國立交通大學

資訊學院

資訊科學與工程研究所

博士論文

精準且有自適能力之環場視覺技術及應用之研究

A Study on Accurate and Adaptive Omni-Vision Techniques and Applications

研究生: 石神恩

指導教授: 蔡文祥 博士

中華民國 一百零二 年 八 月

精準且有自適能力之環場視覺技術及應用之研究

A Study on Accurate and Adaptive Omni-vision Techniques and Applications

研　究　生：石 神 恩　　**Student: Shen-En Shih**

指 導 教 授：蔡 文 祥 博士　　**Advisor: Dr. Wen-Hsiang Tsai**

國 立 交 通 大 學 資 訊 學 院

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Dissertation Submitted to
Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science and Engineering

August 2013
Hsinchu, Taiwan, 300
Republic of China

中華民國 一百零二 年 八 月

精準且有自適能力之環場視覺技術及應用之研究

研究生：石神恩　　指導教授：蔡文祥博士

國立交通大學資訊學院

資訊工程與科學研究所

摘要

為了能使電腦與四周環境互動，環場視覺是一項極其有效且十分重要的技術。與傳統電腦視覺技術相比，環場視覺強調其在單一時間點能對大範圍環境取景之能力，而不用在攝影機上加裝馬達裝置來週期性地轉移攝影機，更不需用多部攝影機來對環境取景。由上述可知，在環場視覺技術中我們可以避免影像接合、攝影機換手、多攝影機特徵連續追蹤等複雜問題。為了達到大範圍取景之目的，有兩種特殊設計的環場攝影機較常被使用，其一是反射式環場攝影機，另一種是魚眼攝影機。其中，前者是將一個特殊形狀的反射式鏡面擺放在一傳統攝影機前方，藉由該鏡面來研展攝影機的可視範圍；後者是利用一特殊的魚眼透鏡來研展其可視範圍。然而，因為極大範圍的環境資訊被濃縮於一張傳統大小的影像中，環場攝影機所擷取到的影像必定有十分嚴重的扭曲，這也使得後續影像分析的工作變得困難且複雜許多。雖然將影像扭曲校正回來是其中一種簡單的解決方式，然而因為扭曲造成影像解析度的不同，校正回來的影像在某些區域會十分地模糊，在影像分析後會產生不穩定的結果。更甚者，上述扭曲校正的過程也需要些許運算能力，在即時應用及嵌入式系統中都較不適用。

為了克服環場攝影機擷取到的嚴重扭曲，我們提出了在扭曲影像上精準且穩定地偵測空間中直線的方法。另外，我們也針對各種利用環場攝影機偵測空間中直線的應用，提出改良的攝影機模型，並且也提出一套方便的校正程序來校正環場攝影機。此校正程序只需使用空間中的一條直線特徵，且不需要測量其位置及

方向，使得整個校正程序變得十分簡單，且可讓一般使用者方便地進行校正，使環場視覺技術朝消費電子更邁進一步。

另外，從消費者的角度上來看，我們應該也要能讓一般使用者方便地架設一套環場視覺系統。在此方面，我們提出一套新的雙眼視覺系統，此系統可讓使用者任意地擺放兩部環場攝影機。在擺放完成後，系統會自動利用環境中之直線特徵來回推攝影機的位置及角度，從而正確地計算立體資訊，以供各種人機互動應用使用。另一方面，針對需要取得十分精準之立體資訊的應用中，我們也提出一套最佳化架構以及三個最佳化演算法，其可告訴使用者如何擺放該二部環場攝影機的位置及角度，以求得最佳之立體資訊。根據這些最佳化演算法，使用者將可建構出能進行精準立體測量之雙眼環場視覺系統。

最後，我們也對上述所提出之各種環場視覺技術進行延伸研究，開發一套室內停車場管控系統。此系統可利用假設於天花板之各環場攝影機，自動地分析停車場中各停車格之位置，並自動找出空的停車格位置，以利駕駛找尋停車位。與現有系統相比，我們提出的系統因為攝影機的可視範圍較大，所以只需要較少的攝影機數量；另外，因為我們提出的系統可自動分析停車格位置，因此其系統建置過程會便利許多。

在可行性及效率評估中，我們已對上述各方法及技術進行理論及實驗分析，並得到良好之實驗結果。

A Study on Accurate and Adaptive Omni-Vision Techniques and Applications

Student: Shen-En Shih **Advisor: Dr. Wen-Hsiang Tsai**

Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

Abstract

Omni-vision is an important and effective technique to make computers be aware of the surrounded environment. Different from traditional computer vision techniques, omni-vision ones emphasize more on capturing the environment information within a very wide area at one time without adding a motor control to the camera, moving the camera periodically, or using multiple cameras. Such techniques can avoid the difficulties of image stitching, camera hand-off, feature tracking over different cameras, etc. To achieve the capability of capturing information of a wide area, two special kinds of cameras are commonly used, which are catadioptric omni-directional cameras, and fisheye-lens cameras. The formal ones use a specially-designed reflective mirror to extend the viewing field, and the latter ones use a fisheye-lens to achieve the goal. However, since the environment information captured from a wide area is all compressed in a relatively small image, the captured image is inevitably heavily distorted, which makes the image analysis task much more difficult and complicated. Although, an easy and feasible way to deal with the heavy distortion is to unwarp the captured images to yield an image looking like one captured by a conventional perspective camera. However, since the resolution distributions captured by omni-directional cameras and by conventional perspective camera are quite different from each other, an unwrapped image becomes much more blurred in some regions, making image analysis tasks unstable and unreliable. Furthermore, the

unwarping process needs some computation power, making it unsuitable to real-time applications and embedded systems with restricted computation power.

To deal with the heavily-distorted images captured by omni-directional cameras, an accurate and reliable space line detection method without unwarping the distorted image is proposed. Also, to model the imaging process conducted by an omni-directional camera, a new camera model along with a convenient calibration process to calibrate an omni-camera easily is proposed. This new calibration technique requires only one straight line in the environment without knowing the position or direction of the line, making it possible for non-technical user to conduct the calibration work which is a big step toward consumer electronics.

In addition, from the viewpoint of a consumer, the setup procedure of an omni-vision system should be sufficiently convenient for use by a typical user with no technical background. In this sense, a new binocular omni-vision system is proposed, which allows the user to place the two omni-directional cameras freely at any positions and with any orientations. After the two cameras are placed, the system can automatically derive the cameras' positions and orientations via analysis of the space lines within the environment. As a result, the binocular omni-vision system can calculate 3D information correctly for use in many advanced human-machine interaction applications. Furthermore, for applications requiring precise 3D information, an optimization framework along with three different optimization algorithms are proposed as well to tell the user where to place the two omni-cameras, and what are the best orientations. With these optimization algorithms, the user can set up a binocular omni-vision system which acquires the most precise 3D data.

Finally, the proposed omni-vision techniques are extended for uses in the application of indoor parking lot management. The proposed system for this application utilizes the omni-directional cameras mounted on the ceiling, and automatically analyzes the acquired images to obtain the locations of the parking spaces and detect vacant parking spaces. Different from existing similar application systems, the proposed one requires fewer cameras due to the wider fields of view of the cameras, and is much more convenient to set up because of the developed automatic parking-space analysis capability.

The feasibility and effectiveness of all the above proposed methods and systems are demonstrated by theoretical analyses and good experimental results.

Acknowledgements

I would like to express my sincere appreciation to my advisor, Professor Wen-Hsiang Tsai, for his guidance and patience throughout the course of this dissertation study. I would also like to acknowledge the very helpful comments and suggestions from the members of the oral defense committee and also those from the reviewers for parts of this dissertation that were submitted for journal publication. Thanks are also extended to the colleagues in the Computer Vision Laboratory at National Chiao Tung University for their valuable help and comments during this study.

Finally, I am so grateful to my parents and brother for their love, support, and endurance. This dissertation is dedicated to them.

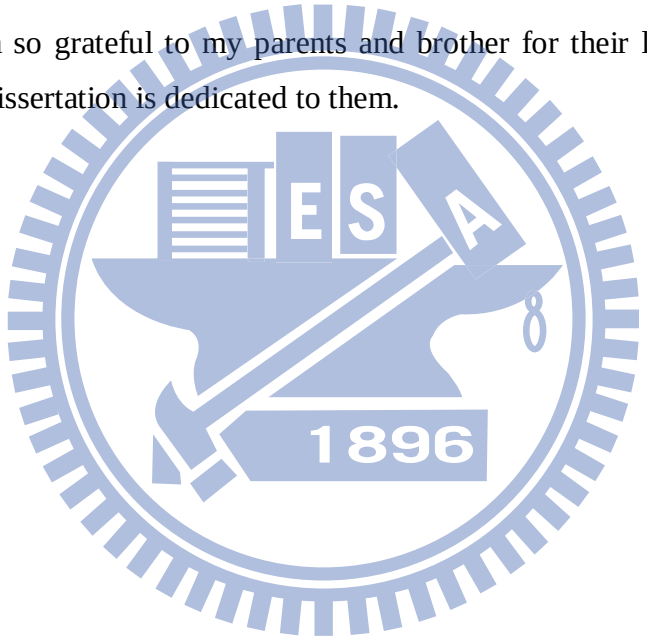


Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Research Motivation

With the advance of technologies, various types of vision systems have been designed for many applications [1]-[5], like virtual and augmented reality, video surveillance, environment modeling, TV games, etc. Most of these applications require acquisitions of the 3D data, meaning in turn the need of precise system calibration and setup works to yield accurate 3D data computation results in the application environment. However, from a consumer's viewpoint, it is unreasonable to ask a user to set up a vision system very accurately, requiring, e.g., the system cameras to be affixed at accurate locations in precise orientations. Contrarily, it is usually desired to allow a user to choose freely where to set up the system components. Additionally, many vision systems used for the previously-mentioned applications are composed of traditional projective cameras which collect less visual information than systems using omni-directional cameras (*omni-cameras*).

To overcome these difficulties, a 3D vision system which consists of two omni-cameras with a capability of *automatic adaptation* to any camera setup is proposed in this dissertation study. While establishing this system, the user is allowed to place the two cameras *freely* in any orientations with any displacement. Then, the system can *accurately* analyze some environment features, and *automatically* adapt to the unknown system configuration by deriving the locations and orientations of the two cameras. As a result, an accurate and adaptive omni-vision system can be constructed.

On the other hand, in the binocular omni-vision applications which requires precise 3D acquisitions, an optimization method is required to tell the user where to place the cameras, and what are the best orientations. Since many existing optimization techniques only focused on dealing with binocular vision systems using conventional perspective cameras, a specially optimization framework and algorithm are thus required to produce the optimal *configuration*.

Furthermore, it is noticed that the omni-vision techniques are not yet popular enough. To bring the omni-vision into our daily life, the omni-vision system must be able to be easily set up, including the calibration process and the system setup phase.

To this aim, an convenient indoor parking lot management system is desired, which uses the omni-cameras mounted on the ceiling, and the system must be able to be easily calibrated and set up by a typical user with no technical background.

Fulfillments of the aforementioned goals of this thesis study together enhance the state-of-art studies on omni-vision techniques, yielding a further step toward to the area of consumer electronics.

1.2 Survey of Related Works

The related works are categorized into several directions and reviewed as follows.

(1) Catadioptric camera

A catadioptric omni-camera is a combination of a reflective mirror and a CCD camera as shown in Fig. 1.1(a). An image taken by such a kind of camera is shown in Fig. 1.1(b). With the aid of reflective surface from mirror, a camera of this type can obtain larger fields of view in the acquired images. The lens of the CCD camera may be of a perspective or orthographic projection type, and the mirror surface of a catadioptric omni-camera may be in various shapes such as hyperbolic, circular, parabolic, or conical one, as illustrated in Fig. 1.2. With distinctive mirrors or lenses, the images and calibration methods of the cameras are different in this category. Some works of using this type of camera can be found in [6]-[14].

