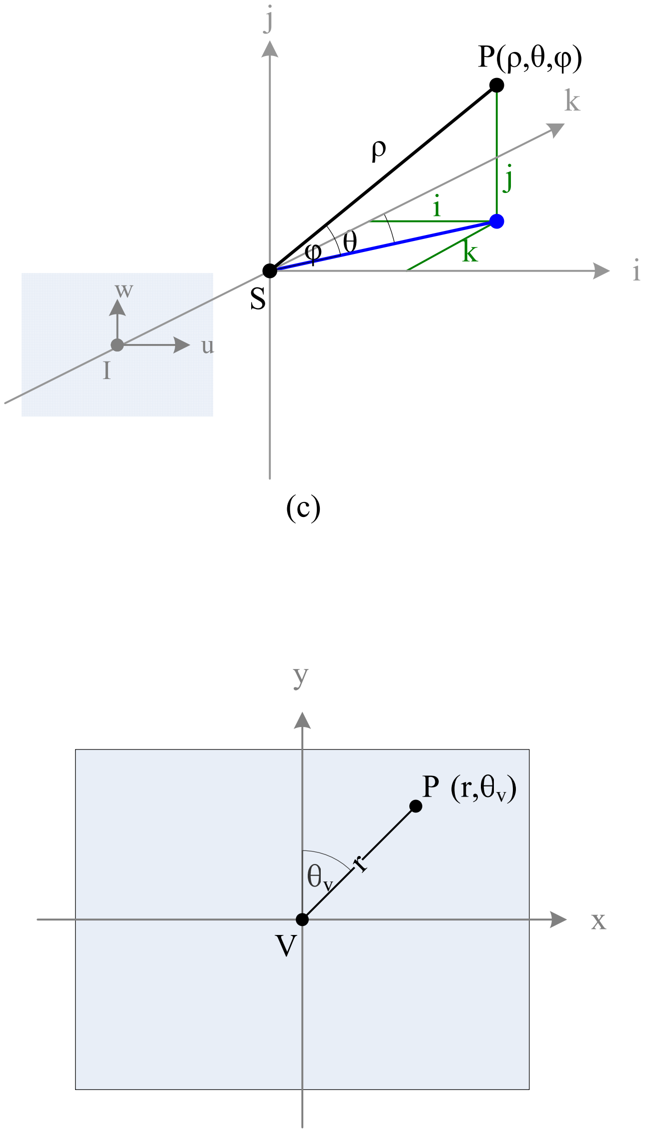
(4) Polar coordinate system (PCS): denoted as (r, θ_v). It is a 2D system which may be explained in terms of the VCS. A point *P* is represented by a 2-tuple $(r; \theta_v)$, where *r* is the distance from the origin *V* of the VCS to the point *P*, and θ_v is the angle between the positive *y*-axis and the line from the origin *V* to the point *P*. The range of θ_v is from 0 to π if *P* is in the first and second quadrants, and is from 0 to –π, else.



(a)

(b)

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



(c)

(d)

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

3.1.2 Viewing Angles of Camera

When a non-panoramic camera makes a projection on a plane, only a limited part of a scene can be imaged. The handiest parameter to describe the viewable area is the viewing angle, also called the *field of view*, of the camera. The horizontal viewing angle of a camera is the angle spanned from the left edge of the viewable region through the eyepoint to the right edge. The vertical viewing angle is the angle spanned from the top edge through the eyepoint to the bottom edge. As shown in Figure 3.2, we use α to denote the horizontal viewing angle and β the vertical viewing angle, of the camera. Each point in the image is formed by a different light ray through the optical center. As a result, each point in the image can be represented by two angles, the longitude and the latitude, of the light ray, described previously. In the image coordinate system, the u-axis has a longitude of 0^o and the v-axis has a latitude of 0^o. The latitude of the left edge of the image is $-\alpha/2$ and the latitude of the right edge of the image is $\alpha/2$. The longitude of the top edge of the image is $\beta/2$ and the longitude of the bottom edge of the image is $-\beta/2$. Then, each point in the image can be represented by a pair of longitude and latitude values in the range of the viewing angles as shown in Figure 3.3.

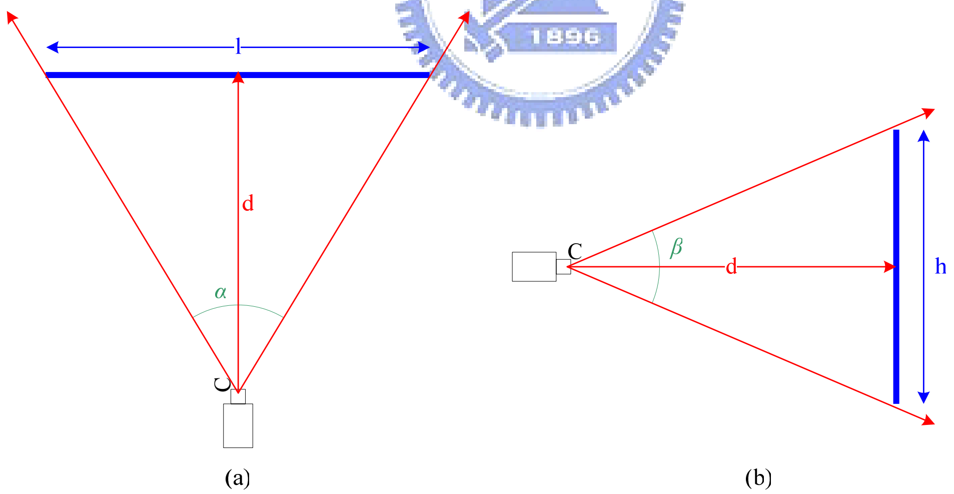


Figure 3.2 The viewing angles of the camera. (a) The horizontal viewing angle α. (b) The vertical viewing angle β.

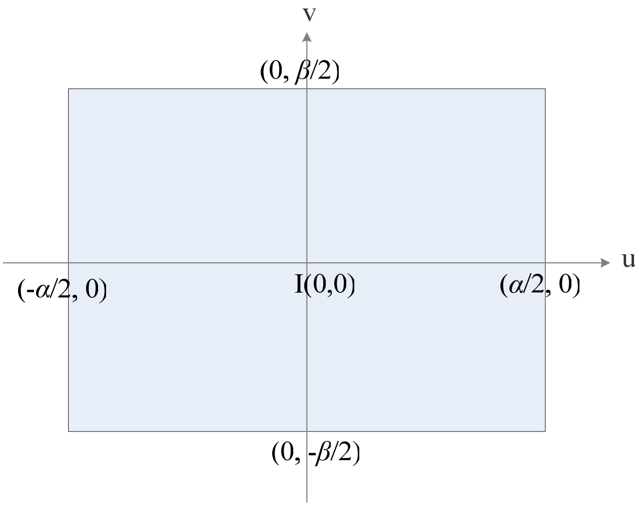


Figure 3.3 An illustration of the image coordinate system confined by the ranges of the longitude and the latitude values.

3.1.3 Directional Angles of Camera

There are two kinds of directional angles of a camera. One is the pan angle and the other is the tilt angle. The pan angle of the camera is defined in the VCS and denoted by θ_c. It represents the degree of rotation of the camera and is important for coordinate transformation.

We define the direction of the y-axis to be zero. The value of θ_c is exactly the angular span between the camera direction and the direction of the y-axis. The range of θ_c is between 0 and π if θ_c is in the first and fourth quadrants and between 0 and –π, else, as shown in Figure 3.4. The directional angle θ_c can be set as any value within the range.

The tilt angle of the camera is defined by the angle between the optical axis of the camera and the ground. The angle, denoted as φ_c, represents the degree of tilting of the camera. We define the angle to be zero when the optical axis of the camera is parallel to the ground. It is set to be zero at the beginning of a navigation session. The range of φ_c is between 0 and $\pi/2$ if the direction of the camera tilts up and is between 0 and $-\pi/2$ else, as shown in Figure 3.5.

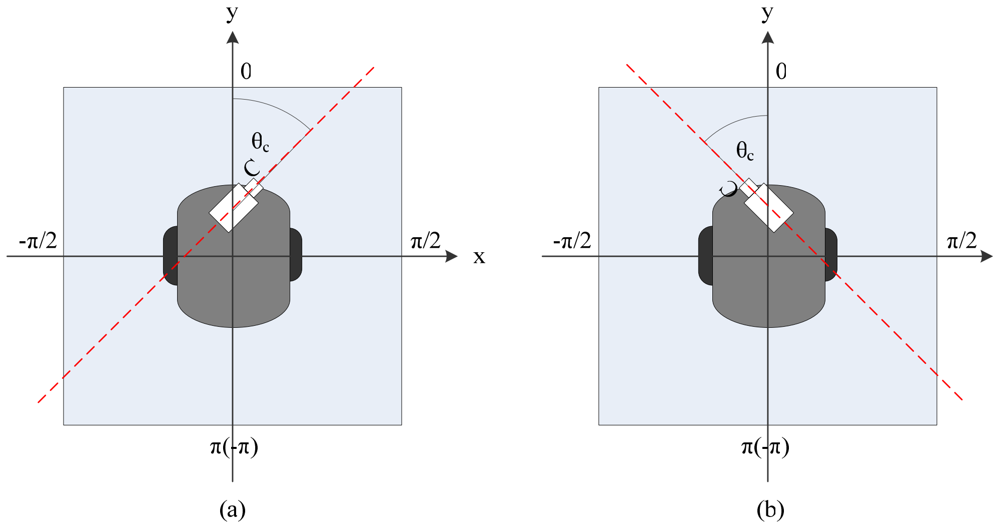


Figure 3.4 The pan angle of the camera. (a) $0 \le \theta_c \le \pi$ (b) $0 \ge \theta_c \ge -\pi$

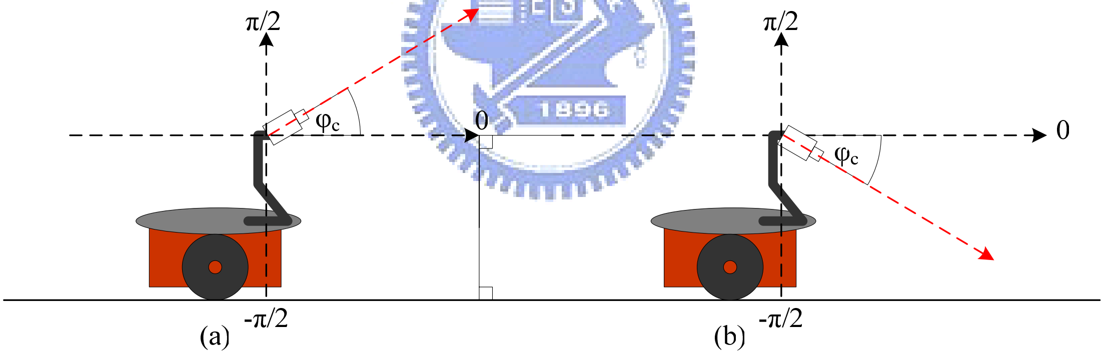


Figure 3.5 The tilt angle of the camera. (a) $0 \le \varphi_c \le \frac{\pi}{2}$ (b) $0 \ge \varphi_c \ge -\frac{\pi}{2}$

3.2 Review of Perspective Projection

Perspective projection is a phenomenon conveyed by a classical pinhole camera. A pinhole camera coordinate system is shown in Figure 3.6. The vector plane, which is formed by vectors x and y, is parallel to the image plane Π'. The point C' where it pierces Π' is called the image center, and the distance between C' and O is f'.

Let P denote a scene point with coordinates (x, y, z) and P' denote its image with coordinates (x', y'). Since the three points P, O, and P' are collinear, we get

$$\frac{x'}{x}=\frac{y'}{y}=\frac{f'}{z},$$

and so by the triangulation principle, we have

$$\begin{cases} x'=f'\dfrac{x}{z}, \\ y'=f'\dfrac{y}{z}. \end{cases} \tag{3.1}$$

As shown in Figure 3.7, consider the fronto-parallel plane Π_0 defined by $z = z_0$. For any point P in Π_0 we can rewrite the perspective projection Equation (3.1) as

$$\begin{cases} x'=-mx \\ y'=-my \end{cases} \text{where } m=-\frac{f'}{z_0}.$$

The image center of Π_0 is point C. Consider a point P in Π_0, and its image P'. Let $\overline{CP}=(\vec{x},\vec{y})$ and $\overline{C'P'}=(\vec{x'},\vec{y'})$. From the mapping of the image coordinate system to the world coordinate system, it is impossible to figure out the distance between eyepoint and the point P due to the inherent ambiguity of the light ray projection. However, the point P', the projection of the point P, can be represented by the longitude and the latitude of the point P in the real world. For example, let the longitude and the latitude of the point P be θ and φ, respectively. Then the point P' can be presented as well by the pair of the longitude and the latitude (θ, φ), as shown in Figure 3.7.

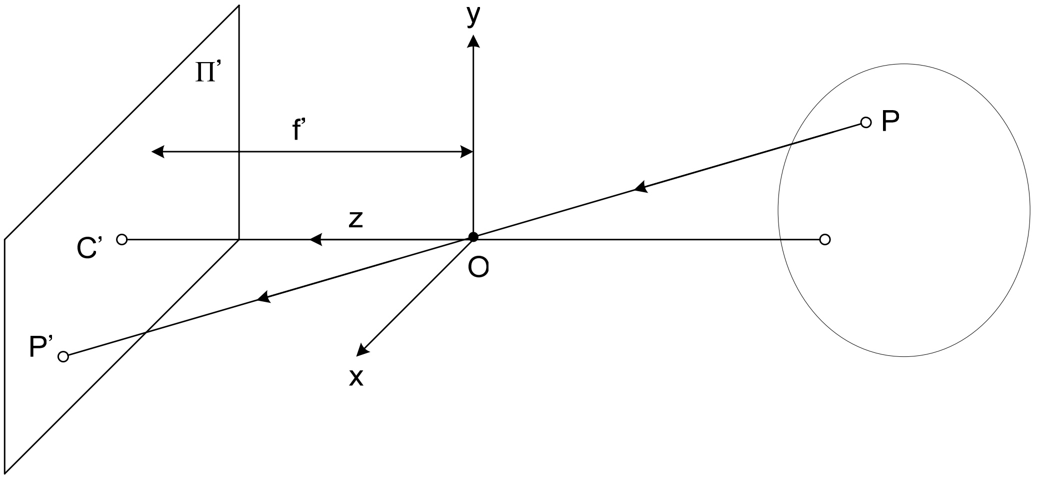


Figure 3.6 Concept of perspective projection.

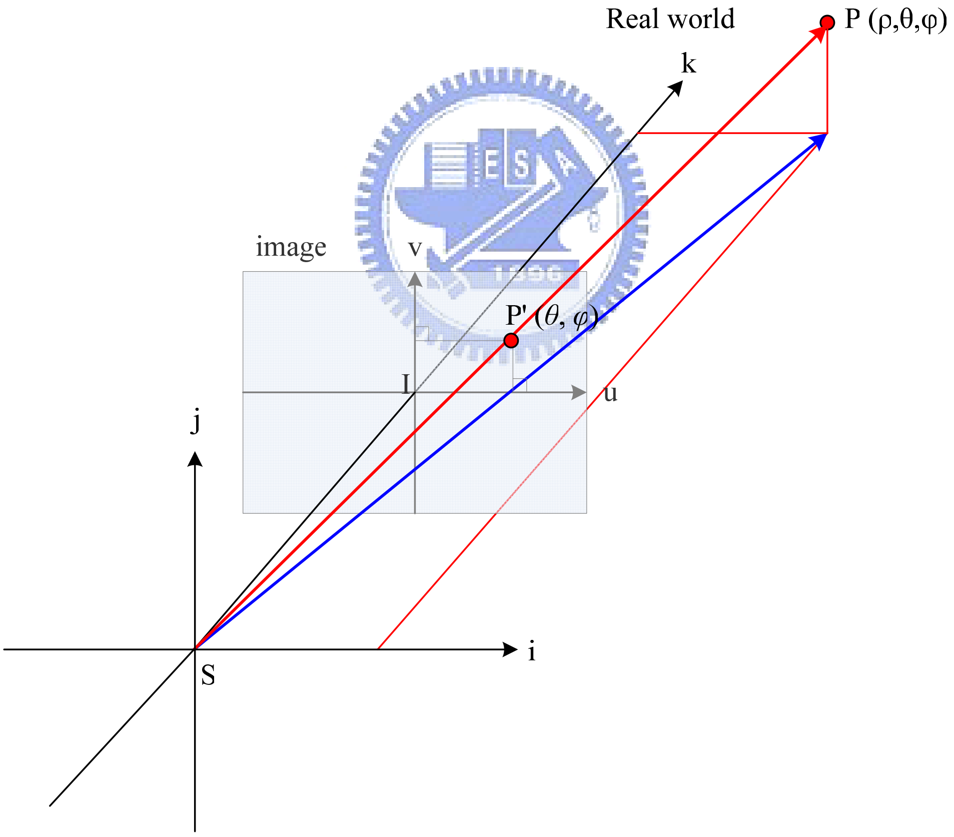


Figure 3.7 Image coordinate system mapping into real world.

3.3 Calibration of Camera by Angular Mapping

Each pixel in the image can be represented by a longitude value and a latitude value. With no camera distortion, each pair of longitude and latitude values is linearly mapped to a pixel in the image. It means that if we get a pixel $P(u, v)$ in the image coordinate system, according to the linear mapping, we can get the longitude of P via u and the latitude of P via v independently. Unfortunately, this assumption of no camera distortion is ideal, and the coordinate transformation from the real world to the 2D image is *nonlinear*. We have to considerate the u-coordinate and the v-coordinate at the same time to get the longitude and latitude values of the pixel in the image. We will introduce the concept of a linear angular transformation which is the basic idea of the proposed calibration method described in Section 3.3.1. To correct the effect of the distortion, we will propose a nonlinear angular mapping method in Section 3.3.2.