(2) Dioptric camera

A dioptric omni-camera, looking like a traditional camera, has no reflective mirror, but is with a “wider-angle” lens. It can capture incoming light rays from a wider field of view to form an omni-image. An illustration of such an imaging difference between traditional and catadioptric cameras is shown in Fig. 1.3. The lens shape design of this group of cameras decides the formed images and their calibration methods. An example of this kind of omni-camera is the *fish-eye camera*. An image acquired by a fish-eye camera is shown in Fig. 1.4. Some works of using fish-eye cameras can be found in [15]-[17].

(a) (b)

Fig. 1.1 A catadioptric camera. (a) Structure of camera. (b) Acquired image.

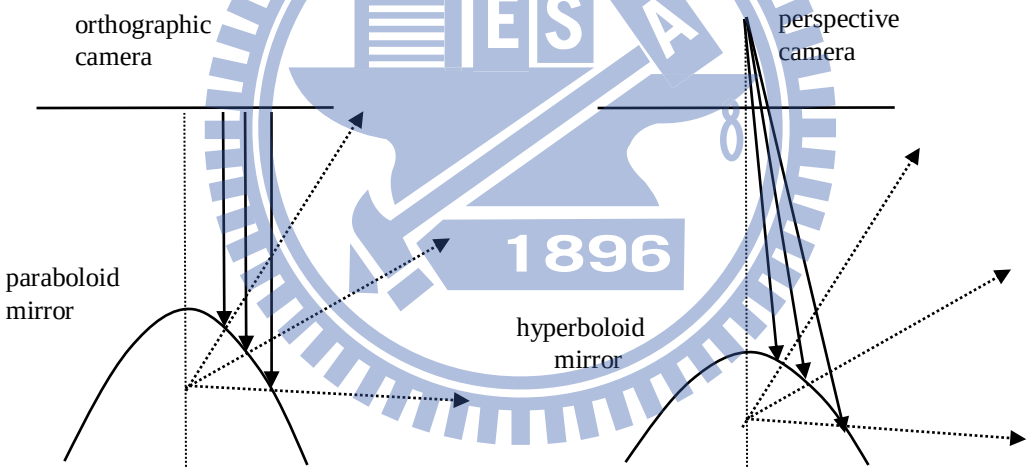
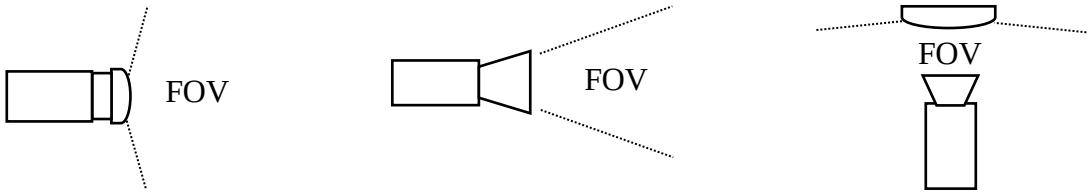


Fig. 1.2 Illustration of camera and reflective mirror types.



(a) (b) (c)

Fig. 1.3 FOVs of different camera types. (a) Dioptric camera. (b) Traditional (perspective) camera. (c) Catadioptric camera. [18]

Fig. 1.4 An image acquired by a fish-eye camera.

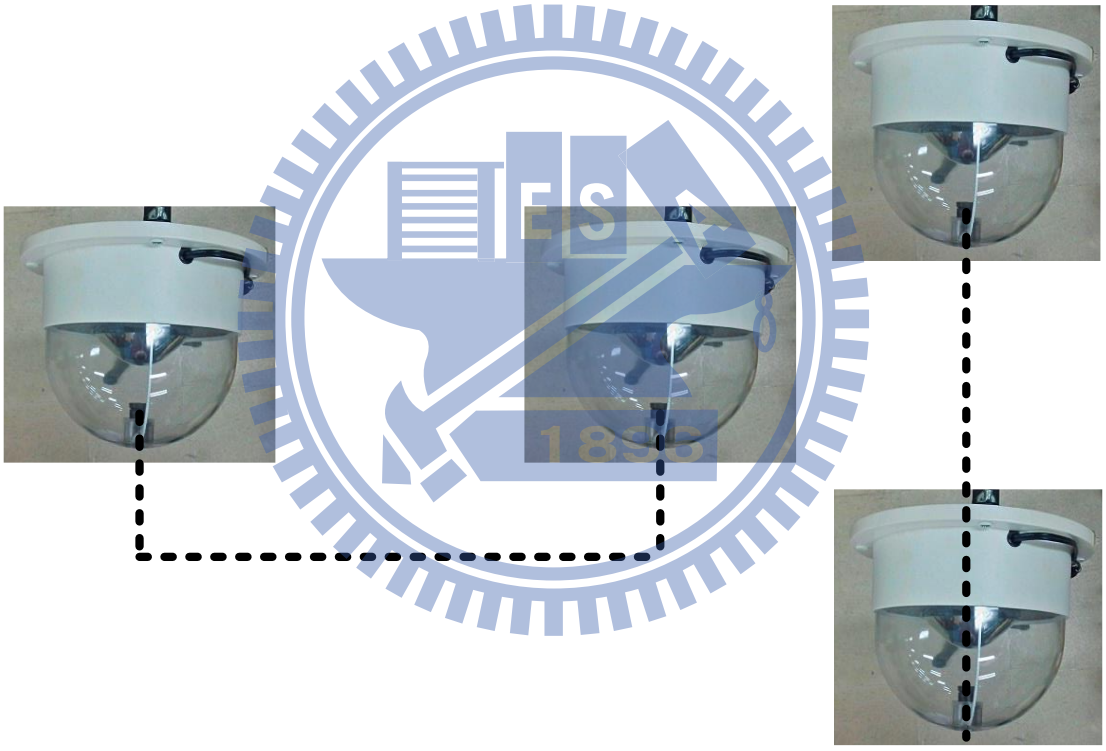
(3) Binocular vision systems

A binocular vision system is composed of two cameras, typically perspective ones, which are placed at different locations. Differently, a binocular omni-vision system consists of two omni-cameras, which can be catadioptric or fisheye-lens ones. An illustration is shown in Fig. 1.5, where two kinds of such camera pairs are seen. In theory, by using the corresponding pixels in the two images acquired from the cameras, stereo information can be derived from such correspondences. Most existing researches were focused on binocular vision systems using perspective cameras [19][20]; contrarily, researches on binocular vision systems with omni-directional cameras are less [21] with many open problems waiting to be solved.

(4) Human-machine interface systems

Human-machine interaction has been intensively studied for many years. Laakso and Laakso [22] proposed a multiplayer game system using a top-view camera, which

maps player avatar movements to physical ones, and used hand gestures to trigger actions. In [23], a special human-machine interface was proposed by Magee *et al.*, which uses the symmetry between the left and right human eyes to control computer applications. Zabulis *et al.* [24] proposed a vision system composed of eight cameras mounted at room corners and two cameras mounted on the ceiling to localize multiple persons for wide-area exercise and entertainment applications. Starck *et al.* [25] proposed an advanced 3-D production studio with multiple cameras. The design considerations were first identified in that study, and some evaluation methods were proposed to provide an insight into different design decisions.



(a) (b)

Fig. 1.5 Two types of binocular omni-vision systems. (a) Laterally parallel combination. (b) Longitudinally coaxial combination.

(5) Geometric feature extractions

Geometric features, like points, lines, spheres, etc., in environments encode important information for on-line calibrations and adaptations [26][27]. Several methods have been proposed to detect such features in environments. Ying [28][29]

proposed several methods to detect geometric features when calibrating catadioptric cameras, which use the Hough transform to find the camera parameters by fitting detected line features into conic sections. Duan *et al.* [30] proposed a method to calibrate the effective focal length of the central catadioptric camera using a single space line under the condition that other parameters have been calibrated previously. Von Gioi *et al.* [31] proposed a method to detect line segments in perspective images, which gives accurate results with a controlled number of false detections and requires no parameter tuning. Wu and Tsai [32] proposed a method to detect lines directly in an omni-image using a Hough transform process without unwarping the omni-image. Maybank *et al.* [33] proposed a method based on the Fisher-Rao metric to detect lines in paracatdioptric images, which has the advantage that it does not produce multiple detections of a single space line. Yamazawa *et al.* [34] proposed a method to reconstruct 3D line segments in images taken with three omni-cameras in known poses based on trinocular vision by the use of the Gaussian sphere and a cubic Hough space [35]. Li *et al.* [36] proposed a vanishing point detection method based on cascaded 1-D Hough transforms, which requires only a small amount of computation time without losing accuracy.

(6) System configuration optimization

Several methods have been proposed to derive optimal vision system configurations. Among them, one popular way is to assess the 3D measurement error by the use of the covariance matrix [37]-[42]. For this, Wenhardt *et al.* [37] determined the locations of mobile cameras to yield the best 3D model reconstruction by assessing the covariance of the resulting 3D data in three ways, namely, using the determinant, eigenvalues, and trace of the covariance matrix, respectively. Hoppe *et al.* [40] used the eigenvalues of the covariance matrix to model the 3D measurement error for precise camera localization and object modeling. Alsadik *et al.* [39] established a camera network for precise reconstruction of a cultural heritage object by the use of the trace of the covariance matrix. Olague and Mohr [41] proposed a multi-cellular genetic algorithm to decide camera locations, which yield minimal 3D measurement errors, by the use of the maximum diagonal element of the covariance matrix. Zhang [38] determined the optimal 2D spatial placement of multiple sensors participating in a robot perception task utilizing the determinant of the covariance

matrix. Rivera-Rios, et al. [42] analyzed 3D measurement errors due to feature-point localization errors and found accordingly the optimal camera pose by the mean-square-error criterion using the covariance matrix of the 3D measurement data. In these methods, the precisions of the 3D measurements are all assessed by the use of the covariance matrix. Additionally, a local-affineness assumption was made when deriving the covariance matrix (as stated in [43]).

1.3 Contribution of This Study

The main contributions of this dissertation study are summarized in the following.

1. A new camera model for omni-directional cameras including catadioptric ones and fisheye-lens ones is proposed. This new model utilizes a new invariant property found in this study, so that the camera model is simpler than the conventional one, and can be calibrated more robustly using a small amount of environment features.
2. A new calibration technique is proposed for the aim of easily calibrating the omni-directional camera by a typical user with no technical background. This new calibration technique requires only one straight line in the environment without knowing the position or direction of the line, so that it is possible to be calibrated by a non-technical user, yielding a big step toward consumer electronics.
3. An easy-to-set-up binocular omni-vision system is proposed in this study, which allows the user to place the two omni-directional cameras freely at *any* positions and with *any* orientations. After the two cameras are placed, the system can automatically derive the cameras' positions and orientations via analysis of the space lines within the environment. As a result, the binocular omni-vision system can calculate 3D information correctly, and can be utilized in many advanced human-machine interaction applications.
4. To the applications requiring the binocular omni-vision systems to acquire precise 3D information, an optimization framework along with three different optimization algorithms are proposed. After telling the requirements and constraints of the 3D application, the optimization algorithm can generate the optimal system configuration, giving the user the best locations to place the two

omni-cameras and their best orientations. With these optimization algorithms, the user can set up a binocular omni-vision system which acquires the most precise 3D data.