3.3.1 Linear Angular Transformation

First of all, assume the swing angle of the camera is zero and the pan and tilt angles, α and β, of the camera are known. The horizontal and vertical resolutions, W and H, of the image are also known, as shown in Figure 3.8. We define the image coordinate system as shown in Figure 3.9. Assume also that the center of the imaging screen and the center of the lens coincide, i.e. the image coordinates (0, 0) specify the center of the image. Using the information mentioned before, we can define each pixel in the image coordinate system by the longitude and the latitude values. We use the vector (θ, φ) to represent the corresponding longitude and latitude pair of the pixel $P(u, v)$ in the image. The detailed process of angular coordinate transformation is described in the following algorithm.

Algorithm 3.1 *Angular coordinate transformation by trigonometric function.*

Input: An image I with resolution W×H, and the horizontal and vertical viewing angles α and β, respectively, of the imaging camera.

Output: A longitude set $A = \{\theta_{\frac{-W}{2}}, \theta_{\frac{-W}{2}+1}, \ldots, \theta_{-1}, \theta_0, \theta_1, \ldots, \theta_{\frac{W}{2}-2}, \theta_{\frac{W}{2}-1}\}$ and a latitude set $B = \{\varphi_{\frac{-W}{2}}, \varphi_{\frac{-W}{2}+1}, \ldots, \varphi_{-1}, \varphi_0, \varphi_1, \ldots, \varphi_{\frac{W}{2}-2}, \varphi_{\frac{W}{2}-1}\}$ of the image points of I.

Steps:

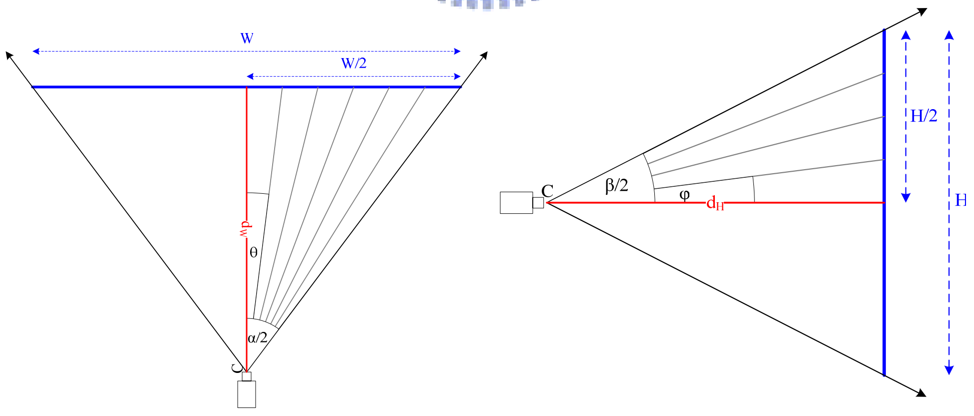
Step1. Compute d_W by $\tan\frac{\alpha}{2} = \frac{W/2}{d_W}$ and compute d_H by $\tan\frac{\beta}{2} = \frac{H/2}{d_H}$.

Step2. With d_W known, compute θ_n by

$$\tan\theta_n = \frac{i}{d_W}, \text{ where } i = \frac{-W}{2}, \ \ldots, \ -1, \ 0, \ 1, \ \ldots, \ \frac{W}{2} - 1.$$

Step3. With d_H known, compute φ_n by

$$\varphi_n = \frac{j}{d_H}, \text{ where } j = \frac{-H}{2}, \ \ldots, \ -1, \ 0, \ 1, \ \ldots, \ \frac{H}{2} - 1.$$



(a) (b)

Figure 3.8 The relationship of image resolution and the viewing angles. (a) The horizontal direction. (b) The vertical direction.

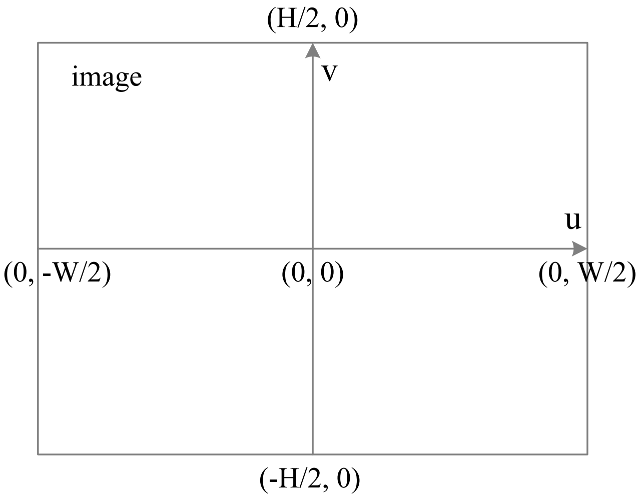


Figure 3.9 An illustration of the image coordinate system confined by the ranges of image resolution.

3.3.2 Nonlinear Angular Mapping

With the camera distortion, the linear angular transformation method mentioned in Section 3.3.1 is not applicable to get the correct corresponding angles θ_i and φ_i of the image coordinate system. To precisely obtain the angular transformation from the real world to the image, a real world data acquisition method by angular-mapping camera calibration is proposed. Since camera distortion exists both horizontally and vertically, we have to consider the horizontal and vertical directions at the same time while we compute the longitude and latitude values of each point in the image.

In the proposed method, we attach a grid with m vertical lines and n horizontal lines on a wall which is perpendicular to the ground. Then we have a real world point set $\boldsymbol{V} = \{V_{00}, V_{01}, ..., V_{mn}\}$, where $V_{ij} = (\theta_{ij}, \varphi_{ij})$ is a pair of the longitude and latitude values in the SCS of the point V_{ij} at the intersection of the ith vertical line and the jth horizontal line. The set $\boldsymbol{V}$ of intersection points is known in advance. And the corresponding point set $\boldsymbol{P} = \{P_{00}, P_{01}, ..., P_{mn}\}$ appearing in the image may be identified manually, where $P_{ij} = (u_{ij}, v_{ij})$ is a point in the ICS corresponding to point V_{ij}. The detailed process of the previously-mentioned nonlinear angular mapping is described as an algorithm in the following.

Algorithm 3.2 *The real location data acquisition by image taking and mapping.*

Input: An image I, as shown in Figure 3.11, and a set of longitude and latitude pair $\boldsymbol{V} = \{V_{00}, V_{01}, \ldots, V_{mn}\}$, as mentioned above.

Output: A point set $P = \{P_{00}, P_{01}, \ldots, P_{mn}\}$ in I corresponding to V, with P_{ij} corresponding to V_{ij}, where $i = 0, 1, \ldots, m$ and $j = 0, 1, \ldots, n$.

Steps:

Step 1. Attach a grid with m vertical lines and n horizontal lines on a wall, which is perpendicular to the ground.

Step 2. According to the interval distance of the grid on the wall and the distance D_{ic} from the wall to the camera, measure the longitude and latitude values of each point V_{ij} in the set **V**.

Step 3. Fix the interval of the longitude and the latitude to be 5° by adjusting the interval distance of each vertical line and each horizontal line of the grid on the wall based on the constraint of D_{ic} = 170cm as shown in Figure 3.10.

Step 4. Mark yellow points at the intersections of the lines, as shown in Figure 3.11.

Step 5. Record the coordinates of each yellow point $P_{ij}(u_{ij}, v_{ij})$ in the ICS and group all such points as a set $\boldsymbol{P}$.

Step 6. For each point P_{ij} in $\boldsymbol{P}$ in the image, manually identify the corresponding point V_{ij} in V with the longitude and latitude values θ_{ij} and φ_{ij}, as shown in Figure 3.12 and set up the mapping.

We have known the longitude and the latitude values of the yellow points in the image from Algorithm 3.2. To compute the longitude and the latitude values of the other pixels in the image, we use an interpolation method, as described in the following algorithm.

Algorithm 3.3 *Interpolation for computing viewing angles of any point.*

Input: An image point $I(u, v)$ in the ICS, the point set $\boldsymbol{P}$ and the point set $\boldsymbol{V}$ mentioned in Algorithm 3.2.

Output: The longitude and latitude values $V_I(\theta_I, \varphi_I)$ in the SCS of the image point I in the ICS.

Steps:

Step 1. Compute the coefficients a and b of the line equation $y = ax+b$ for lines L_0, L_1, L_2, and L_3 by using the image coordinates of the four endpoints, P_{ij}, $P_{(i+1)j}$, $P_{i(j+1)}$, and $P_{(i+1)(j+1)}$ in the following ways, as illustrated in Figure 3.13, where we assume (u_1, v_1) and (u_2, v_2) are two endpoints of any of L_i with $i = 0, 1, 2, 3$:

$$a = \frac{v_2 - v_1}{u_2 - u_1};$$

$$b = \frac{v_1 \times u_2 - v_2 \times u_1}{u_2 - u_1}.$$

Step 2. Decide whether the point I is in the region surrounded by the coordinates (u, v) of the four endpoints, P_{ij}, $P_{(i+1)j}$, $P_{i(j+1)}$, and $P_{(i+1)(j+1)}$ by substituting (u, v) for (x, y) of the line equation in the following way:

$$(a_0 \cdot u + b_0 - v) \cdot (a_2 \cdot u + b_2 - v) \leq 0; \qquad (3.1)$$

$$(a_1 \cdot u + b_1 - v) \cdot (a_3 \cdot u + b_3 - v) \leq 0. \qquad (3.2)$$

If the inequalities (3.1) and (3.2) are satisfied, the point I is regarded to be in the region; else, repeat Step 2 to check the next region.

Step 3. Define a line M_h which passes the point I and its slope is the average of the slope of L_1 and L_3, and so obtain two intersections $q(u_q, v_q)$ and $r(u_r, v_r)$ of

M$_h$ with *L*$_0$ and *L*$_2$ as shown in Figure 3.13.

Step 4. Define a line M_v which passes the point *I* and its slope is the average of the slope of L_0 and L_2, and so obtain two intersections $s(u_s, v_s)$ and $t(u_t, v_t)$ as shown in Figure 3.13.

Step 5. Use an interpolation method to obtain the longitude and the latitude (θ_I, φ_I) of *I* in the SCS by the following equations according to the geometric ratio principle:

$$\theta_I = \theta_{ij} + 5 \times \frac{d(q,I)}{d(q,r)}$$

$$\varphi_I = \varphi_{ij} - 5 \times \frac{d(t,I)}{d(s,t)}$$

where $d(a, b)$ is the distance from *a* to *b*.

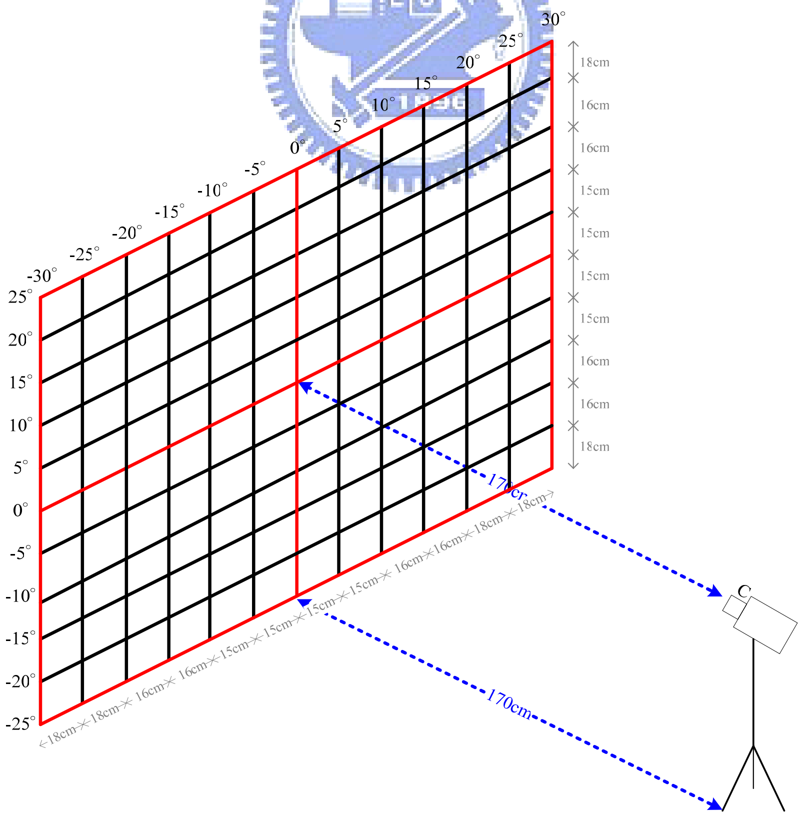


Figure 3.10 An illustrate of Attaching the lines on the wall.

By the interpolation method, each pixel in the image coordinate system can be mapped into the longitude and the latitude values in the SCS. By this information, we can get the angular position of objects in the image, as described in the following.

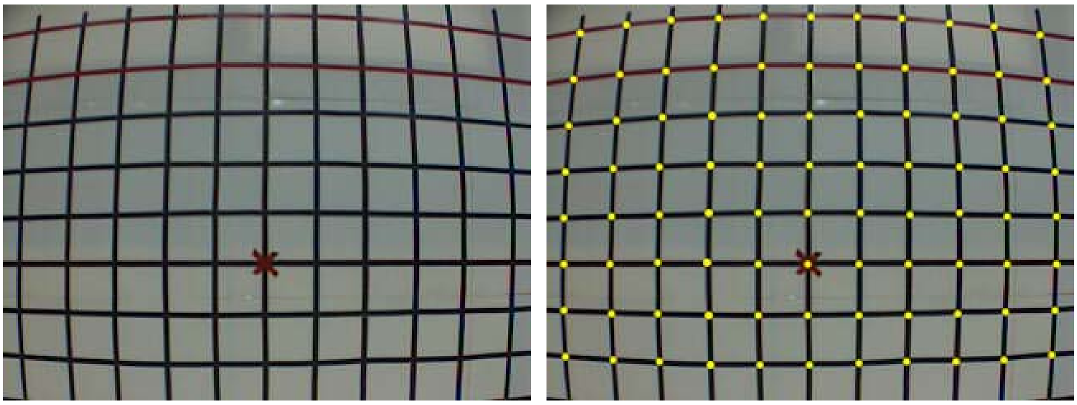

(a) (b)

Figure 3.11 A method of finding image coordinates of tessellated points in the grabbed image. (a) A grabbed image with tessellated points. (b) The tessellated points marked by yellow points.

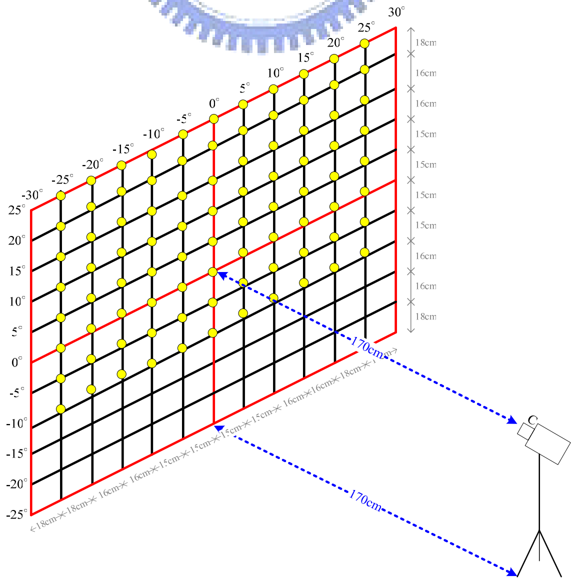


Figure 3.12 The points on the wall corresponding to of yellow points in Figure 3.11(b).

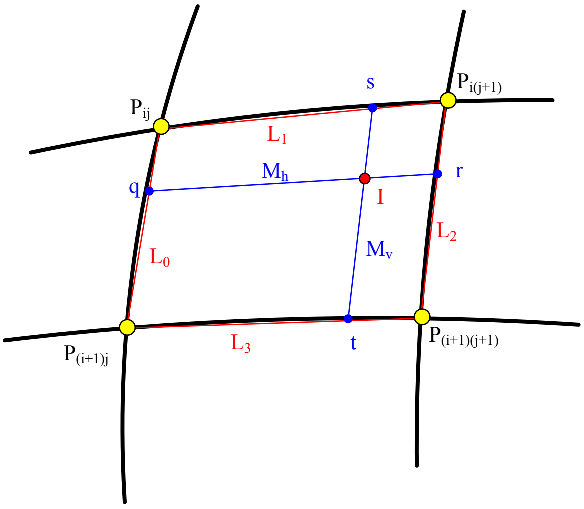


Figure 3.13 An illustrate of the interpolation method that a region contains the point *I* in the ICS.

3.4 Vehicle Location Techniques Using Angular Mapping

Using the calibration by the non-linear angular mapping method mentioned in Section 3.3.2, we can get the longitude and the latitude values of each pixel in the image. Since the camera is equipped on the arm of the vehicle, the directional angles of the camera are not always zero. When the pan angle of the camera is not zero, the directions of an object with respect to the camera and the vehicle also are both different. To track the target object correctly, we have to transform the directional angles with respect to the camera to ones with respect to the vehicle. In order to obtain the transformation, the information of the longitude and latitude values which are obtained from mapping the image coordinates is not enough. We must have more data to solve the ambiguity in distance estimation.

First, we have to know how to calculate the distance between the object and the camera by the latitude value and the distance from the camera to the ground, and we will discuss our method for this purpose in Section 3.4.1. And the way we propose for computing the directional angles of the camera with respect to the vehicle will be stated in Section 3.4.2.

3.4.1 2D to 3D Distance Transformation

As shown in Figure 3.14, we knew the coordinates of an object in the ICS after we take the image of the object. After the angular mapping, we have the information of the longitude and the latitude of each image point of the object in the SCS. Since we have the latitude value of the object and the knowledge about the height of the camera, if we know the ground contact point of the object in the image, we can compute the distance between the camera and the object. The algorithm is described in the following.

Algorithm 3.4 *Transformation of 2D image point to 3D real world point.*

Input: Camera height H_c, and the ground contact point $P(u, v)$ in the ICS of an object.

Output: The distance D_{oc} between the object and the camera.

Steps:

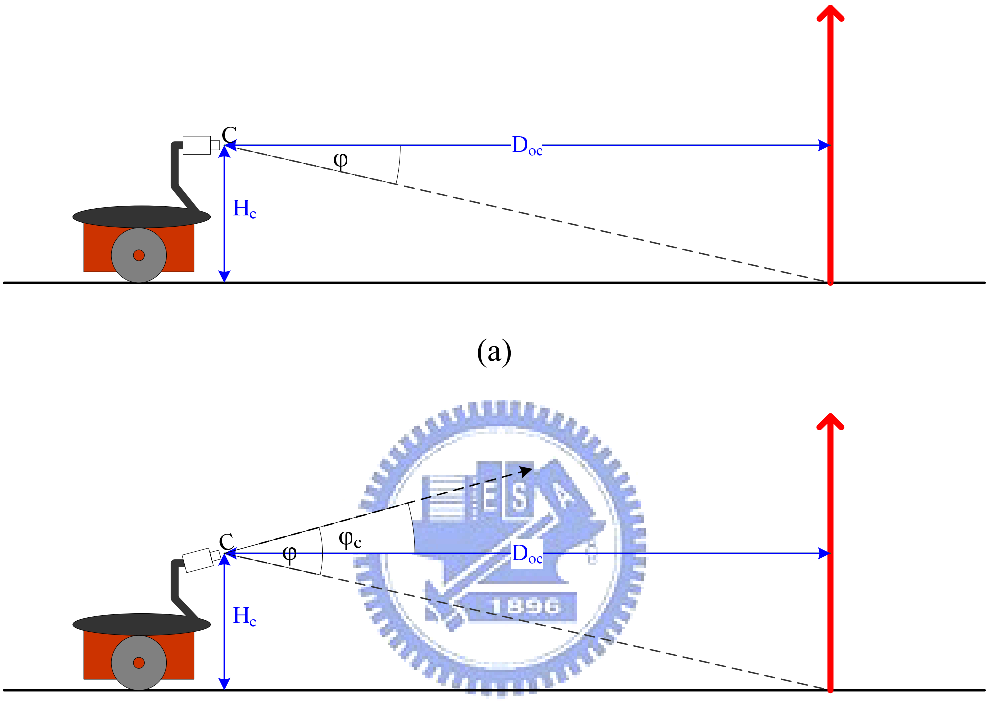
Step 1. Transform (u, v) of P into its (θ, φ), the longitude and the latitude in the SCS, by the proposed mapping method described in Section 3.3.2.

Step 2. Compute D_{oc} by the following equation:

$$\tan(\varphi - \varphi_c) = \frac{H_c}{D_{OC}}$$

where φ_c is the tilt angle of the camera, as shown in Figure 3.14.

Since we know the distance between the vehicle and the object by applying the above algorithm, what we have to do now is to turn the direction of the vehicle toward the object and moves forward. The way we propose to find the angle the vehicle has to turn will be introduced in the next section.



(a)

(b)

Figure 3.14 The distance between the object and the vehicle. (a) The camera has no tilting, i.e. $\varphi_c = 0$. (b) The camera has a tilt angle of φ_c.

3.4.2 Angle Transformation between Coordinate Systems

As shown in Figure 3.15, we can know the point of the object in the image and get the distance from the vehicle to the object according to the above algorithms. However, the rotation center of the camera is different from the one of the vehicle. Thus, the longitude value θ of the object in the ICS is different from the directional angle θ_v, the angle that the vehicle has to turn to aim at the object, in the PCS. The transformation from (u, v) in the ICS to θ_v in the PCS is described in the following algorithm.

Algorithm 3.5 *The angular transformation from the ICS to the PCS.*

Input: The distance between the rotation center of the camera and the vehicle, and the ground contact point (u, v) of the object in the ICS.

Output: The directional angle θ_v of the object in the PCS

Steps:

Step 1. Transform (u, v) to (θ, φ), the longitude and latitude in the SCS, by the proposed mapping method in Section 3.3.2.

Step 2. Compute θ_v in the PCS by the following equation according to Figure 3.15:

$$\tan\theta_v = \frac{D_{oc}\cdot\sin\theta}{D_{oc}\cdot\cos\theta + D_{cv}}.$$

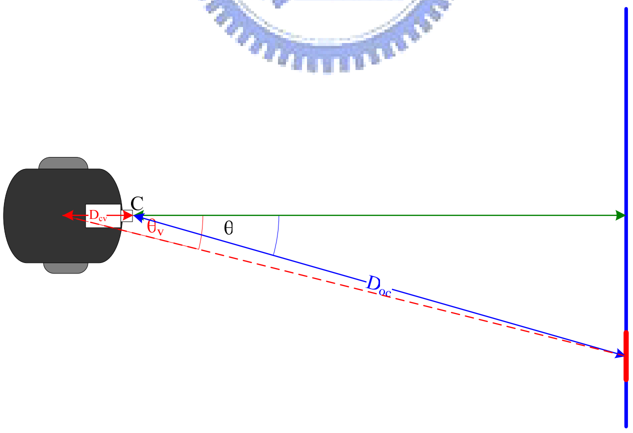


Figure 3.15 The rotation angle of the vehicle to obtain the tracking of the object.

From the above algorithm and the one in the last section, we can transform the position of the object in the image to the polar coordinate system with the distance and directional angles (D_{ic}, θ_v). By the transformation, the location of an object in the real word now is clear and can help us to conduct tracking of target objects.

Chapter 4
Human Detection by Image Analysis for Indoor Security Patrolling

4.1 Overview of Human Detection

There are many kinds of features and sensors to detect human beings. Since visual perception is the only sensing capability of the proposed system in this study, image analysis is one of the solutions to detect human beings. The face is an obvious characteristic of human beings. As the result, we propose a method to detect human faces by color and shape features in images. The method for face detection will be described in Section4.3. Sometimes, the limitation of camera resolution makes the acquired image unclear. A far distance from a person to the camera might cause difficulty in segmenting a clear human face region out of an image of the person. To redeem the limitation, we propose a blockwise frame difference method to extract moving objects in the image and decide if the moving object is similar to a human body. The motion detection method will be proposed in Section 1.4. Before all the details of the mentioned techniques are described, we will give a brief introduce to the proposed process in Section 4.2 first.

4.2 Proposed Process of Human Detection

The proposed process of human detection has two major parts: human face detection and human body detection. The features we adopt to detect a human face are color and shape. The color of the face undoubtedly is just the skin color, and the skin color has been studied intensively in recent years. In this study, we adopt an elliptic skin model to determine if the color of a pixel is skin color or not.

After getting all the skin color regions in an image, we have to recognize which one is similar to the shape of a human face. As the contour of a human face is roughly elliptic in shape, we propose a method for matching each skin color region with an elliptic shape mask. On the other hand, to avoid erroneously recognizing an elliptic non-face region as a face from skin color regions, we make a double check by motion detection.

If nothing is detected by the face detection process, we decide that a person might exist at a far distance. Then, we try to confirm the decision further by detecting the existence of a human body using moving regions in the image, which can be obtained by an additional process of frame differencing. The technique of frame differencing does not work finding the case of having a moving region in a changing background. We propose therefore a blockwise frame differencing technique to detect moving regions. After performing this technique, we can get moving regions in an image and detect any human body by applying a shape recognition technique to these moving regions.

The system will stop the human detection process and start a human tracking process as long as a face is detected in an image. The process of human tracking will be described in the next chapter. The major steps of the proposed process of human detection are presented as follows.

Step 1. Capture an image.

Step 2. Apply region segmentation by skin color identification and motion detection by blockwise frame differencing to extract motion regions.

Step 3. Fit each extracted skin region with an ellipse to detect a possible human face.

Step 4. Apply human body detection by applying shape recognition to extracted motion regions.

The proposed process of human detection is illustrated in Figure 4.1.

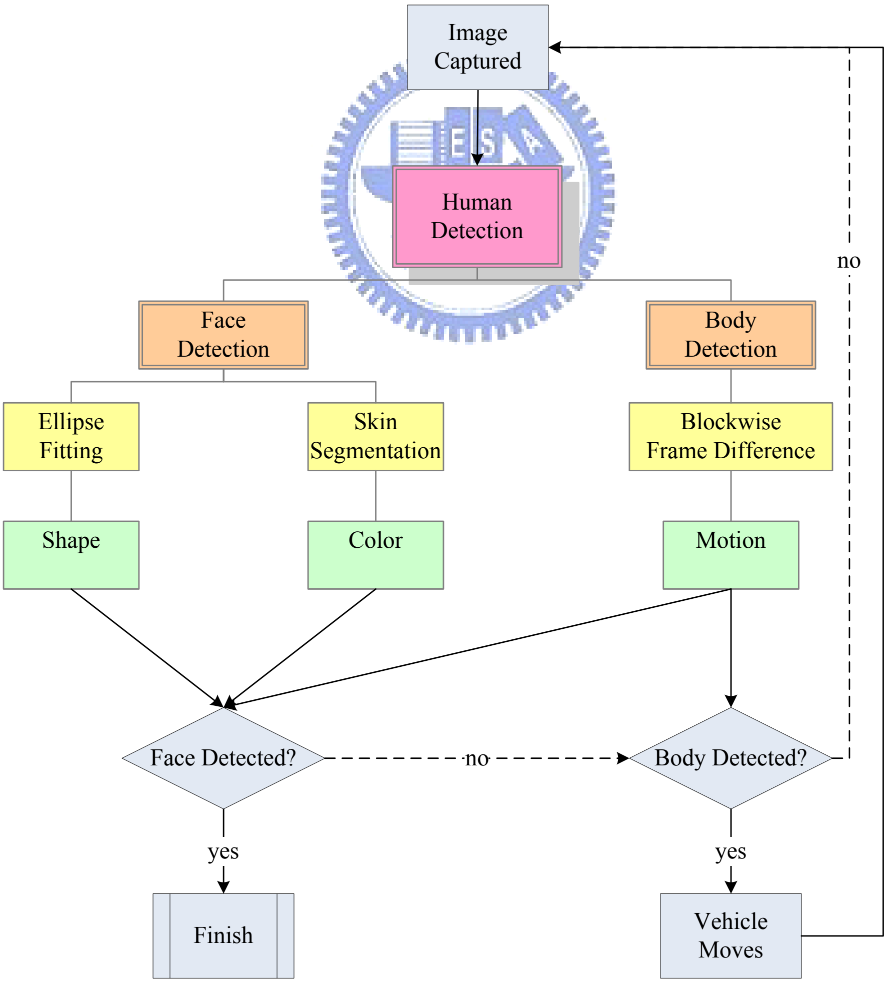


Figure 4.1 The proposed process of human detection.

4.3 Human Detection by Faces

In order to detect faces in images, we have to choose features to define a face. The features we used in this study are color and shape, as mentioned previously. The rough sketch of a face can be represented by the shape of an ellipse with the skin color. Thus, we detect a human face in the image by searching a skin-colored ellipse.

More specifically, recognizing a skin-colored ellipse in an image needs two tasks: giving the definition of skin color and conducting pattern recognition of ellipses. With a captured image, we first segment out any skin region and then fit shapes to the regions. If a skin color region is close to an ellipse in shape, it is decided that a face is detected. In Section 4.3.1, we will introduce the proposed method of classification of skin color. And in Section 4.3.2, we will describe the proposed method for ellipse shape recognition.

4.3.1 Skin Region Segmentation Using Color Classification

Skin region segmentation is commonly used for face detection. Determining a color pixel is of a skin color or not is the goal. Before defining the classifier for skin color, the choice of color representations is important, which affects the complexity of the classifier. In this section, we describe the color space and the classification algorithm proposed in this study.

4.3.1.1 YC_bC_r Color Space

Many common color models are used in the field of computer vision, for examples, RGB, HIS, HSV, YC_bC_r, CMY, CIE, etc. Each one of the color models has its own characteristics and is applicable to a specific set of applications. In the application of skin region segmentation, classifiers for different color models are proposed in many research works.

In this study, we choose the YC_bC_r to be the color space for detecting skin color in images. According to Chai and Bouzerdoum [20], the distribution of skin color in YC_bC_r color space is concentrative and the distribution of the skin colors of different human races are similar. As the result, transforming images in the RGB color space into ones in the YC_bC_r color space can reduce the complexity of skin-color pixel classification.

In the YC_bC_r color space, Y represents the luminance, C_b represents the chrominance of blueness, and C_r represents the chrominance of redness. Y is coded from 16 to 235, where code 16 is black and 235 is white. And C_b and C_r range from 16 to 240. RGB values can be transformed into the YC_bC_r color space by (4.1) below:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.533 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -84.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.1}$$

Given the input RGB values which are within the range of [0, 1], the output values will be within the range of [16, 235] for Y and [16,240] for C_b and C_r.

Figure 4.2 shows the YC_bC_r color model with the value of Y being 126. Because the transformation from the RGB space to the YC_bC_r is linear but not one-to-one, some value sets of C_b and C_r are not meaningful when the value of Y is 126. Some YC_bC_r color models with different Y values are shown in Figure 4.3. When Y is 16 which is the darkest luminance, C_b and C_r can only have the value of 128. Likewise, when Y is 235 which is the brightest luminance, C_b and C_r also only have the value of 128. Figure 4.4 shows a 3D YC_bC_r color model [23][24].

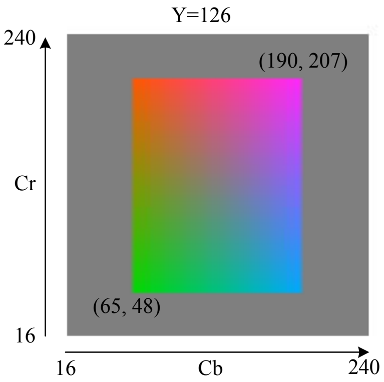


Figure 4.2 YC_bC_r color model with Y = 126

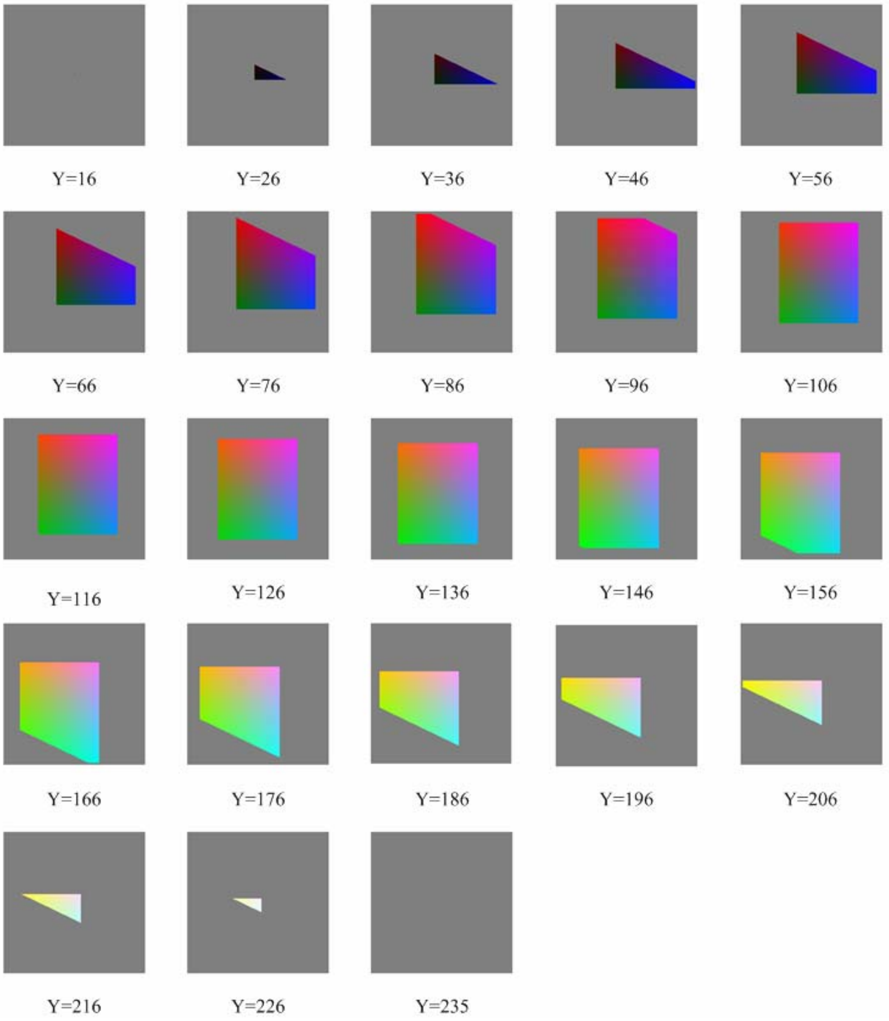


Figure 4.3 YC_bC_r color models with different Y values.