5. To extend the proposed omni-vision techniques for uses in other applications, a parking-lot management system is proposed, which uses the omni-directional cameras (including catadioptric and fisheye-lens ones) mounted on the ceiling. After the omni-cameras are set up, the system automatically analyzes the locations of the parking spaces and detects vacant parking spaces after conducting a background learning process. Different from conventional parking-lot management system, the proposed system requires fewer cameras due to the wider fields of view of the used cameras, and is much more convenient for the user to set up because of its automatic capability of analyzing parking spaces.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, an overview of the proposed techniques are given. In Chapter 3, the structure of omni-cameras and the unifying camera model for catadioptric omni-cameras and fisheye-lens cameras are briefly reviewed, followed by an introduction to the proposed simplified camera model. The proposed space line detection method is introduced in Chapter 4, including the problems of existing methods, derivations of the proposed method, and some experimental results. In Chapter 5, the proposed easy-to-set-up binocular omni-vision system is described, which allows the user to place the two omni-directional cameras freely at *any* positions and with *any* orientations, and can automatically derive the camera poses using environment features. the proposed optimization framework along with three different optimization algorithms for applications requiring acquisitions of precise 3D information are described in Chapter 6, which generates the optimal system configuration to give the best locations and orientations of the two omni-cameras. The proposed convenient vision-based parking lot management system is described in Chapter 7, which uses the omni-cameras mounted on the ceiling to automatically analyze the locations of the parking spaces and detect vacant parking spaces after conducting the background learning process. Finally, in the last chapter, conclusions of this study and some suggestions for future research are included.

Chapter 2 Overview of Proposed Techniques and Ideas

In this chapter, we describe the main ideas and techniques of the proposed methods for used in omni-vision systems.

2.1 A Modified Unifying Model for Omni-cameras

A new modified camera model for omni-cameras, including single viewpoint catadioptric omni-directional cameras, fisheye-lens cameras, etc., is proposed in this study. This new camera model is modified from the unifying omni-camera model proposed by Geyer and Daniilidis [44], and it is with fewer camera parameters than the original unifying model. In the proposed modified model, we investigated an important invariant property, so that one of the parameters in the original unifying omni-camera model can be eliminated by replacing it with an optimal approximated value, and in the meantime preserve an important property regarding to space line detection. Then, according to this invariant property, we design a series of experiments to find an optimal value to approximate one of the parameters in the original unifying model proposed by Geyer and Daniilidis. Furthermore, since the proposed modified model has fewer parameters than the original one, we also proposed a new calibration model using only one straight line in the 3D world, without knowing its position, direction, or length.

To sum up, we investigate a new invariant property regarding to the projections of the straight lines by omni-directional cameras, and accordingly proposed a new modified unifying omni-camera model which is with fewer parameters than the original model. Furthermore, a new calibration method is also proposed which can be used to calibrate an omni-camera using only one straight line with no knowledge about its position, direction, or length. Comparing with existing omni-camera models, the proposed one has an great advantage since it can be calibrated reliably from one straight line, so it facilitates a non-technical user to conduct the calibration process without difficulty because it requires no extra calibration target.

2.2 Space Line Detection Techniques for Omni-cameras by Equal-width Curve Extractions

An improved detection method is proposed in this study to robustly detect straight lines (i.e., space lines) in the 3D world from the images captured by omni-cameras. The proposed method detects the space lines from the captured omni-images directly without unwarping the captured omni-images, so the processing time may be faster than the methods needing unwarpings. Traditionally, the Hough Transform can be used to detect space lines from omni-images without unwarping it; however, it is showed in this study that the traditional ways all have some problems when detecting the space lines, so the results are imprecise and unrobust. In this study, we first identified three main different conventional approaches to detect space lines, and analyzed the reasons of the imprecision and unrobustness. From the analyze results, it is figured out that an equal-width curve extraction technique can be used to yield a more precise and robust results when detecting space lines. As a result, a technique to extract equal-width curves is proposed using total differential concepts; consequently, an improved Hough transform technique is proposed to detect equal-width curves using the previously-mentioned equal-width extraction method.

From the experimental results, the proposed line detection method can detect space lines more precisely and robustly then the conventional methods. To the processing time, the proposed method only requires a bit longer running time than the conventional ones, but produces a much more precise results than the conventional ones.

2.3 Automatic Adaptation Techniques of Binocular Omni-vision Systems to Any System Setup

From the viewpoint of consumers, one of the important facts when deciding to buy a new system is the convenience of the system setup process, so a system with a convenient setting up process is very important in consumer electronics. In this aspect, a new binocular omni-vision system is proposed in this study, which can be easily deployed by users without any restrictions on the locations or orientations of

the cameras, and then the system can automatically adapt system parameters using only the straight lines in the environment. Specifically, the proposed vision system, as shown in Fig. 2.1, consists of two omni-cameras facing the user's activity area. Each camera is affixed firmly to the top of a rod, forming an *omni-camera stand*, with the camera's optical axis adjusted to be horizontal (i.e., parallel to the ground). After the two cameras are placed *freely* by the user with arbitrary locations and orientations, we utilize the straight lines within the surrounding environment as a hint to tell us the orientations of the cameras, and then use these lines to derive the two cameras' orientations. After deriving the orientations, the system will ask the user to stand at the middle region in front of the two cameras, and derive the distance between the two cameras (i.e., the baseline) with the use of the user's height. After deriving the orientations and the baseline, a coordinate system can be defined with no ambiguity, so the 3D data can be computed correctly. To further improve the correctness and robustness of the adaptation process, we also take the advantage of the property that the straight lines are mostly parallel or perpendicular to each other, so that the proposed adaptation process can be conducted without finding the line correspondences among the two omni-cameras, and in the meantime improves the correctness and robustness of the adaptation results.

To sum up, we proposed a new 3D vision system using two omni-cameras, which has a capability of *automatic adaptation* to *any* system setup for convenient *in-field* uses. The cameras are allowed to be placed *freely* in the environment at *any* location in *any* orientation, resulting in an *arbitrary system setup*. Then, by the use of space line features in environments, the proposed vision system can adapt *automatically* to the arbitrarily-established system configuration by just asking the user to stand still for a little moment in the middle region of the activity area in front of the two cameras. Contrarily, in traditional vision systems, the two cameras may be required to be parallel to each other, and the distance between the cameras may be required to be a fixed length. After this adaptation operation, 3D data can be computed *correctly* and *precisely*.

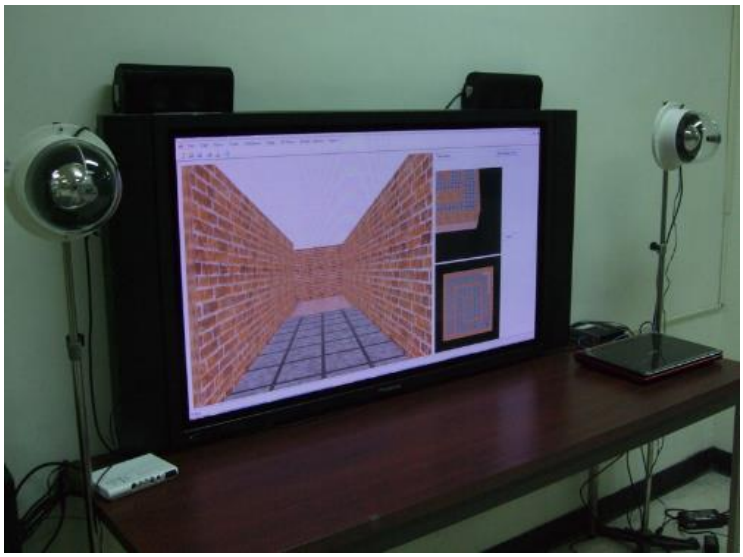


Fig. 2.1 The proposed binocular omni-vision system.

2.4 Optimal Design and Placement of Omni-cameras in Binocular Vision Systems for Accurate 3D Data Measurement

The optimal design of a vision system is an important issue for system deployments, and in this study, the optimization problem of designing an binocular omni-vision system is analyzed and solved by three different optimization methods. In a binocular omni-vision system, which is composed of two catadioptric omni-directional cameras with hyperboloidal-shaped mirrors, an optimal system design includes the optimal shapes of the two hyperboloidal mirrors, the optimal viewing angles of the perspective cameras, the optimal locations of the cameras, and the optimal directions of the camera, and in this study, we focused on finding a such configuration which can yield the most accurate 3D measurements. To solve this optimization problem, the first step is to design a function to estimate the *goodness* of a system configuration, and then design optimization methods which can minimize the function. In more detail, a *criterion* function is proposed in this study to estimate the accuracy of the 3D measurements yielded by a system configuration. That is, the criterion function takes the system parameters as its input arguments, including the camera poses, camera parameters, and the location of the feature point

whose 3D data is going to be measured, and produces a value indicates the *accuracy* of the 3D data yielded by the binocular omni-vision system.

For the criterion function, we use error propagation analysis techniques to estimate the accuracy of the 3D measurement, and the proposed criterion function is with analytical formulas, making it possible to design an analytical and non-iterative optimization method. After defining the criterion function, it is then used in an optimization framework to find the optimal system configurations for different shapes of system setup environments. For regular cases with rectangular cuboid-shaped 3D measurement and camera placement areas, two fast algorithms are proposed to solve the problem, one being bisection-based and relatively slower for deriving the optimal solution; and the other faster using analytic formulas for deriving a sub-optimal solution which is proved to be close to the optimal one in precision. For general cases, an iterative optimization method is proposed along with several speeding-up techniques to accelerate the optimization process. Experimental results of simulations and real application cases show the feasibility of the proposed optimization methods.

2.5 An Omni-vision-based Indoor Parking Lot System with the Capability of Automatic Parking Space Detection

A convenient indoor vision-based parking lot system using wide-angle fisheye-lens or catadioptric cameras is proposed, which is easy to set up by a user with no technical background. Easiness in the system setup comes mainly from the use of a new camera model which can be calibrated using only one space line without knowing its position and direction, as well as from the automatic detections of the parking space boundaries. Comparing with traditional parking lot systems, the traditional ones usually use perspective cameras, rather than use wide-angle cameras, such as fisheye-lens or catadioptric ones, are not commonly adopted yet. Furthermore, another one problem exists in the traditional systems is the complicated system setup procedure, including camera calibration, environment learning, object modeling, etc., whose implementation usually requires the user to have a lot of technical knowledge. From these viewpoints, an *intelligent* vision-based system using omni-cameras for parking lot management is proposed, which has the

following two merits: 1) the camera system can be set up *easily* by a common user with no technical background; 2) parking spaces can be detected *precisely*; and 3) vacant parking spaces can be identified *automatically* for convenient car parking.

In more detail, the omni-cameras mounted on the ceiling can be easily calibrated using straight lines in the environment using the new camera model and the new calibration method proposed in this study. After camera calibration is conducted, parking-space boundary lines are extracted automatically from input omni-images by a modified Hough transform with a new cell accumulation scheme, which can generates more accurate equal-width curves using the geometric relations of line positions and directions. To further improve the detection results of the parking-space boundary lines, the property that the boundary lines are either parallel or perpendicular to each other is taken into consideration to improve the results. After the boundary lines are detected, the user may easily add or remove the boundary lines by single clicks on images, and parking spaces can be segmented out by region growing by the use of the boundary lines. Finally, vacant parking spaces can be detected by a background subtraction scheme. A real vision-based parking lot has been established and relevant experiments conducted. Good experimental results show the correctness, feasibility, and robustness of the proposed methods.

Chapter 3 Omni-camera Structure and Models

In this chapter, the structure of the catadioptric omni-cameras are introduced first. Then, the conventional model for omni-cameras are briefly reviewed.