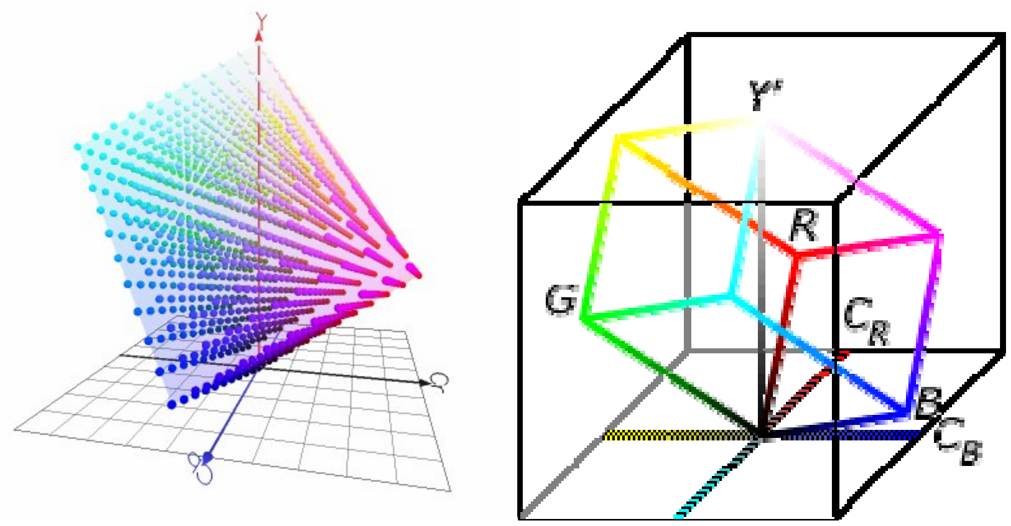


Figure 4.4 3D YC_bC_r color model in [23][24].

4.3.1.2 Adopted Skin Color Model

Previous studies have found that pixels belonging to skin regions exhibit similar C_b and C_r values [20][21]. Chai and Mgan [21] used a fixed-range skin color map in the C_b-C_r plane for face segmentation, and the range of C_b is between 77 and 127 and the range of C_r is between 133 and 173. And the region of the skin color is shown to be a rectangle. However, when we observe the distribution of skin color in the C_b-C_r plane, it is found more similar to an oblique ellipse, as shown in Figure 4.5. In the study by Lee and Yoo [22], a new statistical color model for skin detection with elliptical boundaries was suggested. Thus, we define an oblique ellipse in the C_b-C_r plane to be the skin color model in this study, and the parameters of the elliptic skin model are adjusted by experiments. The center of the elliptic model is taken to have the values 103 for C_b and 158 for C_r. And the angle of rotation is set to be 145 degrees, and the lengths of major and minor axes are set to be 25.39 and 14.03 respectively.

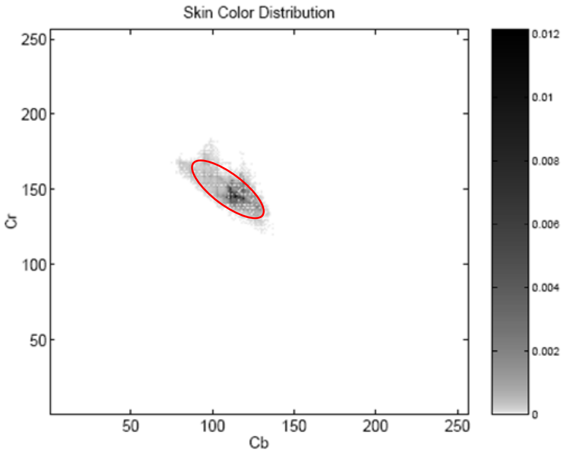


Figure 4.5 Distribution of conditional probability density function of skin color in C_b-C_r plane [20].

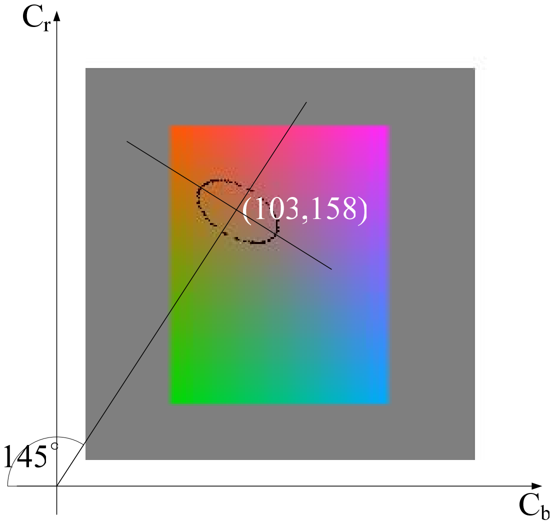


Figure 4.6 The elliptic skin model used in this study.

4.3.2 Detection of Human Face by Ellipse Shape Fitting

Since the shape of a human face is close to an ellipse, we propose in this study a pattern recognition method for ellipse shape detection to distinguish face regions from other skin regions segmented out by the skin color classification mentioned in Section 4.3.1. More specifically, after segmenting the skin color regions out of an image, we determine if the region is similar to an ellipse. If so, we take it to be a human face. The method to decide whether a region is elliptic in shape is described as an algorithm in the following.

Algorithm 4.1 *Face detection by pattern recognition of ellipses*.

Input: A skin region set $R = \left\{R_1, R_{2,} ..., R_n\right\}$ as shown in Figure 4.7(b). The width of a region R_i in R is denoted by w_i, and the height of R_i by h_i. The boundaries of the region R_i are denoted as *left*$_i$, *right*$_i$, *top*$_i$, *buttom*$_i$. And the number of pixels in region R_i is denoted by p_i.

Output: A face region R_{face}.

Steps:

Step 1. Get a new skin region R' by

$$R' = \{R_i \;\middle|\; \frac{w_i}{h_i} \le 1 \text{ and } p_i > c, \; \forall i = 1, 2, ..., n\}$$

where c is a pre-selected constant.

Step 2. Make a rectangular mask *rectangle*$_i$ for R_i with width w_i and height 1.2×w_i, and an elliptic mask *ellipse*$_i$ for R_i with its major axis length being w_i and its minor axis length being $1.2 \times w_i$, as shown in Figure 4.7(c).

Step 3. To fit each region in R' with an ellipse shape, compute the number, *in*$_i$, of the pixels of region R_i within *ellipse*$_i$. Additionally, compute the number, *out*$_i$, of the pixels of region R_i within *rectangle*$_i$ and without *ellipse*$_i$. That is, compute

$$in_i = R_i \cap ellipse_i,\ \forall R_i \in R'$$
$$out_i = R_i \cap (rectangle_i - ellipse_i),\ \forall R_i \in R'$$

Step 4. Calculate a *score* S_i for each skin region R_i in R' by

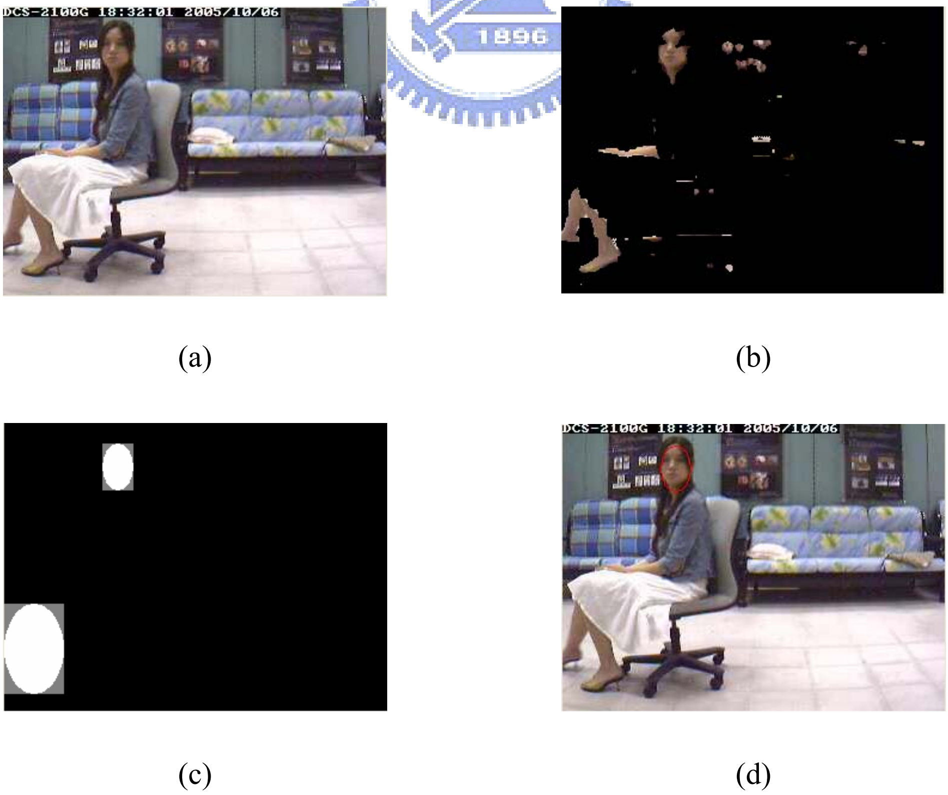
$$S_i = in_i - out_i\,.$$

Step 5. Normalize S_i into the range [0, 1] by

$$S_i' = \frac{S_i}{\text{area of } ellipse_i}\,.$$

Step 6. Decide region R_i to be a face region R_{face} if the score S_i' of R_i is highest among all regions in R and S_i' is higher than a threshold h in the range of [0, 1] which is defined in advance.

By the ellipse shape fitting as described above, we can detect the face region in images, as shown by the example in Figure 4.7.



(a) (b)

(c) (d)

Figure 4.7 The detection of human face by ellipse shape fitting. (a) Input image. (b) Skin Segmentation. (c) Rectangular and elliptic mask. (d) Detected face region.

4.4 Human Body Detection by Motion Analysis

Two kinds of misjudgments happen in the human detection work using the proposed human face detection method mentioned in the last section. One is recognizing a face-like object to be a human face in the image, and the other is detecting nothing when a person does exist at a distance from the vehicle. To avoid the first kind of mistake, we need an advanced feature to confirm if the detected face region is a human face or not. Also, to avoid the second type of mistake, we have to detect humans by other features, not only the feature of face. In this study, we use techniques of motion detection and human body recognition to reduce the effects of these drawbacks and so increase the reliability of the proposed system.

In a fixed camera system, the moving parts of the scene can be detected by frame differencing with fixed backgrounds learned in advanced. Unfortunately, this method is not working in our system because the background in the image is always changing with the camera on the moving vehicle and robot arm. We thus propose in this study a method of frame differencing for use by our vehicle system, which is presented in Section 4.4.1. In Section 4.4.2, we will describe the method of human body recognition by motion detection.

4.4.1 Motion Detection by Shift Tolerance Blockwise Frame Differencing

First, we define some terns for use in the proposed method.

(1) *Current image*: The image captured from the camera at the current moment or equivalently in the current navigation cycle.

(2) *Reference image*: The image captured from the camera at the last moment.

(3) *Block*: A block consists of a square region of pixels, which is the unit of the image.

(4) *Searching window*: A searching window consists of a square region of pixels, whose size is larger than the size of a block.

Subtracting the current image from the reference image block by block is the basic idea of blockwise frame difference. If the difference between the target block in the current image and the candidate block at the same position in the reference image is below some threshold, then it is may be considered that no motion has taken place, i.e. the target block is *stationary*. If it is not, we will find the best match block for the target block within the searching window in the reference image. If the best match block is below the threshold, we say that the target block is *stationary*; otherwise, *moving*. Repeating these steps for each block in the current image, we can get all the moving parts in the current image. The detail is stated in the following algorithm.

Algorithm 4.2 Shift tolerance blockwise frame differencing.

Input: current image I_c, reference image I_r, block size $s \times s$, and the size of a searching range w, which makes the size of the searching window being $(2w+s)\times(2w+s)$.

Output: a difference image I_d.

Steps:

Step 1. Segment I_c into a block set

$$B_c = \{b_{11}, b_{12}, \ldots, b_{1m}, b_{21}, b_{22}, \ldots, b_{2m}, \ldots, b_{n1}, b_{n2}, \ldots, b_{nm}\},$$

where the size of B is $m \times n$ as shown in Figure 4.8. Also, segment I_d into a block set B_d in the same way.

Step 2. Define the range of the searching window to be $(2w+s)\times(2w+s)$, as shown in Figure 4.9. Subtract the target block b_{ij} from the candidate block at the same position in the reference image. If the difference is below the threshold t, regard the target block b_{ij} as *stationary* and go to Step 5. Otherwise, go to Step 3.

Step 3. Find the best match block of the target block b_{ij} within the searching window in the reference image by subtracting the target block b_{ij} from each of the blocks within the searching window, as shown in Figure 4.10.

Step 4. If the difference between the target block b_{ij} and the best match block is below the threshold t, regard the block b_{ij} as *stationary*; else, *moving*.

Step 5. Repeat Step 2 for each block in B_c to decide the state, *stationary* or *moving*, of it.

Step 6. Get a complete frame difference image I_d by filling the *moving* blocks with white color and the *stationary* blocks with black color, as shown in Figure 4.11.

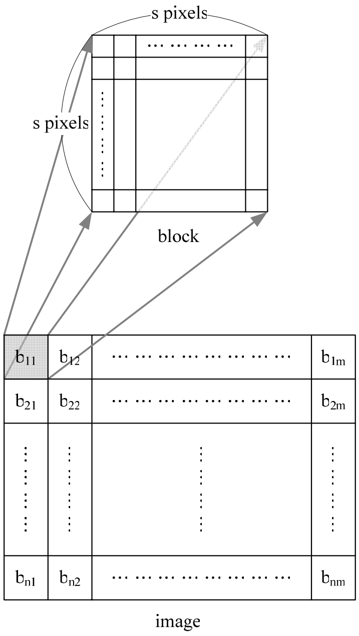


Figure 4.8 The image is segmented into blocks.

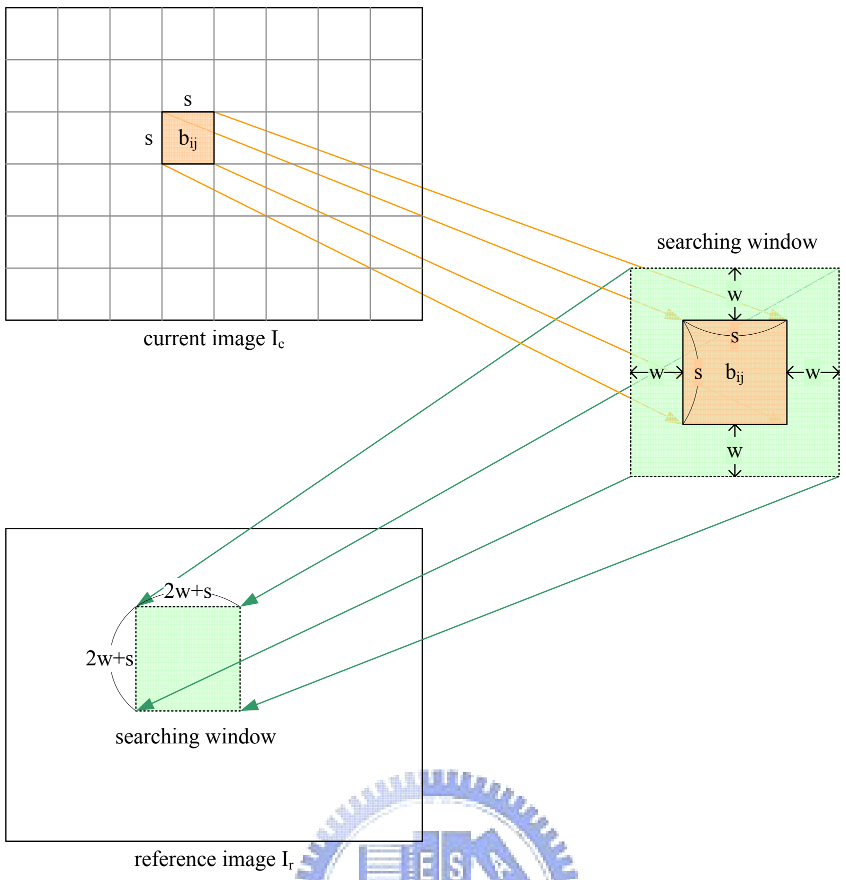


Figure 4.9 The searching window.

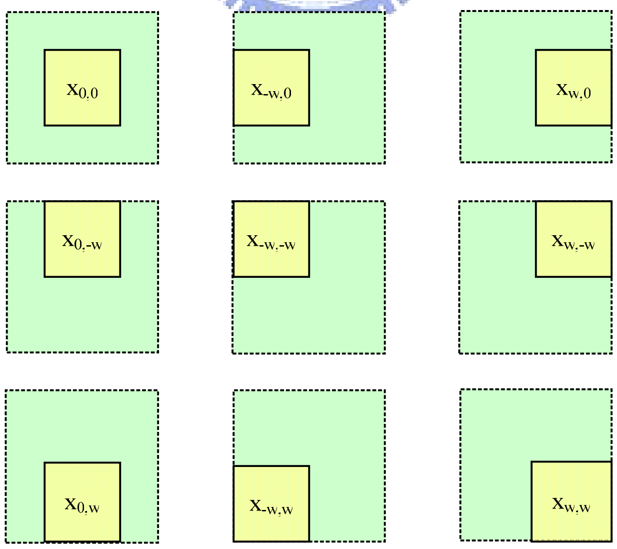


Figure 4.10 The blocks within in the searching window in I_r.

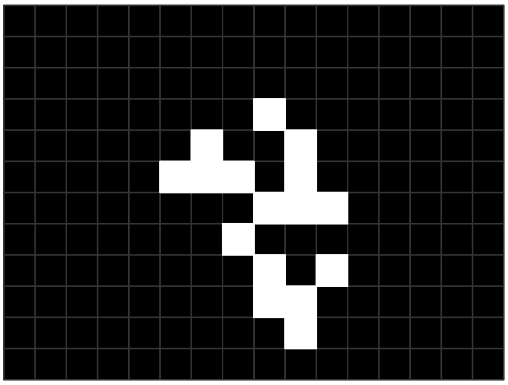


Figure 4.11 An example of blockwise frame difference images.

4.4.2 Human Body Detection

By the blockwise frame differencing result obtained by the method described in the last section, we can get a difference image from every two sequential image frames. The difference image shows the moving regions at the current moment. Since we compute the moving regions by *blockwise* frame differencing instead of a pixelwise operation, the regions do not appear to have a complete shape of a human body. Moreover, the shape of a human body is irregular in shape, so it is impossible to detect a human body by fitting a detected region with a certain well-known shape.

Furthermore, when the vehicle is moving in an open space, if the system detects some moving regions, we can assume that there is something moving in the filed of view of the camera. Since the shape of a human body is complicated and is hard to define, we use only the ratio of the width to the length of a moving region as a feature for human body detection in this study. This feature of a normal person is defined to be the ratio of the shoulder width to the body height. And a reference range of this ratio is around 1/4 as shown by the *Vitruvian Man* painted by Leonardo da Vinci. However, if two sequential images both include the same person but at different positions, the difference of these two images cannot form a complete shape of a human body. If the positions in these two images are close, the result of the difference image of these two images might show a moving region which is too “thin,” and if the positions are far away, the difference image might show a moving region which is too “thick.” Thus, we widen the range from 1/8 to 1 to consider the situation of overlapping of body regions in consecutive images. The following algorithm presents the human body detection method by the feature of body proportion discussed above.

Algorithm 4.3 *Human body detection by moving region proportion.*

Input: A moving region set $R = \left\{R_1, R_{2,} ..., R_n\right\}$ as shown in Figure 4.7(b). The width of a region R_i in R is denoted by w_i, and the height of R_i by h_i.

Output: A human body region R_{body}.

Steps:

Step 1. Get a new moving region R' by

$$R' = \{R_i \left| \frac{1}{5} \le \frac{w_i}{h_i} \le 1 \text{ and } p_i > c, \ \forall i = 1, 2, ..., n\right\}$$

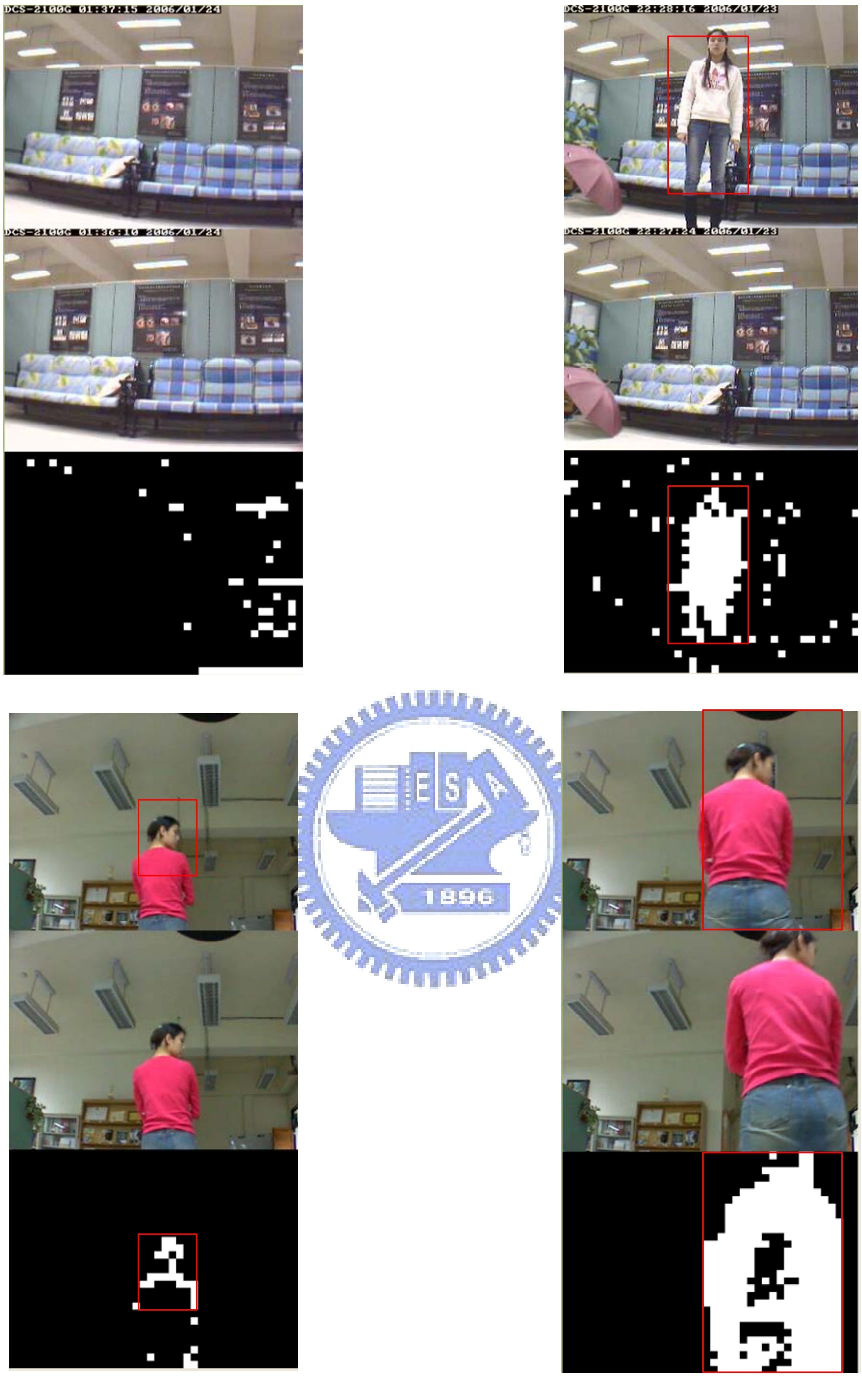
where c is a pre-selected constant.

Step 2. If $R' = \varnothing$, it means no human body is detected in the moving regions. Else, if $|\mathrm{R}'| = 1$ where $R' = \left\{R_i\right\}$, we decide the moving region R_i is the human body region: $R_{body} = R_i$.

Step 3. Else, if $|R'| \ge 2$, we have to choose which region in R' is a human body. We decide the moving region R_i to be R_{body} if the product of w_i and h_i is the maximum among all products of w_j and h_j, where $R_j \in R'$.

4.4.3 Some Experimental Results

In this section, we show some experimental results of human body detection in Figure 4.12. Figure 4.12(a) shows a case of both sequential images with no human beings, and the difference image showing some spotty moving regions. As shown in Figure 4.12(b), the difference image shows the complete shape of a person. However, if two sequential images both include the same person but at different positions, the difference of these two images cannot form a complete shape of a human body. If the positions in these two images are close, the result of the difference image of these two images might show the contours of a moving person, as shown in Figure 4.12(c). And if the positions are far away, the difference image might show a union region of the moving person, as shown in Figure 4.12(d).


DCS-2100G 01:37:15 2006/01/24
DCS-2100G 01:36:10 2006/01/24
DCS-2100G 22:28:16 2006/01/23
DCS-2100G 22:27:24 2006/01/23

(c) (d)

Figure 4.12 The order of the images: the current image, the reference image and the difference image using the proposed blockwise frame difference. (a) The person regions are close. (b) The person regions are far. (c) The person only exists in one of the images (d) No person exists in both images.

Chapter 5
Human Tracking in Indoor Environment

5.1 Basic Idea of Human Tracking

After a human face is detected by the proposed human detection method mentioned in the previous chapter, the system will extract the color of the person's clothes and save the image part of the clothes in the PC. The vehicle can then track the target person by continuous detection of the clothes. In this chapter, we will describe the entire process of human tracking in detail. In Section 5.2, we first present the process of human tracking step by step. The vehicle navigates according to the position of the target person and conducts face detection at the same time. The system will also compute the distance of the target person using the face detected.

In Section 5.3, a method for extraction of colors of human clothes is proposed. According to the position of the human face which is detected before, we compute the position of the human body and extract the color of the clothes by region growing and save the image part of the clothes. Then, we will describe the detail of human tracking by clothes in Section 5.4. We use the intersection of the cloth images to compute the position in the image of the target person. Two applications, stranger tracking and person following, of the proposed human tracking method are stated in Section 5.5.

5.2 Proposed Process of Human Tracking

In the process of human tracking, the vehicle tracks the target person by detecting the clothes of the target person consecutively. In the previous chapter, we describe how to estimate the location of the target person's face in the image. Then, the system will extract the cloth region of the person to facilitate human tracking. The idea behind the tracking method is to make the target person always appear at the center of the image, which means that the head of the vehicle always aims at the target person and moves forward. After turning the head of the vehicle, the vehicle will move to the person for a constant distance. Figure 5.1 shows a cycle of the human tracking process.

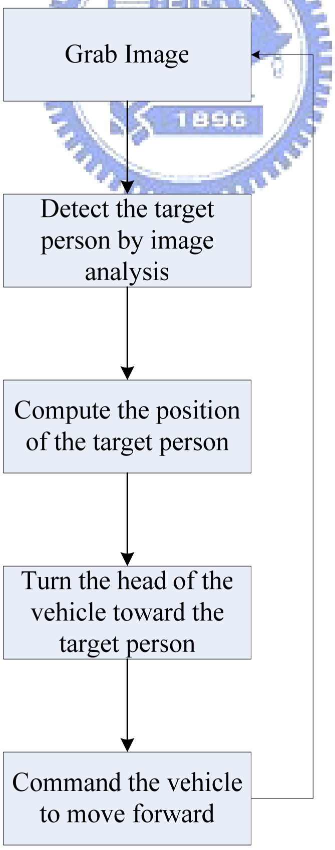

Grab Image
Detect the target person by image analysis
Compute the position of the target person
Turn the head of the vehicle toward the target person
Command the vehicle to move forward

Figure 5.1 A cycle of the human tracking process.

The proposed process of human tracking is described in the following algorithm.

Algorithm 5.1 *Process of human tracking.*

Input: The region of the detected human face R_{face}.

Output: The commands for the vehicle.

Steps:

Step 1. Save the image I_{cloth} of the clothes of the target person by a region growing technique described in the next section.

Step 2. Capture an image $I_{current}$ with the camera.

Step 3. Set the missing counter C_m to be 0.

Step 4. Obtain the difference image of I_{cloth} and $I_{current}$ to compute the position of the target person. If the target person is checked to be missing, increase C_m by 1. Else, reset $C_m = 0$.

Step 5. Command the vehicle to turn its head to aim at the target person and move forward.

Step 6. Check the number of the missing counter $\mathrm{C_m}$: if it is below the threshold, go back to Step 2; else, end the process of human tracking. Because the missing counter represents the number of times that the target person is missing, give a threshold t, if C_m is larger than t, then we decide that the target person is lost and command the vehicle to stop tracking.

5.3 Extraction of Colors of Human Clothes

Since we already know the face region, we can infer that the body region is below the face region. We choose a point which belongs to the body region to be the start point for region growing, and give a square boundary for region growing. By region growing, we can get the image part and the location of the clothes. After cloth extraction, the vehicle can track the human by the clothes region in the image sequence.

According to the detected face region, we know the width and height of the face region in images. To infer the body region of the person from the face region, we reference the drawing of Vitruvian Man by Leonardo da Vinci, as shown in Figure 5.2. According to Leonardo's notes in the accompanying text, it was made as a study of the proportions of the human body as described in a treatise by Ancient Rome Architect Vitruvius, who wrote in Vitruvius De Architectura 3.1.3 that: "*for measuring from the feet to the crown of the head, and then across the arms fully extended, we find the latter measure equal to the former; so that lines at right angles to each other, enclosing the figure, will form a square.*" According to the body proportions defined by Vitruvius, the distance from the hairline to the bottom of the chin is one-tenth of a man's height and the maximum width of the shoulders is a quarter of a man's height. As shown in Figure 5.3, by the body proportions, if we know the region of the human face, we can infer the width of the human shoulder and the distance from the face to the body. Since the image captured with the camera is a geometric ratio projection, after we detect the face region in the image, we can approximately infer the region of his/her clothes. We use the center of the clothes region to be the start point for region growing of the clothes region. And the red square is the boundary for region growing. The detail is presented in the following algorithm.

Algorithm 5.2 *Computing start point and boundary square for region growing of clothes.*

Input: The detected face region R_{face} in the image, where R_{face} has two pairs of the coordinates (u, v) in the ICS: (f_{left}, f_{top}) and (f_{right}, f_{bottom}) which represent the boundary box of R_{face}.

Output: The start point $S(u_s, v_s)$ and the boundary box B described by (B_{left}, B_{top}) and (B_{right}, B_{bottom}) for region growing.

Steps:

Step 1. Compute the height of R_{face}, by $height_{face} = f_{bottom} - f_{top}$.

Step 2. Compute $width_{shoulder} = \frac{2}{5} height_{face}$ based on the facts:

$$\begin{cases} height_{face} = \frac{1}{10} \cdot height_{human} \\ width_{shoulder} = \frac{1}{4} \cdot height_{human} \end{cases}.$$

Let $width_B = width_{shoulder} = \frac{2}{5} height_{face}$.

Step 3. Define the boundary box B for region growing as a square with

$$width_B = height_B = \frac{2}{5} height_{face}.$$

Step 4. Define the start point S at (u_s, v_s) to be the center of the width of the face and the bottom of the face region to be the top of the boundary region where

$$u_s = \frac{f_{left} + f_{right}}{2},$$

$$v_s = f_{bottom} + \frac{1}{2} height_B \,.$$

Step 5. Since *S* is the center of *B*, set

$$\begin{cases} B_{left} = u_s - \frac{1}{2} width_B; \\ B_{right} = B_{left} + width_B; \\ B_{top} = v_s - \frac{1}{2} height_B; \\ B_{bottom} = B_{top} + height_B. \end{cases}$$

In the above algorithm, we have the coordinates of the start point *S* and the boundary box *B*, so we can get the cloth image I_{cloth} of the person by region growing. Also we know the clothes region R_{cloth} in the image where the face is detected. R_{cloth} has two pairs of the coordinates (*u*, *v*) in the ICS: (c_{left}, c_{top}) and (c_{right}, c_{bottom}) which represent the boundary box of R_{cloth}.

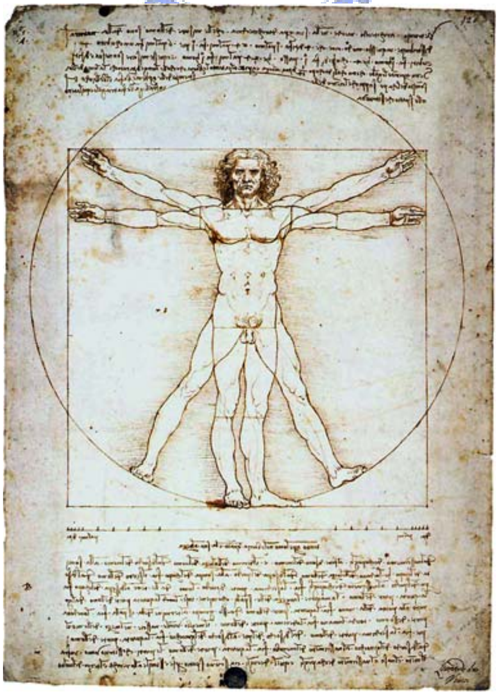


Figure 5.2 The drawing of Vitruvian Man by Leonardo da Vinci.

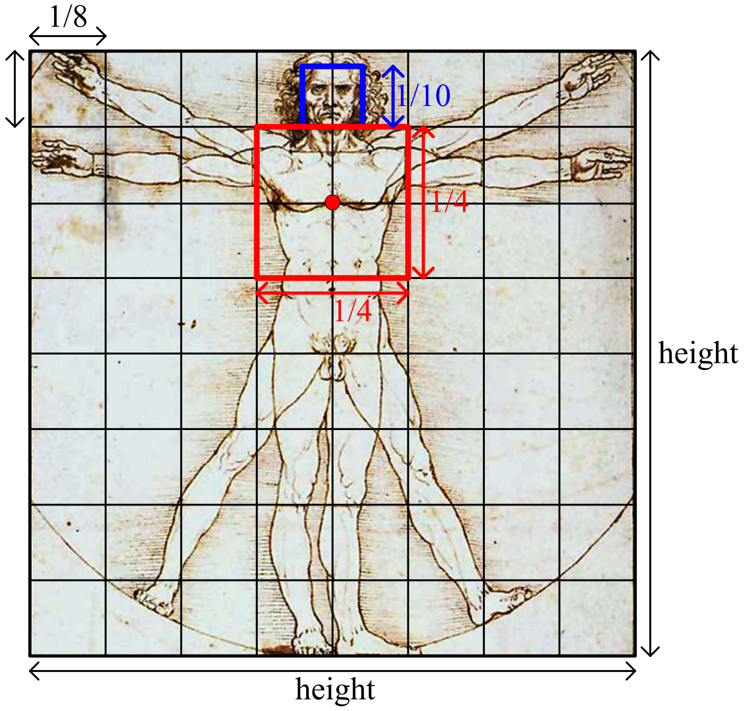


Figure 5.3 The body proportion according to Vitruvius.

5.4 Human Tracking by Motion Analysis of Human Clothes

In this section, we introduce the method for human tracking by human clothes. To track a person, besides knowing the position of the target at the moment, it has to predict his motion at the same time because of the delay of the vehicle for moving. To predict the motion of the target person, we have to record the motion of the target person as the basis for prediction. The recording format will be described in Section 5.4.1. In Section 5.4.2, we will describe the method for motion detection by cloth region intersection.