3.1 Catadioptric Omni-camera Structure

The catadioptric omni-camera is composed of a hyperboloidal mirror and a perspective camera looking toward the mirror, as depicted in Fig. 3.1. A camera coordinate system (CCS) *x*-*y*-*z* and an image coordinate system *u*-*v* are defined in such a way that the *x*- and *y*-axes are parallel to the *u*- and *v*-axes, respectively, and the two focal points of the mirror are at $O(0, 0, 0)$ and $O_c(0, 0, 2c)$. In this way, the mirror shape can be expressed [45] by

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

with its *eccentricity* ε being defined by $c/a > 1$. A *projection equation* describing the relation between the *complementary elevation angle* ϕ of a space point P and image coordinates (u, v) of the projected pixel p can be expressed [32] as

$$\tan\phi = \frac{(\varepsilon^2-1)\sin\tau}{(\varepsilon^2+1)\cos\tau - 2\varepsilon}, \tag{2}$$

where $\cos\tau$ and $\sin\tau$, as seen from Fig. 3.1, can be expressed by

$$\cos\tau = \frac{f}{\sqrt{u^2+v^2+f^2}}; \quad \sin\tau = \frac{\sqrt{u^2+v^2}}{\sqrt{u^2+v^2+f^2}}. \tag{3}$$

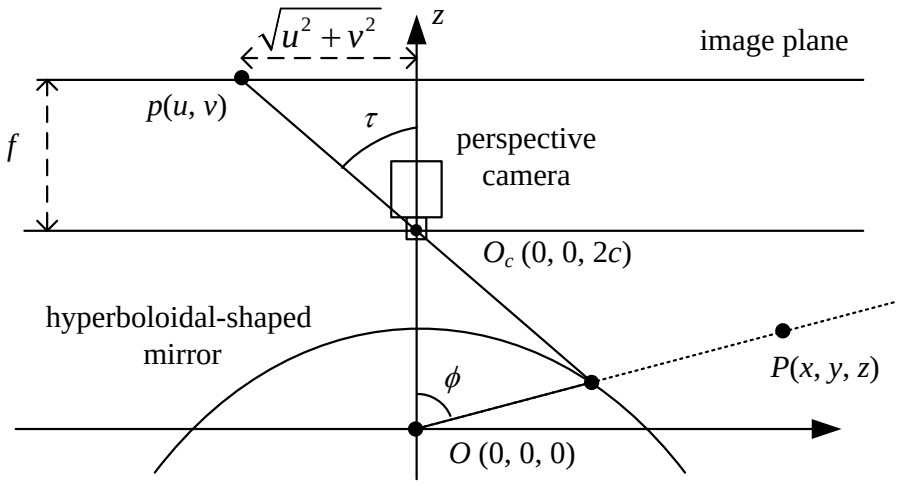


Fig. 3.1 Catadioptric omni-camera structure and its camera coordinate system.

3.2 Review of the Unifying Model for Omni-cameras

In the unifying model proposed by Geyer and Daniilidis [44], the relation between a space point P and its corresponding pixel p is described by a two-step mapping via a unit sphere as illustrated in Fig. 3.2. Specifically, point P is firstly projected onto a point P_s on the unit sphere like being viewed from the *effective viewpoint* O through the sphere. Then, P_s is projected perspectively into the pixel p on the image plane Π like being viewed from the "pinhole" point O_c (usually the lens center). The two involved parameters l and f_e as illustrated in Fig. 3.2 are defined in the same way as suggested by Ying and Hu [46].

The unifying model described above has been proved equivalent to models for perspective, parabolic catadioptric, and hyperbolic catadioptric cameras [44]. Additionally, it was shown by Ying and Hu [47] to be suitable for modeling fisheye-lens cameras as well.

To the resolution formula, Baker and Nayer [56] proposed a formula to calculate the resolutions at different pixels in an omni-image as follows. Let dA be an infinitesimal area on the image plane near a pixel p, which, as illustrated in Fig. 3.3, is the projection of an area in the space described by an infinitesimal solid angle $d\nu$ coming from a point P. The resolution of pixel p is formulated as

$$R(\varepsilon, f, \phi) = \frac{dA}{d\nu} = \frac{(\varepsilon^2 - 1)^2 (\varepsilon^2 + 2\varepsilon \cos\phi + 1)}{\left[2\varepsilon + (\varepsilon^2 + 1)\cos\phi\right]^3} f^2, \tag{4}$$

where ε is the eccentricity of the mirror, f is the focal length of the camera, and ϕ is the complementary elevation angle of P.

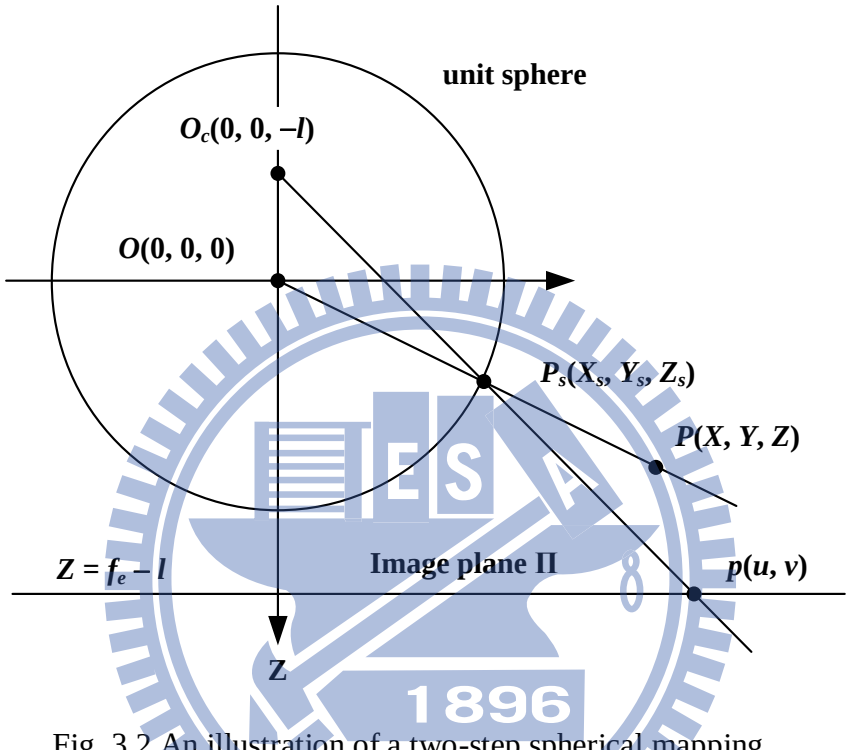


Fig. 3.2 An illustration of a two-step spherical mapping.

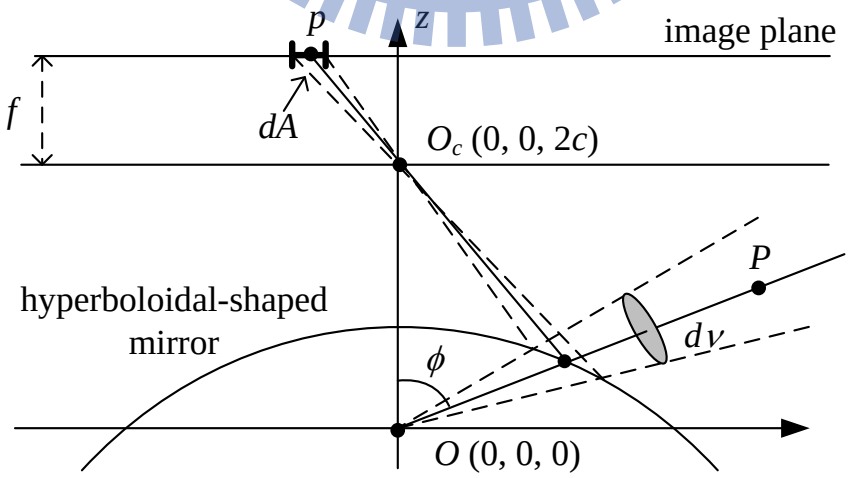


Fig. 3.3 Illustration for defining the resolution at a pixel.

Chapter 4 Space Line Detection for Omni-cameras by Equal-width Curve Extractions

In this chapter, the problems of existing methods for line detections in omni-images are described in Section 4.1, the proposed method to deal with this problem is described in Section 4.2, and some experimental results are presented in Section 4.3.

4.1 Problems of Existing Methods

The Hough transform is widely used in computer vision applications to detect shapes in noisy images. It includes three main steps: image thinning, cell value accumulation, and voting for peak value detection. In the image thinning step, a standard method is to conduct edge detection to extract edge pixels in the input image. In the cell value accumulation step, each edge pixel is transformed into a curve in the parameter space (also called *Hough space*), and the values of the corresponding elements in the Hough space, called *Hough cells*, are all incremented by one. Subsequently, the voting step is conducted to find the peaks (local maximums) in the Hough space, which are taken finally as the parameters of the detected shapes.

The cell value accumulation step is essential in the Hough transform, also known as the *evidence gathering* step. if a shape to be detected is described by a function F, then pixels contributing to the accumulation of the (largest) peak cell value in the Hough space theoretically are just those with their coordinates (u, v) satisfying the equation $F(u, v) = 0$. However, since these coordinates (u, v) in practice are with errors coming from quantization, noise, edge detection, imprecise camera calibration, etc. [51][52], the mentioned pixels, with such erroneous coordinates (u, v), instead will *not* all lie precisely on the curve $F(u, v) = 0$. To endure such imprecision, three different methods exist as described in the following.

The first conventional method is called *constant thresholding*, the inequality

$$\left|F(u,v)\right| \le T\,. \tag{5}$$

is used to decide whether an edge pixel with coordinates (u, v) contributes to the accumulation of a Hough cell value. This method can generate a desired equal-width curve as shown in Fig. 4.1(a) when detecting simple shapes; however, it generates bad results when dealing with complicated shapes as shown in Fig. 4.1(b). Furthermore, even if an equal-width curve can be generated, the relation between the threshold T and the curve width W is still unclear and it needs further analysis or experiments to derive a reasonable value for T.

The second method is by *examining each Hough cell*, which examines each Hough cell by two steps: (1) use (u, v) and function F to find a *parametric* equation describing all the curves going through (u, v); (2) regard the equation as the description of a hypersurface in the Hough space, and examine each cell to detect those intersecting this hypersurface, and increment their values. One drawback of this method is that, which pixels contribute to cell value accumulations depends on the cell *size*, and these pixels together are not of a desired equal-width shape, as shown later in the experimental results.

The third method is with the use of an *inverse function*. More specifically, the Hough cells with their values contributed by an edge pixel with coordinates (u, v) are determined by: (1) enumerate the first $n - 1$ parameters of the Hough space, where n is the dimension of the space; and (2) derive the n^{th} parameter by the inverse function of F and the coordinates (u, v). This method is faster than the previous one, but has some drawbacks. First, it cannot generate desired equal-width shapes. Second, the inverse function might be difficult to derive. Furthermore, different parameterizations and different ways of parameter enumerations might yield different results, as shown in Fig. 4.2.

4.2 Proposed Method

In the cell value accumulation step where pixels contributing to Hough cell values are determined, it is desired, as described previously, to develop a method for detecting pixels of an equal-width shape both centered at the *thin* curve $F(u, v) = 0$ and with an equal width W everywhere on the curve.

Given a pixel P with coordinates (u, v), two cases can be identified. One is that $F(u, v) < 0$, where the coordinates (u', v') of the closest pixel P' on the curve $F = 0$, as

depicted in Fig. 4.3, can be estimated by the use of the direction of the gradient vector at (u, v) to be

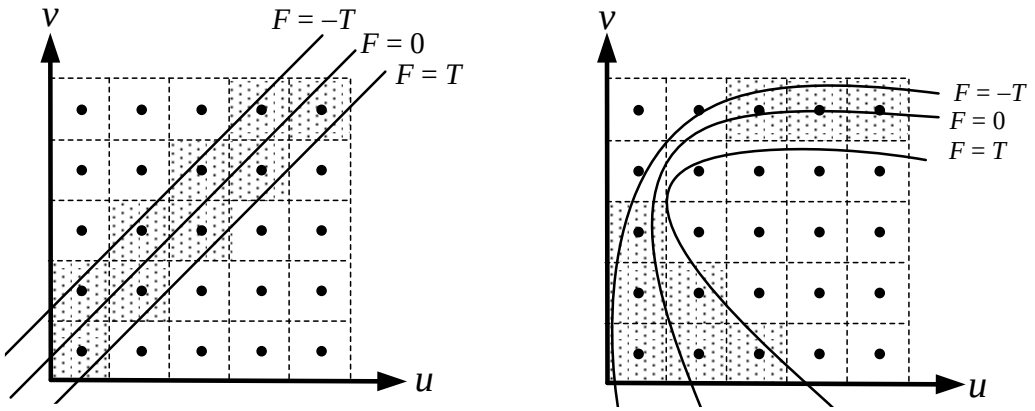


Fig. 4.1 Pixels (marked in dotted blocks) contributing to a Hough cell value when detecting (a) a line, and (b) a curve.