5.4.1 Recording of Human Motion

Because the vehicle tries to make the target person always appear at the center of the image, the movement of the target person from the center to the present location in the image can be seen as the *relative* movement of the target person and the vehicle. Therefore, we record the relative movement in the last cycle to be a reference for predicting the movement of the target person in this cycle. As we know the position, ($u_{current}$, $v_{current}$), of the person and the image center, (u_{center}, v_{center}), in the ICS, we can know the relative movement (u_{move}, v_{move}) of the target person by

$$\begin{cases} u_{move} = u_{current} - u_{center}; \\ v_{move} = v_{current} - v_{center}. \end{cases}.$$

5.4.2 Motion Detection by Cloth Region Intersection

To detect the location of the target person in this cycle by clothes, we use a cloth intersection region to predict the direction of the target person. The method only computes the directional variation of the target person. The detail of the proposed clothes region intersection is described in the following algorithm.

Algorithm 5.3 *Cloth region intersection.*

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

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

Steps:

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

Step 2. Subtract $I_{current}$ at $R_{initial}$ by I_{cloth} pixel by pixel, and get a new image $I_{intersect}$ which is the intersection of the two images at the region $\mathrm{R}_{\mathrm{intersect}}$.

Step 3. Grow a new clothes region $R_{current}$ by region growing that the starting pixels are randomly chosen from $I_{intersect}$.

Step 4. Let $R_{initial} = R_{current}$, and repeat the steps.

5.5 Applications to Stranger Tracking and Person Following

5.5.1 Stranger Tracking

By combining the process of human detection in the previous chapters with the process of human tracking in this chapter, it can achieve the goal of tracking intruding persons in indoor environment for security patrolling.

After we detected a human face in the image, we extract the image of the clothes by the method mentioned in Section 5.3. And after extracting the clothes region, the vehicle is commanded to track the person who wears the same color of the cloth image until the target person disappears in the field of view. In that case, the vehicle will return to the detection mode which was described in Chapter 4.

5.5.2 Person Following

With a pre-learning strategy of the target person, the system can conduct a work of following a specified person, which we call *person following*. Unlike stranger tracking just described, the system learns the clothes of the target person by manual. The user decides the start point and the boundary box in the image for region growing of the clothes. After learning the image part of the person’s clothes, the system will enter the human tracking process mentioned previously in this chapter.

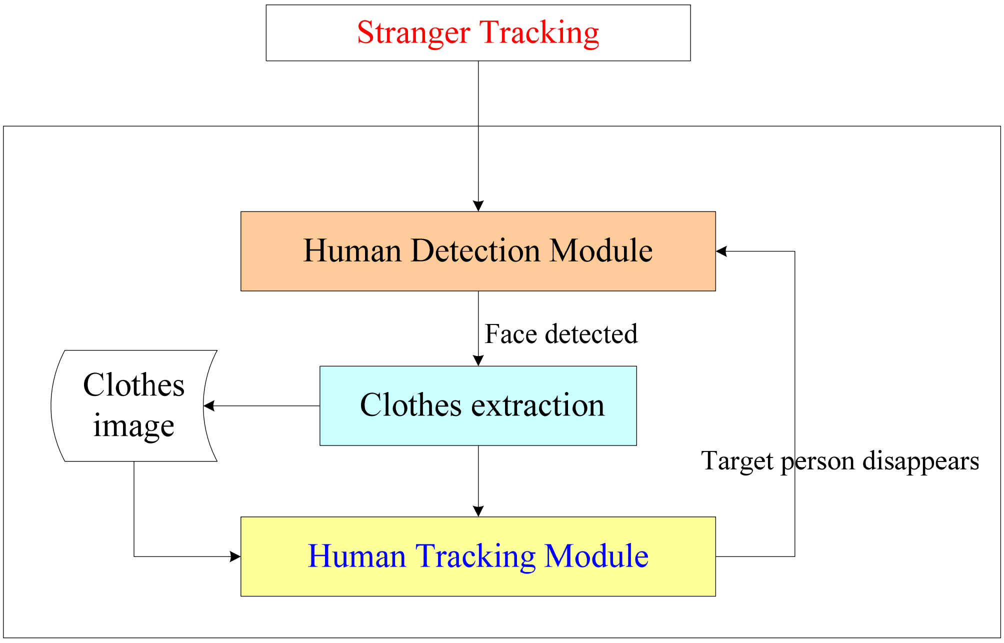


Figure 5.4 The application for stranger tracking.

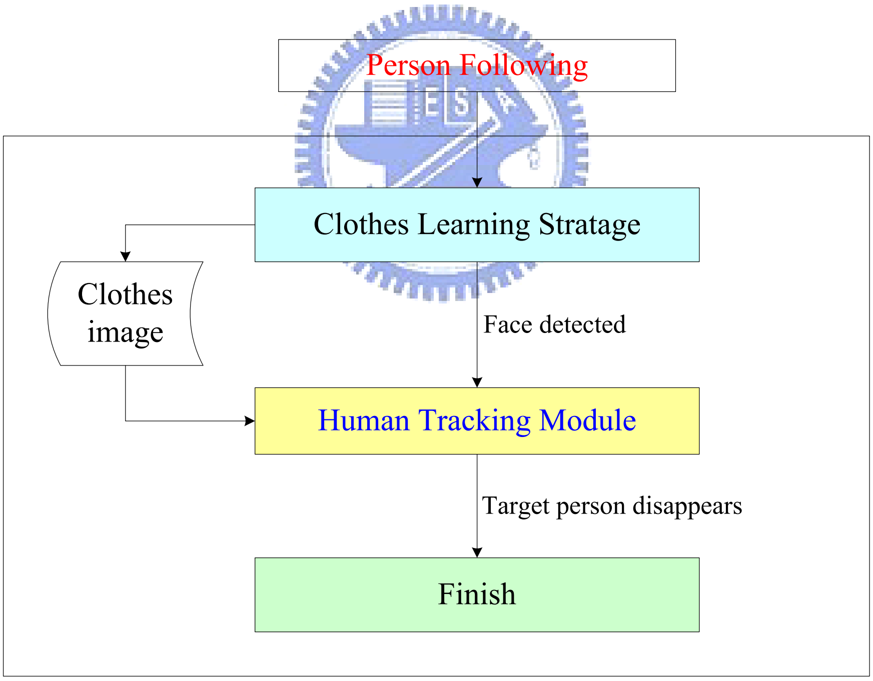
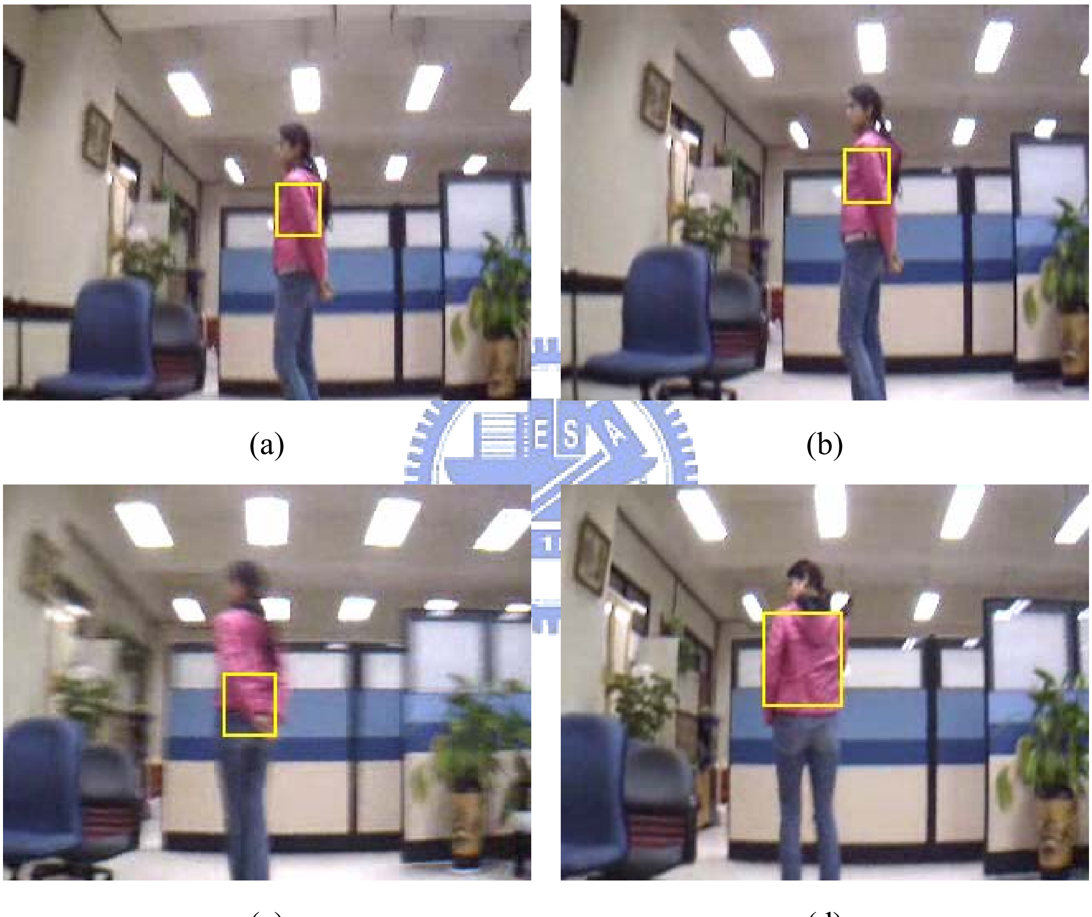


Figure 5.5 The application for person following.

5.5.3 Experimental Results

In this section, we show some experimental results of human tracking in Figure . After the vehicle detected a human face in an image and extracted the clothes of the person, it started the process of human tracking. Figure shows the consecutive images which were taken by the camera equipped on the vehicle when the vehicle tracked the target person. The yellow box in the images represents the intersection of the cloth region of two consecutive images. According to the intersection region, the vehicle computes the present position of the person and turns its head to aim at the target person. In this way, the person appears at the center of the images all the time.



(a) (b)

(c) (d)

Figure 5.6 The consecutive images which were taken by the camera equipped on the vehicle when the vehicle tracked the target person. The order of the images is (a) through (j).

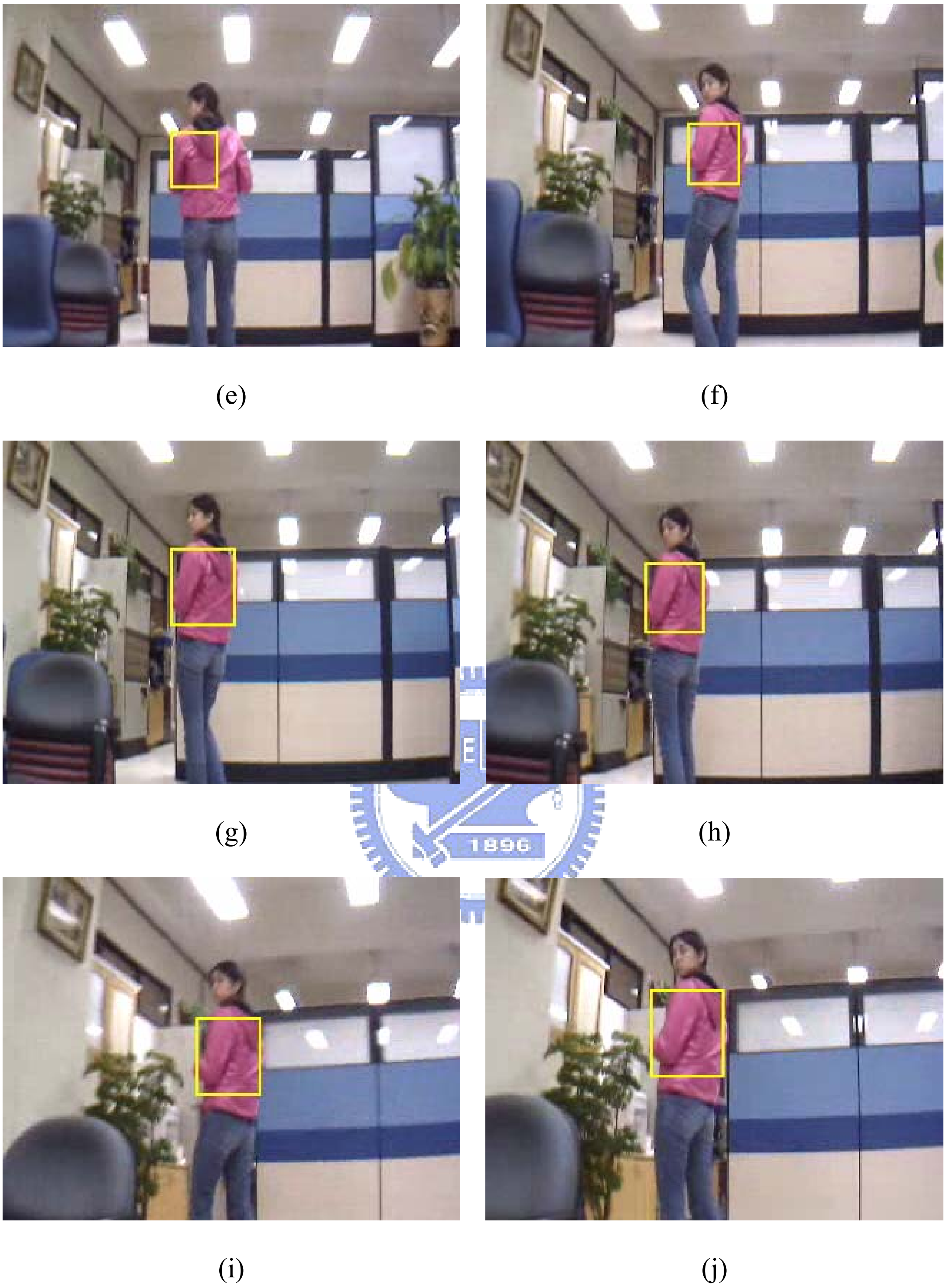

(e)
(f)
(g)
(h)
(i)
(j)

Figure 5.6 The consecutive images which were taken by the camera equipped on the vehicle when the vehicle tracked the target person. The order of the images is (a) through (j). (continued)

Chapter 6
Escape of the Vehicle from Strangers by Safe-Distance Keeping

6.1 Overview

The mobility property of the vehicle makes corners in a house viewable by the camera on the vehicle. On the other hand, this property also makes the risk that the vehicle might be stolen easily by an intruding person.

To avoid attacks from strangers, we design a mechanism for the vehicle to escape from strangers. There are two stages of the escape process, detection of dangerous situations and path planning for escape. We will state the principle of escape, which includes the definition for dangerous situations and the path planning strategy to escape in Section 6.2.

We regard the vehicle to be safe if nobody appears in a pre-defined range of distances from the vehicle. Accordingly, we need to compute the distance between the vehicle and the person which is detected in the image. We will describe how we accomplish this work in Section 6.3. To keep the stranger in the field of view of the camera, we need a method to adjust the orientation of the camera. The proposed method is presented in Section 6.4.

6.2 Principle of Escape

In this section, we describe the principle of escape for the vehicle. Before describing the method for escape of the vehicle, we have to decide what kind of situation in which the vehicle should escape. We define three states for the vehicle: *safe state*, *unsafe state* and *buffer state*. When the vehicle is in an *unsafe state*, we command the vehicle to escape. If the vehicle is in a *safe state*, the vehicle will be commanded to move forward the target person. Else, we command it to keep conducting the human detection and tracking process but do not move forward or backward when it is in a *buffer state*. To define these three states, we introduce the concept of *safe-distance*: if someone is too close to the vehicle, we infer that he might be going to attack the vehicle. Imagine a circle on the *xy*-plane in the VCS with the vehicle as the center and a pre-defined distance *r*, the *safe-distance*, as the radius. If a stranger is within the circle, we say the vehicle is in an *unsafe state*; else, the vehicle is in a *buffer state* or a *safe state*. It means that if the distance between the vehicle and a person is less than the *safe-distance* which is defined in advance, we regard the vehicle to be in an *unsafe state*. Otherwise, we give a buffer between the *safe* and *unsafe states*. If a person stays in the buffer area, the vehicle will neither move forward nor move backward. To keep the vehicle in a *safe state*, as soon as we detect a stranger transgressed the *safe-distance*, we command the vehicle to escape. The process is shown in Figure 6.1.

To compute the distance between the detected stranger and the vehicle, we conduct the face detection process which is described in the previous chapter. We assume the range of the height of a human face to be between 20cm and 25cm and the height of a human body to be between 50cm and 60cm. Because we have this assumption of the height of the face, the height of the human body and the pre-defined *safe-distance*, we can compute the distance between the person and the vehicle by angular mapping mentioned in Chapter 3. Then, we can determine whether the detected person transgresses the *safe-distance* or not.