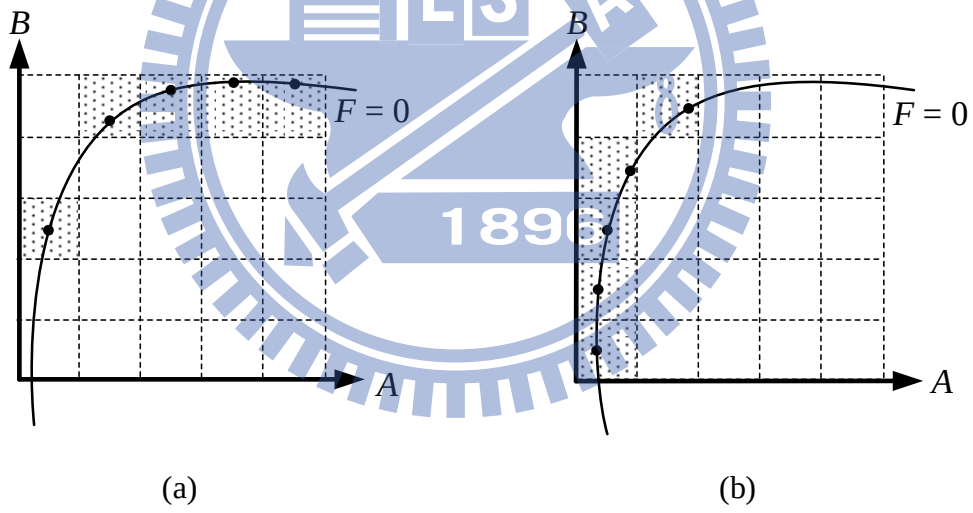


Fig. 4.2 Hough cells (dotted blocks) with values contributed by an edge pixel when enumerating parameter (a) A, and (b) B.

$$(u', v') = (u, v) + d \cdot \frac{\nabla F(u, v)}{\|\nabla F(u, v)\|}, \tag{6}$$

where d is an unknown distance. Then, the function value $F(u', v')$ at (u', v') can be linearly estimated by the use of the first-order directional derivative to be

$$F(u',v') = F(u,v) + \nabla F(u,v) \cdot \left(d \cdot \frac{\nabla F(u,v)}{\|\nabla F(u,v)\|} \right). \tag{7}$$

Also, because pixel P' with coordinates (u', v') is on the curve, we have $F(u', v') = 0$, so that (7) implies that

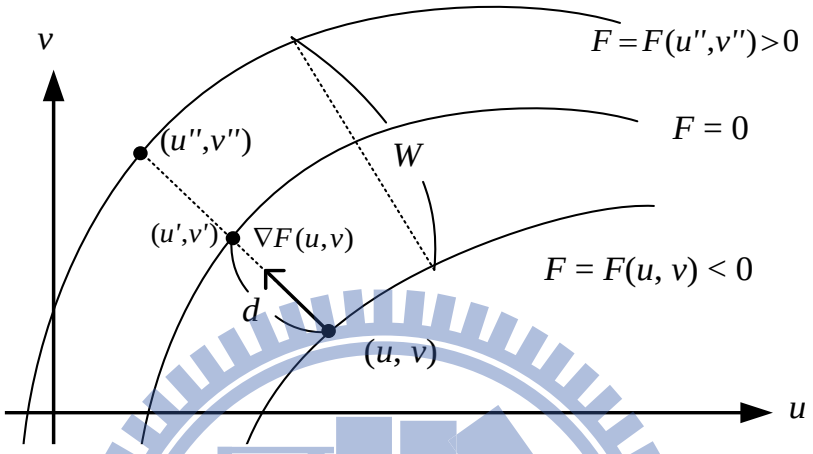


Fig. 4.3 Illustration of proposed dynamic thresholding method.

$$F(u,v) + \nabla F(u,v) \cdot \left(d \frac{\nabla F(u,v)}{\|\nabla F(u,v)\|} \right) = 0,$$

or equivalently, that

$$d = -F(u,v) \cdot \left(\frac{\|\nabla F(u,v)\|}{\nabla F(u,v) \cdot \nabla F(u,v)} \right) = \frac{-F(u,v)}{\|\nabla F(u,v)\|}. \tag{8}$$

Then, as illustrated in Fig. 4.3, the inequality $d < W/2$ can be used to determine whether a pixel with coordinates (u, v) is within the equal-width curve shape. This inequality, when combined with (8), leads to

$$d = \frac{-F(u,v)}{\|\nabla F(u,v)\|} < \frac{W}{2},$$

or equivalently, to

$$-F(u,v) < \|\nabla F(u,v)\| \cdot \frac{W}{2}. \tag{9}$$

In the other case that $F(u, v) > 0$, similarly, the coordinates (u', v') of the closest point P' on the curve $F = 0$ can be linearly estimated by the *negated* gradient vector $-\nabla F(u, v)$, so that coordinates (u', v') can be expressed as

$$(u',v') = (u,v) - d\frac{\nabla F(u,v)}{\|\nabla F(u,v)\|}.$$

Then, following similar derivations, we can get

$$F(u,v) < \|\nabla F(u,v)\| \cdot \frac{W}{2}. \tag{10}$$

The two inequalities of (9) and (10) can be combined to get

$$|F(u,v)| < \|\nabla F(u,v)\| \cdot \frac{W}{2}, \tag{11}$$

which is of the form of the inequality of (5) used in the *constant thresholding* method. But differently, the threshold T can now be taken to be $(\|\nabla F(u, v)\| \cdot W)/2$ whose value can be dynamically determined for pixels with different coordinates (u, v) as well as for Hough cells with different parameters related to the function F, in order to detect a desired equal-width curve shape in the image space.

Theoretically, the dynamic thresholding method proposed above is based on linear approximation. Accordingly, the estimated function value $F(u', v')$ will become inaccurate when the desired curve width W becomes large. However, since the curve width is used to overcome small errors in the input data, the width W may be taken to be a small number. So, the proposed method is expected to yield good results in most applications.

4.3 Experimental Results

In this section, the validity, effectiveness, and robustness of the proposed dynamic thresholding method for the Hough transform are shown by comparing the proposed method with four other methods as listed in Table 4.1 for detecting four different types of shapes as listed in Table 4.2. Among these shapes, The curve of a space line projected on an image taken by the omni-camera can be described by [32]:

$$C_1u^2 + C_2uv + C_3v^2 + C_4u + C_5v + C_6 = 0, \quad (12)$$

where the coefficients are

$$C_1 = A^2 - B^2\left(C_7{}^2 - 1\right); \qquad C_2 = 2A\sqrt{1 - A^2 - B^2};$$
$$C_3 = 1 - A^2 - C_7{}^2B^2; \qquad C_4 = 2ABC_7 f;$$
$$C_5 = 2BC_7\sqrt{1 - A^2 - B^2}\, f; \quad C_6 = B^2 f^2; \text{ and } \quad C_7 = \frac{\varepsilon^2 + 1}{\varepsilon^2 - 1}.$$

The range of the two parameters A and B are both taken to be from –1 to 1 and the Hough space dimension is chosen to be 64×64.

Table 4.1 Used Hough transform algorithms.

No.	accumulation method	described in…
(H1)	the proposed method	Section 4.2
(H2)	constant threshold 1.0	Section 4.1
(H3)	constant threshold 100	Section 4.1
(H4)	examining each cell	Section 4.1
(H5)	inverse function	Section 4.1

Table 4.2 Shapes used in experiments.

shape	equation	cell size
(S1) line	$v = Au + B$	A: 0.1; B: 10
(S2) circle / ellipse	$\left(\frac{u-u_0}{A}\right)^2 + \left(\frac{v-v_0}{B}\right)^2 = 1$	u_0: 10; v_0: 10; A: 10; B: 10
(S3) sine and cosine	$v = A\sin(Bu) + C\cos(Du)$	A: 10; B: 0.1; C: 10; D: 0.1
(S4) space lines in omni-images	described in (12)	A: 0.1; B: 0.1

The pixels contributing to the peak in the Hough space for detecting each of the four shapes listed in Table 4.2 using each of the five methods listed in Table 4.1 are drawn in Fig. 4.4. Recalling that these pixels are desired to form an equal-width shape,

one can see that the proposed method yields the best results for both simple and complex shapes as shown in the leftmost column in the figure, and that the conventional methods cannot generate equal-width curves, especially when detecting complicated shapes.

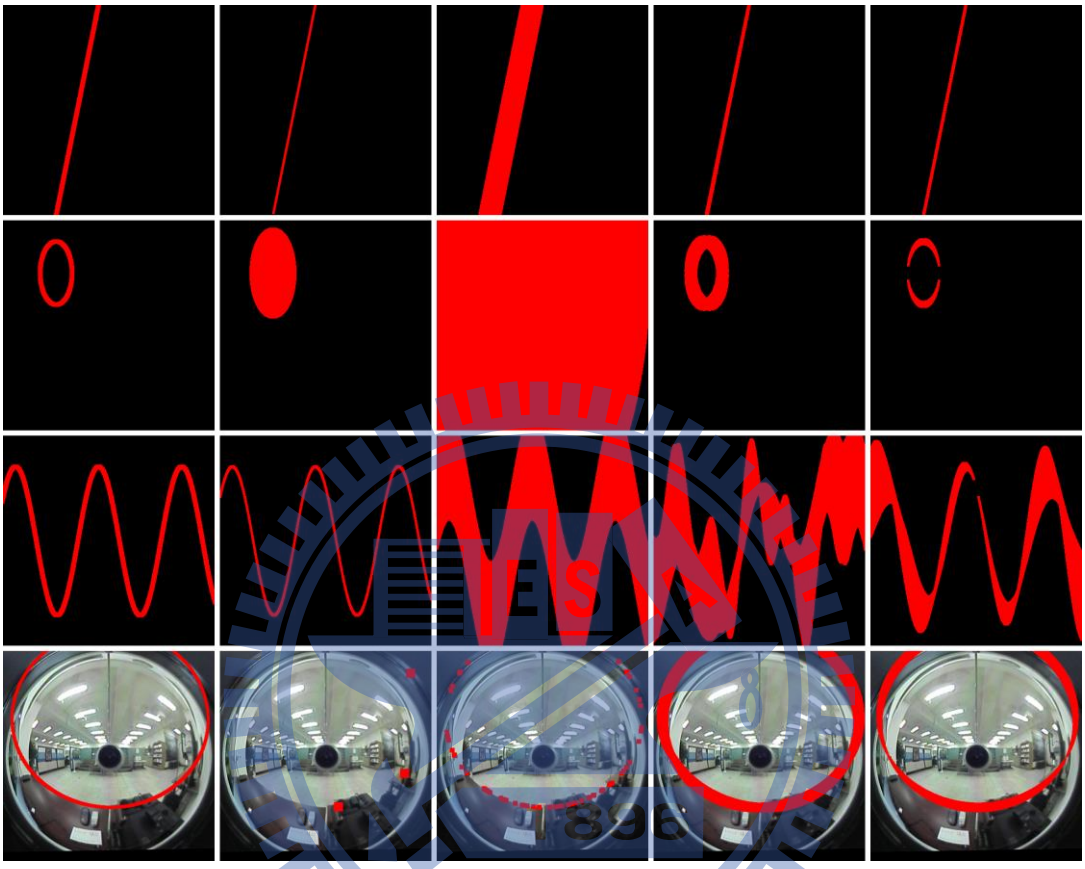


Fig. 4.4 Pixels (in red) contributing to peak cell value. From left to right: results of algorithms H1 to H5, and from top to bottom: results for shapes S1 to S4. The proposed method yields the best results as shown in the leftmost column.

To test the robustness of each different cell-value accumulation method, the ground-truth curve is first drawn on an image. Then, the pixels on the curve are perturbed within a circle with a diameter of 5 pixels to generate *curve pixels* with small errors. Also, 1% pixels of the entire image are randomly noisified as *noise pixels*. The true positive rate (TPR) and the false positive rate (FPR) for each of the five methods are calculated accordingly respectively by:

$$TPR = \frac{\text{\# of curve pixels contributing to the peak cell value}}{\text{total \# of the original curve pixels}};$$

$$FPR = \frac{\text{\# of noise pixels contributing to the peak cell value}}{\text{total \# of the noise pixels}}.$$

As stated previously, it is desired that the *curve pixels* all contribute to the peak-value accumulation, so a high TPR is desired. Contrarily, *noise pixels* which make such contributions should be as few as possible, so a low FPR is desired. As can be seen in Fig. 4.5, the proposed method yields very high TPRs and very low FPRs for all the four types of shapes, showing its robustness; and this is not the case for each of the other four methods.

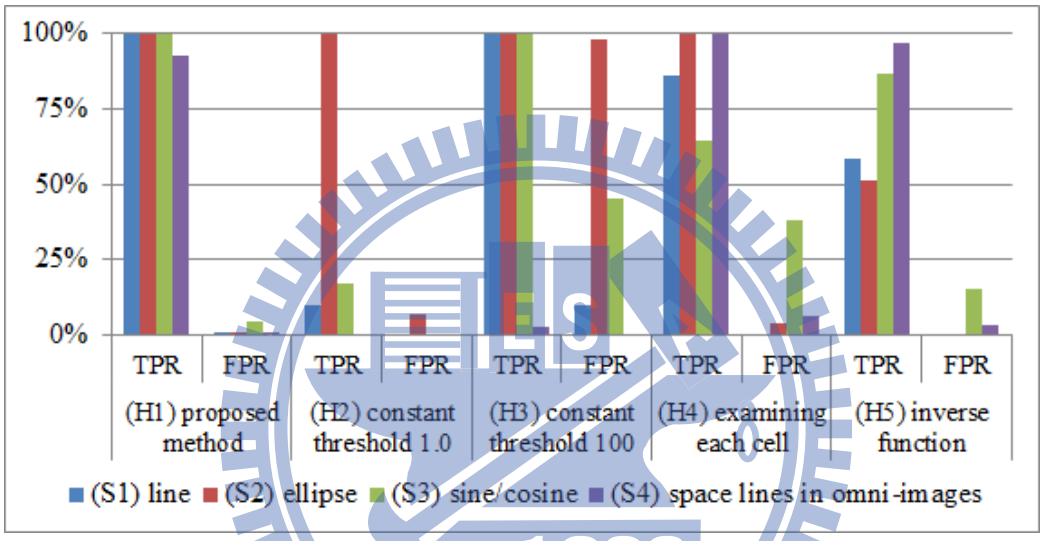


Fig. 4.5 TPR and FPR of five Hough transform algorithms for detecting four types of shapes. The proposed method yields high TPRs and low FPRs for all the shapes, and the others do not.

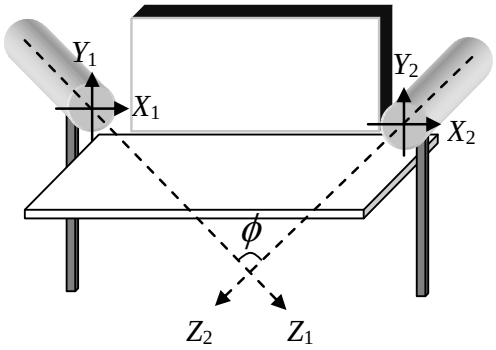
Chapter 5 Binocular Omni-vision Systems with an Automatic Adaptation capability to Any System Setup for 3D Vision Applications

In this chapter, the proposed 3D omni-vision system, which uses two omni-cameras and has a capability of *automatic adaptation* to *any* system setup for convenient *in-field* uses is described. Specifically, the proposed vision system, as shown in Fig. 5.1, consists of two omni-cameras facing the user's activity area. Each camera is affixed firmly to the top of a rod, forming an *omni-camera stand*, with the camera's optical axis adjusted to be horizontal (i.e., parallel to the ground). The cameras are allowed to be placed *freely* in the environment at *any* location in *any* orientation, resulting in an *arbitrary system setup*. Then, by the use of space line features in environments, the proposed vision system can adapt *automatically* to the arbitrarily-established system configuration by just asking the user to stand still for a little moment in the middle region of the activity area in front of the two cameras. After this adaptation operation, 3D data can be computed *correctly* and *precisely*.

In the following, an overview of the adaptation process is described in Section 5.1, and the details of the proposed techniques for use in the system are presented in Sections 5.2 through 5.4, followed by experimental results present in Section 5.5.

5.1 Overview of the Adaptation Process

After the omni-cameras are placed by a user, the poses of the cameras are unknown to the vision system. To derive the unknown poses of the cameras, an in-field adaptation process is proposed in this study, which uses line features in environments to automatically compute the locations and orientations of the cameras. In this stage, a user with a known height is asked to stand in the middle region in front of the two cameras to complete the adaptation. A sketch of the adaptation process is described in the following. To simplify the expressions, we will call the left and right cameras as Cameras 1 and 2, and their camera coordinate systems as CCSs 1 and 2, respectively.



(a)

(b) (c)

Fig. 5.1 Configuration and an illustration of the usage of proposed system. (a) An illustration. (b) Real system used in this study. (c) An omni-image of a user wearing a finger cot (marked as red).

Algorithm 5.1. *Sketch of the proposed adaptation process.*

Step 1. Place the two camera stands at proper locations with appropriate orientations to meet the requirement of the application activity.

Step 2. Perform the following steps to calculate the included angle ϕ between the two optical axes of the cameras as shown in Fig. 5.1(a).

2.1. Capture two omni-images I_1 and I_2 of the application activity environment with Cameras 1 and 2, respectively.

2.2. Detect space line features L_i in omni-image I_1 using the Hough transform technique proposed previously.

2.3. Detect space line features R_i in omni-image I_2 similarly.

2.4. Calculate angle ϕ using the detected line features L_i and R_i in a way as

proposed in Section 5.3.

Step 3. Perform the following steps to calculate the locations and orientations of the two cameras.

3.1. Ask a user to stand in the middle region in front of the two omni-cameras and take two images of the user.

3.2. Extract from the acquired images a pre-selected feature point on the user's body, and compute the respective orientations of the two cameras, as described in Section 5.4.

3.3. Detect the user's head and foot in the images, compute the in-between distance up to a scale, and use the distance as well as the corresponding known height of the user to calculate the locations of the cameras, as described in Section 5.4.

Via the above algorithm, the meaning of *system adaptation*, which is the main theme of this study, can be made clearer now: only with the input of the knowledge of the user's height (see Step 3.3), the proposed system can infer the required values of the cameras' locations and orientations for use in computing the 3D data of space points. This is *not* the case when using a conventional stereo vision system with two cameras in which the configuration of the cameras are *fixed* with their orientations and baseline *unchangeable*. This merit of the proposed system makes it easy to conduct system setup in any room space by any people for more types of applications, as mentioned previously.

5.2 Space Line Detection in Omni-images

The structure of omni-cameras used in this study and the associated coordinate systems are defined as shown in Fig. 5.2. An omni-camera is composed of a perspective camera and a hyperboloidal-shaped mirror. The geometry of the mirror shape can be described in the camera coordinate system (CCS) as:

$$\frac{(Z-c)^2}{a^2}-\frac{X^2+Y^2}{b^2}=1,\ a^2+b^2=c^2,\ Z<c. \tag{13}$$

The relation between the camera coordinates (X, Y, Z) of a space point P and the image coordinates (u, v) of its corresponding projection pixel p may be described [53] as:

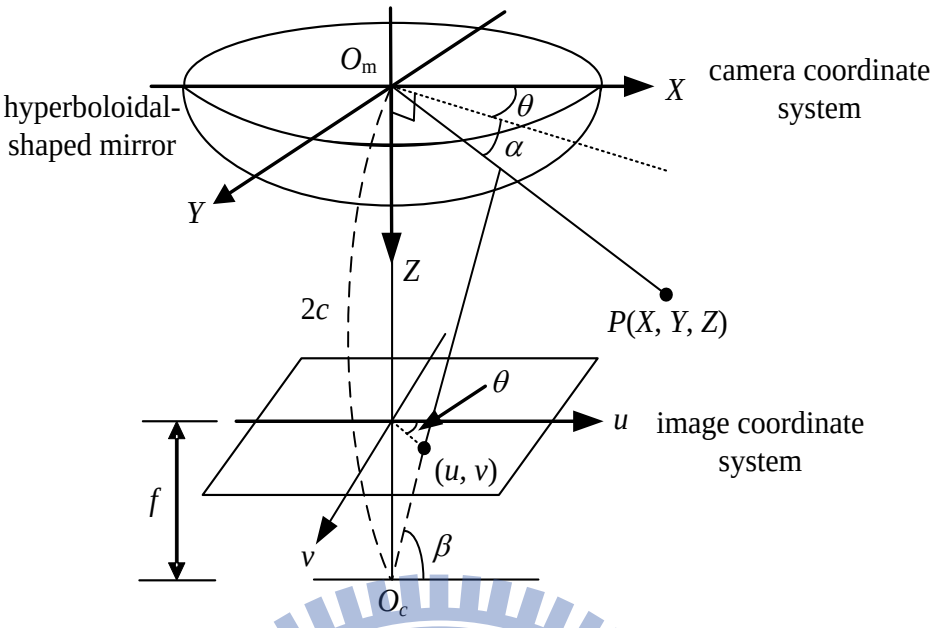


Fig. 5.2 Camera and hyperboloidal-shaped mirror structure.

$$\tan\alpha = \frac{Z}{\sqrt{X^2+Y^2}} = \frac{(\varepsilon^2+1)\sin\beta - 2\varepsilon}{(\varepsilon^2-1)\cos\beta}; \tag{14}$$

$$\cos\beta = \frac{r}{\sqrt{r^2+f^2}};\quad \sin\beta = \frac{f}{\sqrt{r^2+f^2}};\ r = \sqrt{u^2+v^2}, \tag{15}$$

where ε is the eccentricity of the mirror shape with its value equal to c/a, and α and θ are the *elevation* and *azimuth angles* of P, respectively. The azimuth angle θ can be expressed in terms of the image and camera coordinates as

$$\cos\theta = \frac{X}{\sqrt{X^2+Y^2}} = \frac{u}{\sqrt{u^2+v^2}};\ \sin\theta = \frac{Y}{\sqrt{X^2+Y^2}} = \frac{v}{\sqrt{u^2+v^2}}. \tag{16}$$

Now, given a space line L, we can construct a plane S which goes through L and the origin O_m of a CCS as shown in Fig. 5.3. Let $N_S = (l, m, n)$ denote the normal vector of S. Then, any point $P = (X, Y, Z)$ on L satisfies the following plane equation:

$$N_S \cdot P = lX + mY + nZ = 0. \tag{17}$$

where "$\cdot$" denotes the inner-product operator. Combining (17) with (14) and (16), we get