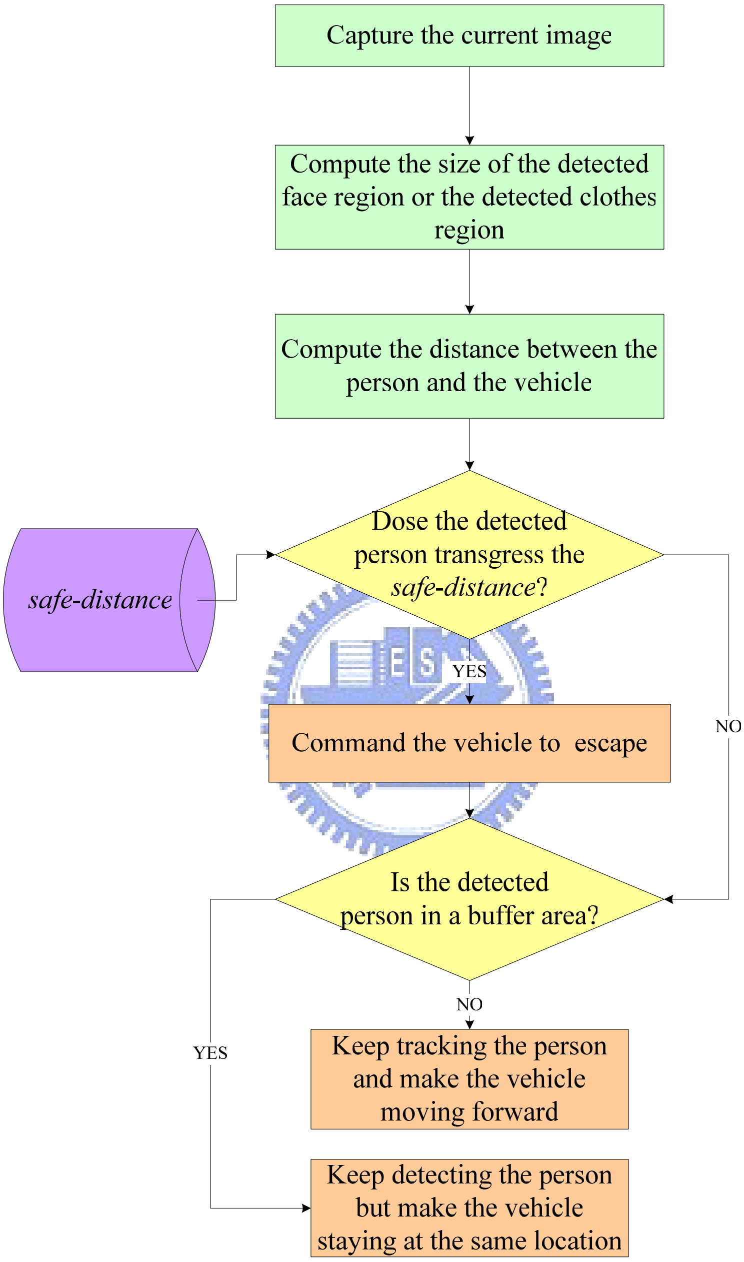

Capture the current image
Compute the size of the detected face region or the detected clothes region
Compute the distance between the person and the vehicle
Dose the detected person transgress the *safe-distance*?
safe-distance
YES
NO
Command the vehicle to escape
Is the detected person in a buffer area?
NO
YES
Keep tracking the person and make the vehicle moving forward
Keep detecting the person but make the vehicle staying at the same location

Figure 6.1 The process of escape for the vehicle.

If the vehicle is in an *unsafe state*, we have to command the vehicle to escape. However, there is only one camera equipped on the vehicle and the camera has to “keep an eye on” the detected stranger, so no more camera can be used to observe the environment to escape. Thus, we record the path the vehicle moved before as a solution to this problem. When the vehicle has to escape, it will be moved backward according to the recorded path. In an unsafe state for escape, the camera observes the stranger continually. When the target person is out of the safe distance, the vehicle will track the person again.

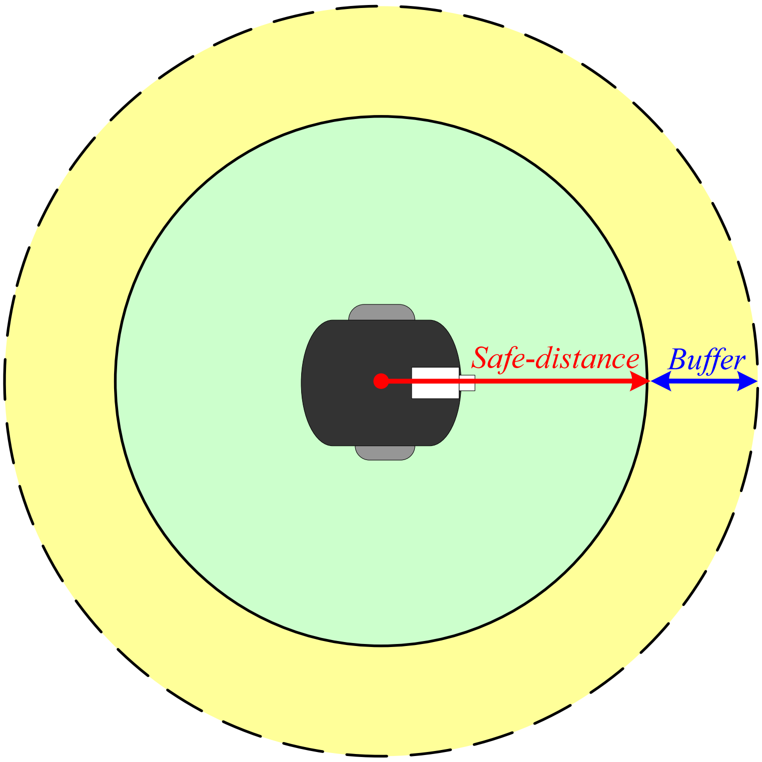
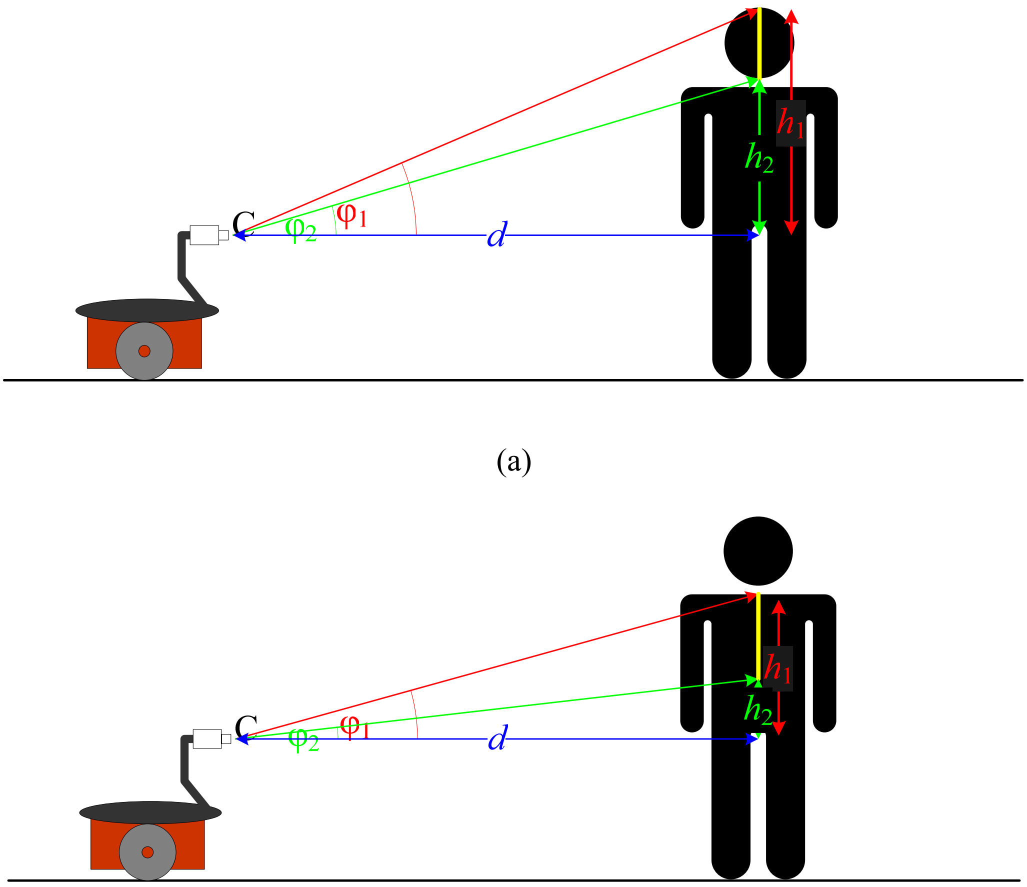


Figure 6.2 The *safe-distance* for the vehicle. The vehicle is in an *unsafe state* when a person detected within the green circle. Else if the person is in the yellow area, the vehicle is in a *buffer state*. Else, it is in a *safe state*.

6.3 Distance Computation from The Vehicle to A Stranger

We use the face region of the clothes region of the person to compute the distance between the person and the vehicle and determine if the state of the vehicle is *safe* or *unsafe*. However, we only have the angular information of the face from the image. We need the height of the face to compute the distance. And we do not need the exact distance between the person and the vehicle to decide the state of the vehicle. We only need to know whether the distance is smaller or larger than the *safe-distance* we defined in advance. To compute the distance of the person to the vehicle, we need to make a few assumptions first. We assume the person is standing on the ground, the length of his/her face is between 20cm and 25cm and the length of his/her body is around 50cm to 60cm. The following algorithm shows the details to compute the distance between the person and the vehicle.



(a)

(b)

Figure 6.3 The illustration of the distance between the person and the vehicle. (a) Distance computing using length of the face. (b) Distance computing using length of the clothes.

Algorithm 6.1 Calculation of the distance from the vehicle to the person by the face region.

Input: The detected face region R_{face} in an image, where R_{face} has two pairs of coordinates in the ICS: (f_{left}, f_{top}) and (f_{right}, f_{bottom}) which represent the boundary box of R_{face}. The range of the length of a human face [C_1, C_2].

Output: The range [D_0, D_1] of the distance between the person and the vehicle.

Steps:

Step 1. Transform the coordinates $(\frac{f_{left}+f_{right}}{2}, f_{top})$ and $(\frac{f_{left}+f_{right}}{2}, f_{buttom})$ into the longitude and latitude values (θ_1, φ_1) and (θ_2, φ_2), respectively.

Step 2. Referring to Figure 6.3(a) and assuming the distance of the person to be d, compute the value of $h_1 - h_2$ by

$$\tan\varphi_1 - \tan\varphi_2 = \frac{h_1 - h_2}{d}. \tag{6.1}$$

Step 3. Knowing that h_1-h_2 is equal to the height of the human face, take the range of $h_1 - h_2$ to be $C_1 \le h_1 - h_2 \le C_2$, and rewrite Equation (6.1) as

$$C_1 \le d \cdot (\tan\varphi_1 - \tan\varphi_2) \le C_2.$$

Step 4. Compute the range [D_0, D_1] of d by

$$D_0 = \frac{C_1}{\tan\varphi_1 - \tan\varphi_2} \text{ and } D_1 = \frac{C_2}{\tan\varphi_1 - \tan\varphi_2}.$$

Algorithm 6.2 Calculation of the distance from the vehicle to the person by the clothes region.

Input: The detected clothes region $R_{clothes}$ in an image, where $R_{clothes}$ has two pairs of coordinates in the ICS: (f_{left}, f_{top}) and (f_{right}, f_{bottom}) which represent the boundary box of $R_{clothes.}$ The range of the length of a human body [C_1, C_2].

Output: The range [D_0, D_1] of the distance between the person and the vehicle.

Steps:

Step 1. Transform the coordinates $(\frac{f_{left}+f_{right}}{2}, f_{top})$ and $(\frac{f_{left}+f_{right}}{2}, f_{buttom})$ into the longitude and latitude values (θ_1, φ_1) and (θ_2, φ_2), respectively.

Step 2. Referring to Figure 6.3(b) and assuming the distance between the person and the vehicle to be d, compute the value of $h_1 - h_2$ by

$$\tan\varphi_1 - \tan\varphi_2 = \frac{h_1 - h_2}{d}. \qquad (6.1)$$

Step 3. Knowing that h_1-h_2 is equal to the height of the human body, take the range of $h_1 - h_2$ to be $C_1 \le h_1 - h_2 \le C_2$, and rewrite Equation (6.1) as

$$C_1 \le d \cdot (\tan\varphi_1 - \tan\varphi_2) \le C_2.$$

Step 4. Compute the range [D_0, D_1] of d by

$$D_0 = \frac{C_1}{\tan\varphi_1 - \tan\varphi_2} \text{ and } D_1 = \frac{C_2}{\tan\varphi_1 - \tan\varphi_2}.$$

6.4 Method for Adjustment of Camera Orientation for Human Monitoring

After deciding the state of the vehicle, if the vehicle is in an *unsafe state*, we command the vehicle to escape. After an escape command is given to the vehicle, the vehicle will move backward to the position in the last navigation cycle. Since the camera is carried by the robot arm on the vehicle, the camera might not face the stranger because of the backward movement. To keep continuous monitoring of the stranger, we have to reset the orientation of the camera cycle after cycle. According the records of the motion of the vehicle, we have the turning angle θ from the records of the last movement of the vehicle. Moving backward means that the vehicle has to turn the angle of $-\theta$ for escape. Thus, as soon as the vehicle moves to the last position, we command the camera to turn the angle θ to keep the camera in the correct orientation for keeping in-view observation of the intruding person.

An experimental result is shown in Figure 6.4. At the cycle time $t = 1$, the vehicle detected a face in the grabbed image, and computed the distance of the person. The red box means the person transgressed the *safe-distance* we defined in advance. Thus, the vehicle moved backward to the last position at $t = 2$. The yellow box points out the detected face of the person who is at a further distance from the vehicle.

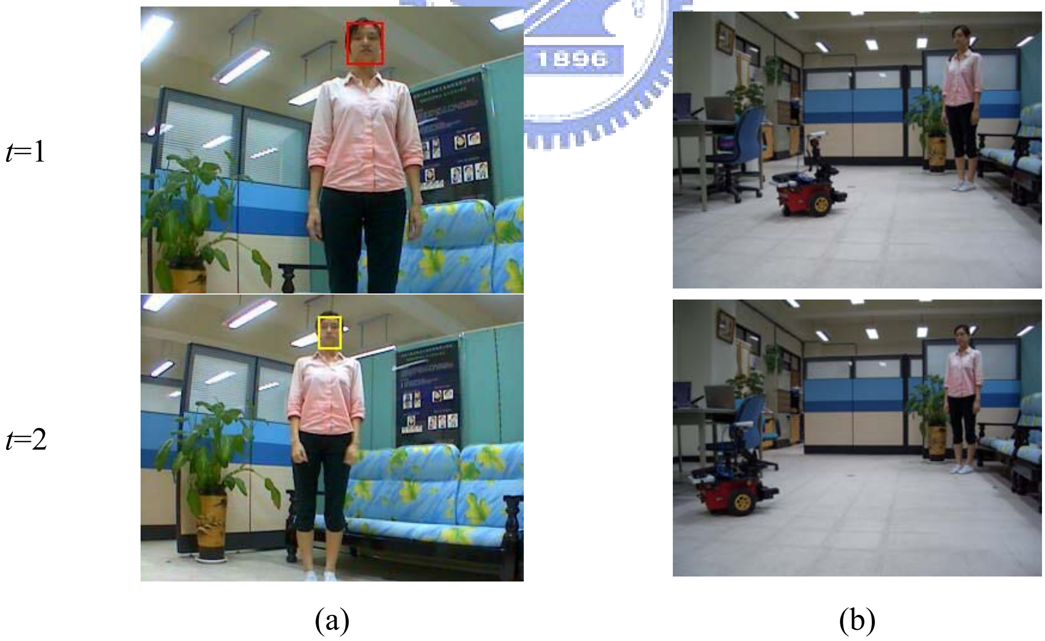


Figure 6.4 The experimental result that the vehicle moves backward to the last position when the vehicle detected the fact that the person is too close. (a) The images grabbed by the camera equipped on the vehicle. (b) The images captured from a third person's viewpoint.

Chapter 7
Experimental Results and Discussions

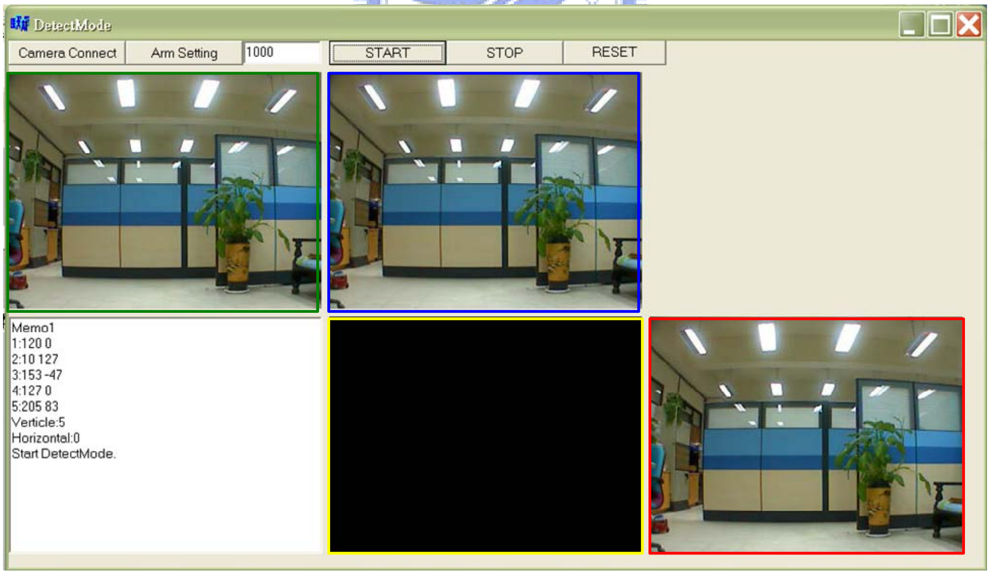
7.1 Experimental Results

We will show some experimental results of the proposed human detection and following system in this section. The user interface of the system is shown in Figure 7.1. All experiments of this study were conducted in our laboratory, Computer Vision Laboratory at the Department of Computer Science at National Chiao Tung University in Hsinchu, Taiwan. The proposed system has two modes: a detection mode and a tracking mode. The system will detect humans in acquired images in the detection mode. In the tracking mode, the system will track a target person by the feature of extracted clothes.