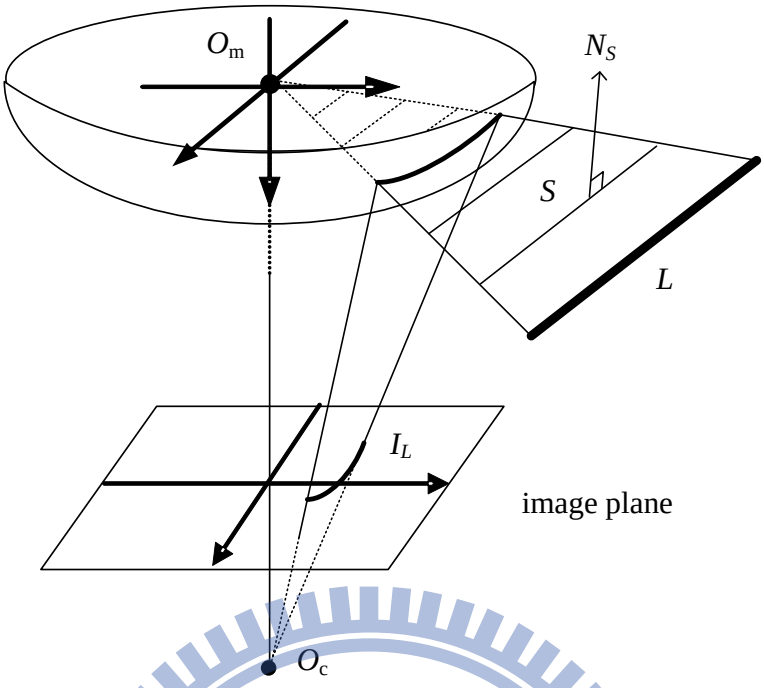


Fig. 5.3 Illustration of a space line L projected on an omni-image as I_L.

$$lR\cos\theta + mR\sin\theta + nR\tan\alpha = 0, \tag{18}$$

where $R = \sqrt{X^2+Y^2}$. Dividing (18) by $R/\sqrt{l^2+m^2+n^2}$ leads to

$$\frac{l\cos\theta}{\sqrt{l^2+m^2+n^2}} + \frac{m\sin\theta}{\sqrt{l^2+m^2+n^2}} + \frac{n\tan\alpha}{\sqrt{l^2+m^2+n^2}} = 0,$$

which can be transformed into the following form

$$A\cos\theta + \sqrt{1-A^2-B^2}\sin\theta + B\tan\alpha = 0 \tag{19}$$

with the two parameters A and B defined as

$$A = \frac{l}{\sqrt{l^2+m^2+n^2}}, \quad B = \frac{n}{\sqrt{l^2+m^2+n^2}}. \tag{20}$$

Accordingly, the normal vector N_S of plane S, originally being (l, m, n), can now be expressed alternatively as

$$N_S = (A,\ \sqrt{1-A^2-B^2},\ B). \tag{21}$$

It is assumed that $m \geq 0$ in (19) and (21) above. In case that $m < 0$, we may consider $N_S = (-l, -m, -n)$ instead, which also represents the same space plane S. Also, it can be seen from (20) that, parameters A and B satisfy the constraint $A^2 + B^2 \leq 1$, implying that the Hough space is of a circle shape.

The parameters A and B are used in the Hough transform to detect space lines in omni-images. These two parameters are skillfully defined in (20), leading to several advantages. First, removals of vertical space lines can be easily achieved by ignoring periphery regions as described later in this section. Next, since the possible values of A and B range from –1 to 1, the size of the Hough space is *fixed* within this range. This is a necessary property in order to use the Hough transform technique, and is an improvement on a previous work [32]. Also, the parameters A and B are used directly to describe the directional vector of the space line L. Hence, one may divide the Hough space into more cells to yield a better precision.

Combining (19) with (14) through (16), we can derive a *conic section* equation to describe the projection of a space line L onto an omni-image as follows:

$$F_{A,B}(u,v) = C_1u^2 + C_2uv + C_3v^2 + C_4u + C_5v + C_6 = 0, \qquad 22$$

where the coefficients C_1 through C_6 are:

$$C_1 = A^2 - B^2\left(C_7^{\,2} - 1\right); \qquad C_2 = 2A\sqrt{1 - A^2 - B^2};$$

$$C_3 = 1 - A^2 - C_7^{\,2}B^2; \qquad C_4 = 2ABC_7 f;$$

$$C_5 = 2BC_7\sqrt{1 - A^2 - B^2}\, f; \qquad C_6 = B^2 f^2; \qquad C_7 = \frac{\varepsilon^2 + 1}{\varepsilon^2 - 1}.$$

The quadratic formula (22) will be called the *target equation* in the Hough transform subsequently, since the goal of the detection process is to find curves described by it in an omni-image.

We define the Hough space to be two-dimensional with the parameters A and B described previously. Furthermore, we define the *cell support* for a cell at (A, B) in the Hough space as the set of those pixels which contribute to the accumulation of the value of that cell. Let L denote a space line described by the two parameters (A, B). Two properties of cell supports are desirable: (1) the pixels of the projection I_L of L onto the omni-image are all included in the cell support for the cell (A, B); and (2) the pixels not on I_L are not included in this cell support. Furthermore, it is desired that the

shape of the cell support is of a certain *fixed* width and not too "thin," so that (edge) pixels originally belonging to I_L but with small detection errors can still contribute to the cell value. In short, a cell support is desired to be a space line projection with a certain width everywhere along the line, which is called an *equal-width projection curve* hereafter. For this, the technique proposed previously can be utilized in the accumulation task, leading to a more precise and accurate detection results as shown later in some experimental results.

In man-made environments, most lines are either parallel to the floor (which is called *horizontal space lines* hereafter) or perpendicular to the floor (which is called *horizontal space lines*). If we can eliminate vertical space lines from the detection results, the rest of them are much more likely to be horizontal ones which are desired. So, a constraint on the vertical space line is derived for the purpose of removing such lines.

As mentioned earlier, the omni-camera stands are vertically placed on the floor, with the *Y*-axis of the camera coordinate system being a vertical line as depicted in Fig. 5.1(a). As a result, the directional vector v_L of a vertical space line L is just (0, 1, 0). Let S be the space plane going through L and the origin O_m which is at camera coordinates (0, 0, 0). Also, let $N_S = (l, m, n)$ be the normal vector of plane S. By definition, normal vector N_S is perpendicular to v_L, leading to the constraint:

$$N_S \cdot v_L = (l, m, n) \cdot (0, 1, 0) = m = 0. \tag{23}$$

This constraint, when combined with (20), results in the equality $A^2 + B^2 = 1$, which shows subtly that the Hough cells of vertical space lines are located in the periphery region of the *circular* Hough space (as mentioned earlier). As a result, vertical space lines can be easily removed by just ignoring the periphery region of the Hough space. In the proposed method, this is achieved automatically by applying a filter on the Hough space as described next.

After the Hough space is generated, the last thing to do is to extract cells with peak values, called *peak cells*, which represent the detected space lines. The simplest way to accomplish this is to find the cells with large values. However, if we do so to get peak cells like those shown in Fig. 5.4(a), we might get a bad detection result like that shown in Fig. 5.4(b) with many of the detected space lines being too close to one another, from which less useful space lines may be extracted.

To solve this problem, we notice that the line edges in an environment mostly are not so close mutually, meaning that two detected horizontal lines usually are separated for a certain distance. This in turn means that extracted peak cells should not be too close to one another. To find the peak cells which are not too close to each other, a filter is applied on the Hough space:

$$\frac{1}{25}\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}. \tag{24}$$

Then, we extract peak cells by choosing the cells with large values in the filtered Hough space to yield a better detection result, as shown by Figs. 5.4(c) and 5.4(d).

Importantly, it is noted that when applying the filter to the Hough space, one of the side effects is the removal of the periphery region. This is a desired property mentioned previously: the removal of the periphery region is equivalent to the removal of vertical space lines. Thus, expectedly we can get more horizontal lines as desired. To sum up, we have proposed a new method to detect horizontal space lines in omni-images, with several novel techniques also proposed in this section to improve the detection result. The proposed method for horizontal space line detection is summarized as an algorithm in the following.

Algorithm 5.2. *Detection of horizontal lines in the form of conic sections.*

Input: an omni-image I.

Output: 2-tuple values (A_i, B_i) as defined in (20) which describe detected horizontal space lines in I.

Step 1. Extract the edge points in I by an edge detection algorithm.

Step 2. Set up a 2D Hough space H with two parameters A and B, and set all the initial cell values to be zeros.

Step 3. For each detected edge point at coordinates (u, v) and each cell C with parameters (A, B), if (u, v, A, B) satisfies the dynamic thresholding constraint, increase the value of C by one.

Step 4. Apply the filter described by (24) to Hough space H, choose those cells with maximum values, and take their corresponding parameters (A_i, B_i) as output.

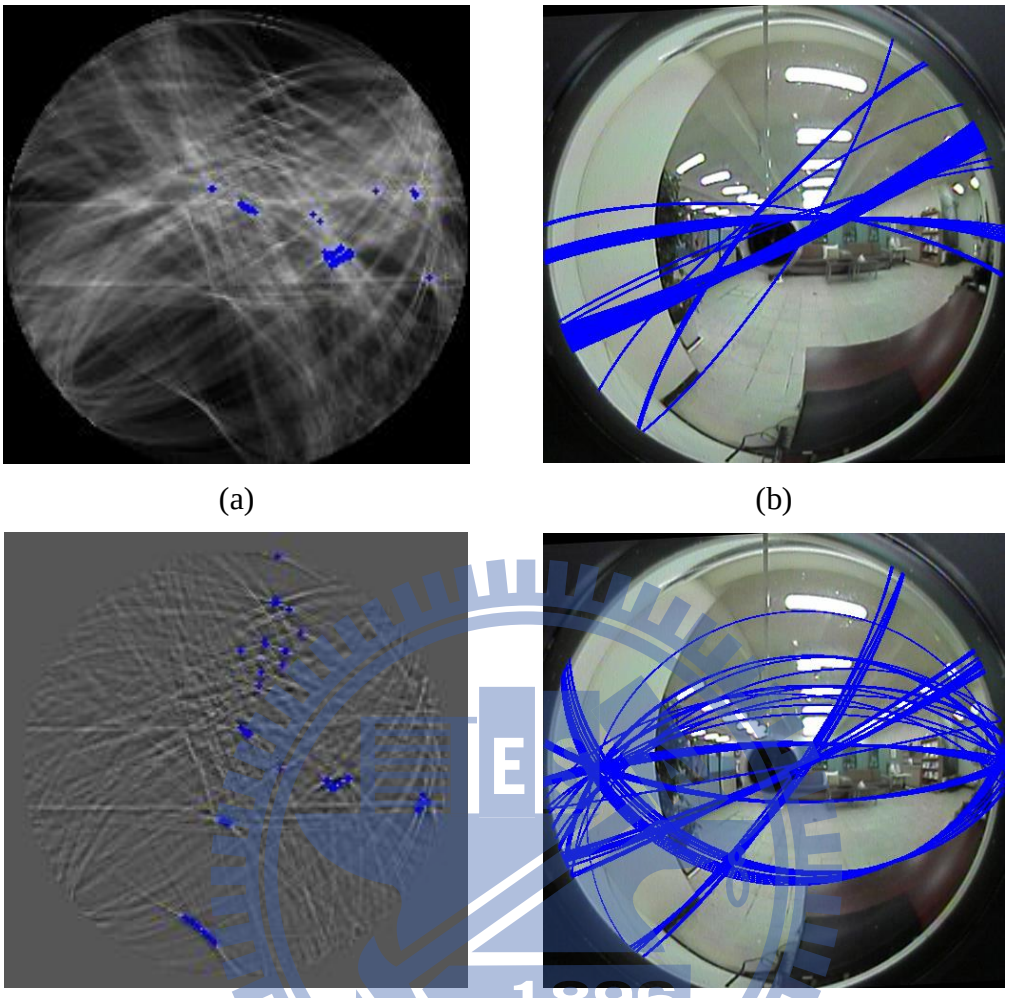


Fig. 5.4 Comparison of traditional peak cell extraction method and proposed one. (a) Hough space. (b) 50 detected space lines using traditional method. (c) Post-processed Hough space. (d) 50 detected space lines using proposed method.

5.3 Calculation of Included Angle between Two Cameras' Optical Axes Using Detected Lines

In the proposed vision system, the omni-cameras are mounted on two vertical stands with the optical axes being parallel to the floor plane as mentioned previously, but the cameras' optical axes are allowed to be non-parallel, making an included angle ϕ as depicted in Fig. 5.1(a). To accomplish the 3D data computation work under an arbitrary system setup, the included angle ϕ must be calculated first. A method to calculate the angle ϕ using a *single manually chosen* horizontal space line is proposed

first. However, in order to conduct the adaptation process automatically, we have to calculate the angle ϕ using *multiple automatically extracted* horizontal space lines. To achieve this, a novel statistical-based method is proposed next in this section, which utilizes *all* the detected space lines from the two omni-images taken with the cameras.