After a user presses the start button, the system will start monitoring the environment with a detection mode. In each cycle, the system determines the state of the vehicle (safe or unsafe) at first. To do this, the system needs to know if a person transgresses the safe-distance of the vehicle. And, a detected face region in the image is the only feature for computing the distance between the person and the vehicle. Therefore, the system conducts face detection first in each cycle, both in a detection mode or a tracking mode. If a face is detected and the distance of the person is smaller than the safe-distance, the vehicle is in an unsafe state and the system will command the vehicle to move backward to the last position and then finishes the current cycle. Otherwise, the system is in a safe state and continues the navigation process.

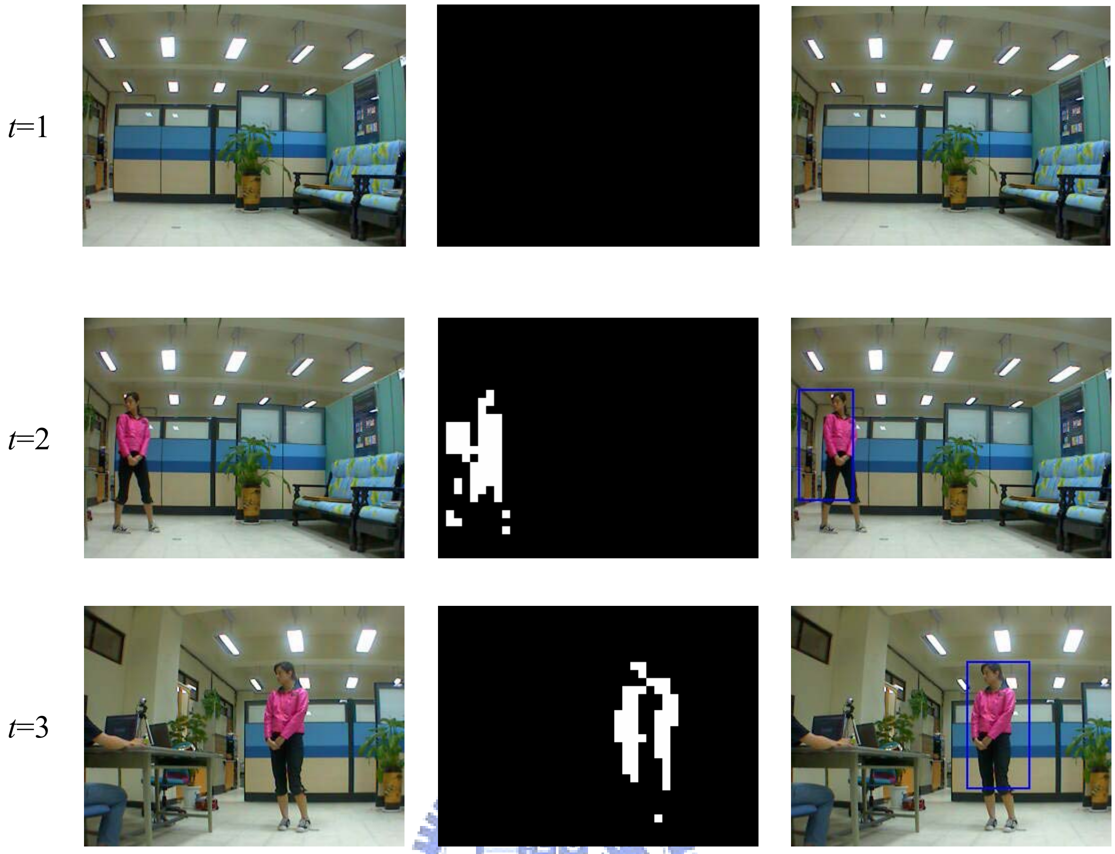
When the vehicle is in a safe state and the system is in a detection mode, the system conducts face detection. If a face is detected in the image, the system will extract the cloth region for tracking and change the detection mode to the tracking mode, and then finishes the current cycle, as shown in Figure 7.3. Else, if nothing is detected in the face detection mode, the system will conduct human body detection based on motion detection. If a human body is detected, the system will command the vehicle to move forward to the person, trying to get an image with a clear face region and then finishes the current cycle, as shown in Figure 7.2.

Otherwise, when the vehicle is in the tracking mode, which means the system already has the image of the clothes of the target person, the vehicle will track the target person using the intersection of the cloth image in each cycle. Until the system loses the target person, the system will change the tracking mode back to the detection mode. An experimental result is shown in Figure 7.4.



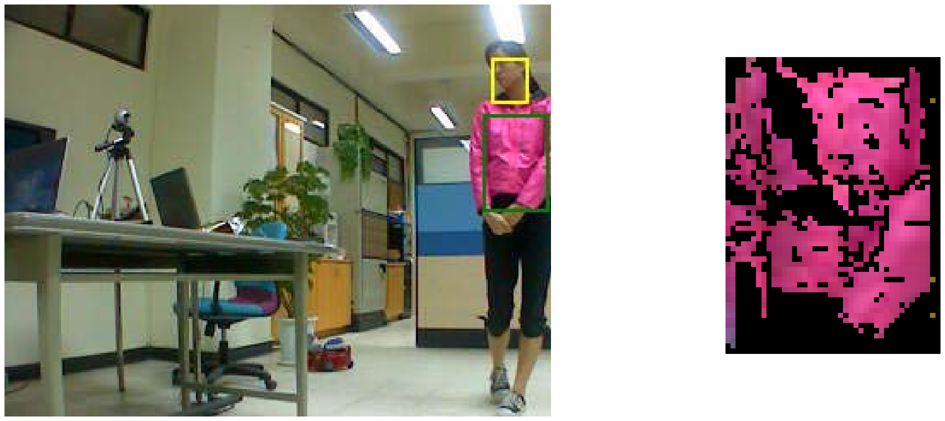
```
Memo1
1:120 0
2:10 127
3:153 -47
4:127 0
5:205 83
Verticle:5
Horizontal:0
Start DetectMode.
```

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



(a) (b) (c)

Figure 7.2 An experimental result of human body detection in the proposed system.(a) The input image. (b) The difference image. (c) The output image.



(a) (b)

Figure 7.3 An experimental result of face detection and the extraction of the cloth. (a) The output image with a detected face region and the extracted cloth region by region growing. (b) The image of the extracted cloth.

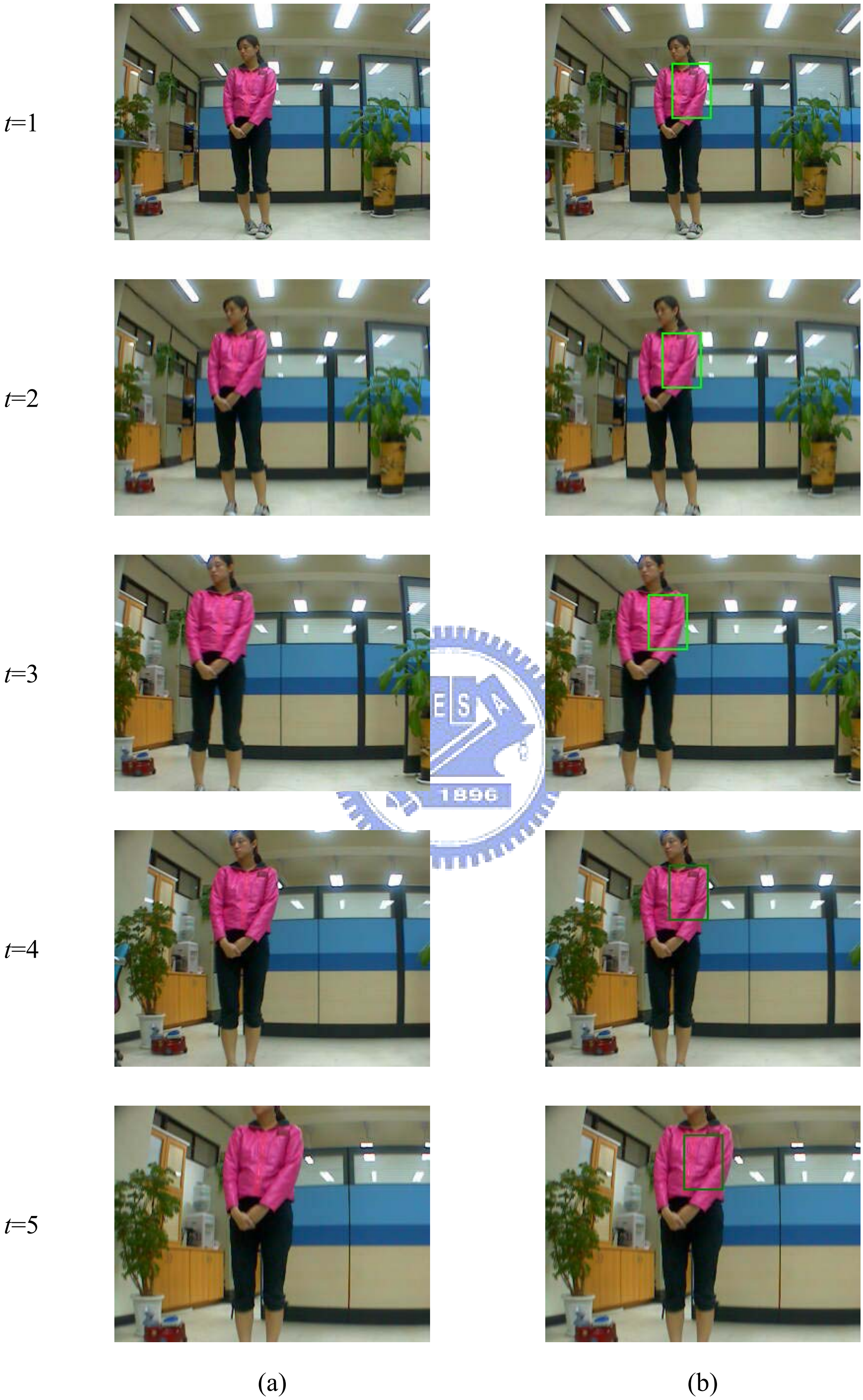


Figure 7.4 An experimental result of human tracking using the intersection of the cloth images. (a) The input image. (b) The output image.

7.2 Discussions

By analyzing the experimental results of navigation, some problems are identified as follows.

(1) The result of detecting the moving region by using blockwise frame differencing might become worse due to the condition of *jammed* environment. When the distance between a stationary object and the vehicle is too close, the relative movement will be amplified. If we slow down the speed of the vehicle, an erroneous judgment sometimes will occur.

(2) The color-based face detection is sensitive to the luminance of light in the environment. We can adjust the hue and the saturation of the camera manually in advance. But the change of the luminance is a factor we cannot predict in advance. Although lighting in indoor environment is more stable than outside, an image still can be affected easily due to the diaphragm of the camera.

(3) The human tracking process by cloth color cannot provide a measure of the distance between the vehicle and the target person. It only can compute the angular position of the target person. The system computes the distance of a person by the face region detected. In other words, only in the case of having detected a person's face can the camera calculate the distance.

Chapter 8
Conclusions and Suggestions for Future Works

8.1 Conclusions

Several techniques and strategies have been proposed in this study and integrated into an autonomous vehicle system for security patrolling in the indoor environments with human detection and following capabilities.

At first, a camera calibration by angular mapping is proposed. We calibrate the camera by a technique of *angular mapping*, which uses the concept of spherical coordinate system. Each point in the image is the projection result of a light ray onto the image sensor. The light ray can be described by a longitude angle and a latitude angle of the ray in the 3D world space. The angular mapping calibration technique using image analysis is used to compute the direction between the vehicle and a target. According to these angles and the height of the camera, we can know the relative locations of targets in images.

Next, some human detection techniques are proposed for indoor environment, including face detection and body detection. A human face is detected by the use of color and shape features in images. We use an elliptic skin model in the YC_bC_r color space to identify skin color regions and adopt an ellipse shape to fit the face contour. Besides, we propose a blockwise frame differencing method to extract moving objects in the image and decide if the moving object is similar to a human body.

Then, the human tracking techniques are proposed. After an intruding person is detected, the system will remember his/her clothes and track him/her. We propose a cloth region intersection method to predict the motion of a person to track him/her. Also, we record all the motions of the target person, and compute accordingly a parameter for the motion prediction of the target person.

In addition, a vehicle escape method by safe-distance keeping is proposed. We designed a function for the vehicle to escape from offensive strangers by a technique of *safe-distance keeping*. From the coordinates of the detected face region in an image, we compute the distance between the person and the vehicle. If the distance is smaller than a pre-defined safe-distance, the vehicle is commanded to escape by moving backward to the last position.

The experimental results shown in the previous chapter 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 vehicle system with a robot arm. Several suggestions and related issues are worth further investigation in the future. We state them as follows.

(1) In this study, we proposed a skin color model assuming an environment with uniform lighting. Thus, adding a capability of skin color learning to adapt the system to the changes of environment lighting is a suggestion for future work.

(2) We use clothes colors as the only feature of the clothes. To improve the extraction of clothes, we suggest conducting clothes tracking by different features, such as texture and shape, to eliminate errors caused by the case where the clothes color is similar to the background.

(3) In this study, we only proposed techniques of human detection and tracking. Adding a face recognition capability to recognize specified persons can make the vehicle react differently with different people.

(4) Since the vehicle only has one camera on it, the escape paths of the vehicle can only follow the previous paths of the vehicle. We suggest adding an omni-directional camera to plan better escape paths of the vehicle more conveniently.

References

[1] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in *Proceedings of the IEEE Image Understanding Workshop*, pp. 129-136. 1998.

[2] J. Hwang, Y. Ooi and S. Ozawa, “A Visual Feedback Control System for tracking and Zooming a target,” in *Proceedings of the International Conference on Industrial Electronics, Control & Instrumentation*, San Diego, USA, vol. 2, pp.740-745 ; November 1992.

[3] P. Nordlund and T. Uhlin, “Closing the loop: detection and pursuit of a moving object by a moving observer,” *Image and Vision Computing,* vol. 14, no.4, pp. 265-275, May 1996.

[4] J.L. Barron, D.J. Fleet, S.S. Beauchemin and T.A. Burkitt, “Performance of optical flow techniques,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, pp. 236-242, June 1992.

[5] D. Murray and A. Basu, “Motion tracking with an active camera,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, no.5, pp. 449-459, May 1994.

[6] J. Odobez and P. Bouthemy, “Detection of multiple moving objects using multiscale MRF with camera motion compensation,” in *Proceedings of the International Conference of Image Processing*, Austin, Texas, vol.2, pp. 257-261, November 1994.

[7] S. Araki, T. Matsuoaka, N. Yokoya and H. Takemura, “Realtime tracking of multiple moving object contours in a moving camera image sequence,” *IEICE*

Transaction on Information and Systems, vol. E83-D, no. 7, July 2000.

[8] A. Arsenio and J. Santos-Victor, “Robust visual tracking by an active observer,” in *Proceedings of the International Symposium on Intelligent Robot Systems,* vol. *3*, pp. 1342-1347, 1997.

[9] L. Zhao and C. Thorpe, “Stereo and Neural Network-based Pedestrian Detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148 -154, September 2000.

[10] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, “Pedestrian Detection in Infrared Images,” in *Proceedings of IEEE Intelligent Vehicles Symposium 2003*, Columbus, USA, pp. 662-667, June 2003.

[11] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, “Skin detection in video under changing illumination conditions,” in *Proceedings of IEEE International Conference on Pattern Recognition*, Barcelona, Spain, vol.1, pp. 839--842, 2000.

[12] P. Viola, M. J. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” in *Proceedings of IEEE International Conference on Computer Vision*, Nice, France, pp. 734–741, October 2003.

[13] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M.Tanaka, “A stereo machine for video rate dense depth mapping and its new applications,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 109-202, June 1996.

[14] Y. Dai and Y. Nakano, “Face-texture model based SGLD and its application,” *Pattern Recognition*, vol. 29, pp. 1007–1017, June 1996.

[15] P. Fieguth and D. Terzopoulos, “Color-based tracking of heads and other mobile objects at video frame rates,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 21-27,

1997.

[16] D. Ayers & M. Shah, “Recognizing human actions in a static room,” *IEEE Workshop on Applications of Computer Vision*, pp. 42-47, October 1998.

[17] D. Li, “Moving objects detection by block comparison,” in *Proceedings of IEEE International Conference on Electronics, Circuits, and Systems*, Beirut, Lebanon, vol. 1, pp. 341-344, 2000.

[18] B. Heisele and C. Wohler, “Motion-based recognition of pedestrians,” in *Proceedings of International Conference on Pattern Recognition*, Brisbane, Australia, vol. 2, pp. 1325-1330, August 1998.

[19] C. Papageorgiou, T. Evgeniou, and T. Poggio, “A trainable pedestrian detection system,” in *Proceedings of IEEE International Conference on Intelligent Vehicles*, Germany, pp. 241–246, October 1998.

[20] D. Chai, A. Bouzerdoum, “A Bayesian approach to skin color classification in YCbCr color space,” in *Proceedings of Region Ten Conference*, Kuala Lumpur, Malaysia, vol. 2, pp. 421-424, September 2000.

[21] D. Chai, K. N. Ngan, “Face segmentation using skin-color map in videophone applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol.9, no.4, pp.551-564, June 1999.

[22] J. Y. Lee and S. I. Yoo, “An elliptical boundary model for skin color detection,” in *Proceedings of International Conference on Imaging Science, Systems, and Technology*, Las Vegas, USA, pp. 579–584, June 2002.

[23] Philippe COLANTONY, “Color Space Transformation,” 2004, Available online: http://www.couleur.org/index.php?page=transformations

[24] BitJazz Inc., “Sheervideo: About: Synchromy, ” Available online: http://www.bitjazz.com/sheervideo/about/synchromy.shtml