Firstly, a method to calculate the angle ϕ between the two cameras' optical axes is proposed, using a single horizontal space line L in the environment. Let (A_1, B_1) be the parameters corresponding to line L in an omni-image taken with Camera 1, $v_L = (v_x, v_y, v_z)$ be the directional vector of L in CCS 1, and S_1 be the space plane going through line L and the origin of CCS 1. The normal vector of S_1 can be derived, according to (21), to be

$$n_1 = (A_1, \sqrt{1 - A_1^2 - B_1^2}, B_1). \tag{25}$$

Since S_1 goes through line L, we get to know that v_L and n_1 are perpendicular, resulting in the following equality:

$$v_L \cdot n_1 = v_x A_1 + v_y \sqrt{1 - A_1^2 - B_1^2} + v_z B_1 = 0. \tag{26}$$

Furthermore, since L, being horizontal, is parallel to the XZ-plane as shown in Fig. 5.1(a), we get another constraint $v_y = 0$. This constraint can be combined with (26) to get

$$v_L = (v_x, v_y, v_z) = (B_1, 0, -A_1). \tag{27}$$

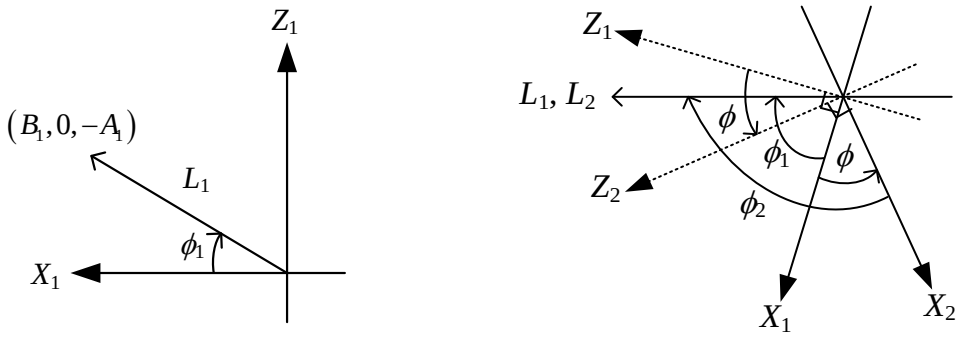
Next, by referring to Fig. 5.5(a), it can be seen that the angle ϕ_1 between the X-axis of CCS 1 and space line L is

$$\phi_1 = \tan^{-1}(-A_1/B_1). \tag{28}$$

Similarly, let (A_2, B_2) be the parameters corresponding to the horizontal space line L in Camera 2. By following the same derivations described above, the angle ϕ_2 between the X-axis of CCS 2 and line L can be derived to be

$$\phi_2 = \tan^{-1}(-A_2/B_2).$$

As depicted in Fig. 5.5(b) where L_1 and L_2 specify identically the single horizontal space line L, the angle ϕ between the two cameras' optical axes can now be computed easily to be



(a) (b)

Fig. 5.5 Illustration of the angles ϕ_1, ϕ_2 and ϕ. (a) The definition of ϕ_1. (b) Relation between ϕ_1, ϕ_2 and ϕ.

$$\phi = \phi_1 - \phi_2 = \tan^{-1}(-A_1/B_1) - \tan^{-1}(-A_2/B_2). \quad (29)$$

Next, to calculate the angle ϕ correctly and reliably from *all* the detected space lines, a statistical-based method is proposed as follows. Recall that horizontal space lines can be detected from an omni-image using Algorithm 5.2. Let L_1 be a space line so detected from the left omni-image with parameters (A_1, B_1), and let L_2 be another detected similarly from the right omni-image with parameters (A_2, B_2). As stated previously, the angle ϕ can be calculated using (29) if the space lines L_1 and L_2 are an identical horizontal space line L in the environment.

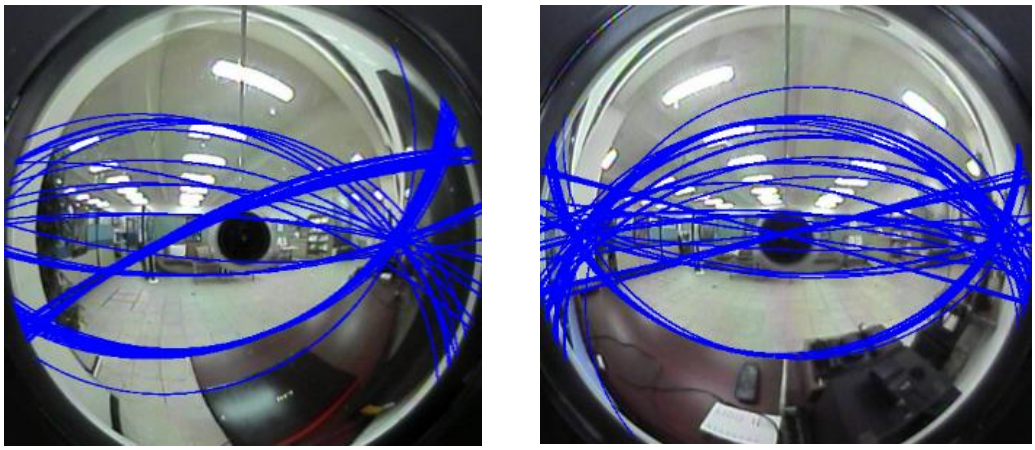
However, the *line correspondence* problem of deciding whether L_1 and L_2 are identical or not is difficult for several reasons, especially for a wide-baseline stereo system like the one proposed in this study. First, the respective viewpoints and viewing fields of the two cameras differ largely. Thus, environment features, like lighting and color, involved in the image-taking conditions at the two far-separated cameras might vary largely as well. Also, the extrinsic parameters of the two cameras are unknown; therefore, the involved geometric relationship is not available for use to determine the line correspondences. To get rid of these difficulties, we propose a novel statistics-based method to reliably find the angle ϕ *without* the need to find such line correspondences.

More specifically, the proposed method makes use of two important properties. First, it is noticed that the correct value of the angle ϕ can still be calculated using (29) even when the two space lines L_1 and L_2 are not an identical one, but are *parallel* to

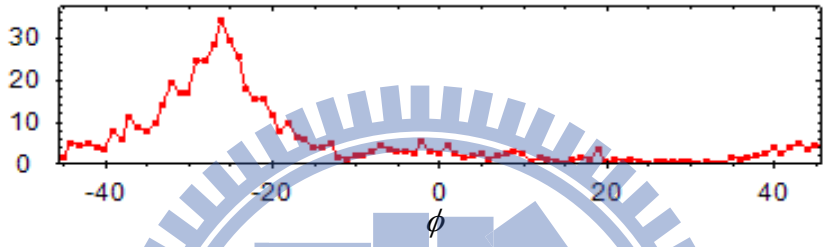
each other. This can be seen from the fact that the angles ϕ_1 and ϕ_2 remain the same if L_1 and L_2 are parallel so that the computed angle ϕ is still correct, as desired. Second, it can be seen that in man-made environments, many of the line edges are parallel to one another in order to make the environment neat and orderly. For example, tables, shelves, and lights are always placed to be parallel to walls and to one another. Combining these two properties, we can conclude that any two detected space lines L_1 and L_2 are very likely to be parallel to each other. Based on this observation, we assume every possible line pair L_1 and L_2 to be parallel, and compute accordingly a candidate value for angle ϕ, where L_1 is one of the space lines detected from the left omni-image, and L_2 is another detected from the right omni-image. Then, we infer a correct value for angle ϕ from the set of all the computed candidate values via a statistical approach based on the concept of "voting."

In more detail, the proposed method is designed to include three main steps. First, we extract space lines from the left omni-image as described in Algorithm 5.2, and denote the line parameters (A, B) of them as l_i. Similarly, we detect space lines from the right omni-image with their parameters denoted as r_j. In addition, we define two *weights* $w(l_i)$ and $w(r_j)$ for l_i and r_j, respectively, to be the cell values in the post-processed Hough space derived in Step 4 of Algorithm 5.2, which represent the *trust measures* of the detected space lines. Then, from each possible pair (l_i, r_j), we calculate a value ϕ_{ij} for angle ϕ using (29), as well as a third weight w_{ij} defined as $w(l_i)\times w(r_j)$. The value w_{ij} may be regarded as the *trust measure* of the calculated angle ϕ_{ij}. Finally, we set up a set of bins, each for a distinct value of ϕ, and for each computed value ϕ_{ij}, we increase the value of the corresponding bin by the weight w_{ij}. After such a weight accumulation work is completed, the bin with the largest value is found out and the corresponding angle ϕ_{ij} is taken as the desired value for angle ϕ.

An experimental result so obtained is shown in Fig. 5.6. In Figs. 5.6(a) and 5.6(b), fifty space lines with parameters l_i and r_j were detected using Algorithm 5.2 from the left and right omni-images, respectively. For each possible pair (l_i, r_j) where $1\le i, j\le 50$, the corresponding angle ϕ_{ij} and weight w_{ij} were calculated and accumulated in bins as described previously. The accumulation result is shown in Fig. 5.6(c) with the maximum occurring at $\phi = -23°$, which is taken finally as the value of angle ϕ.



(a) (b)



(c)

Fig. 5.6 Experimental result of proposed adaptation method for detecting included angle ϕ. (a) and (b) Left/right omni-images, with the detected space lines superimposed on it. (c) Accumulation result with maximum at $\phi = -23°$.

5.4 Proposed Technique for Deriving Camera Poses

The world coordinate system X-Y-Z is defined as depicted in Fig. 5.7. The X-axis goes through the two camera centers O_1 and O_2; the Y-axis is taken to be parallel to the Y-axes of both CCSs; the Z-axis is defined to be perpendicular to the XY-plane; and the origin is defined to be the origin O_1 of CCS 1. It is noted here that, since the two omni-cameras are affixed firmly on the omni-camera stands and adjusted to be of an identical height as described previously, the axes X, Z, X_1, Z_1, X_2, and Z_2 are all on the same plane as illustrated in Fig. 5.7. Since the two omni-cameras are allowed to be placed arbitrarily at any location with any orientation, it is necessary to find the *baseline* D and the *orientation angles* β_1 and β_2 (as defined in Fig. 5.7) in advance to calculate the 3D data of space points. The proposed method of deriving the orientation angles and the baseline are described in the following.

Let the camera coordinates of CCS 1 be denoted as (X_1, Y_1, Z_1), and those of CCS 2 as (X_2, Y_2, Z_2), as shown in Fig. 5.7. As mentioned previously, the two CCSs X_1-Y_1-Z_1 and X_2-Y_2-Z_2 are allowed to be oriented arbitrarily (with Y_1 and Y_2 parallel to each other), and the only knowledge acquired by the proposed system is the angle ϕ between the two optical axes Z_1 and Z_2, which is derived using the detected space lines, as described previously in Section 5.3.

To derive the angles β_1 and β_2, the user is asked to stand in the middle region in front of the two omni-cameras so that a feature point P_{user} on the user's body may be utilized to draw a *mid-perpendicular plane* of the line segment O_1O_2 as shown in Fig. 5.7. Let (X_1, Y_1, Z_1) be the coordinates of P_{user} in CCS 1, and (u_1, v_1) be the corresponding pixel's image coordinates in the left omni-image. From (14) and (16), we have the equality:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \end{bmatrix}^{\mathrm{T}} = \sqrt{X_1^{\,2} + Y_1^{\,2}} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & \tan\alpha_1 \end{bmatrix}^{\mathrm{T}}, \tag{30}$$

where $\cos\theta_1$, $\sin\theta_1$, and $\tan\alpha_1$ are computed from (u_1, v_1) according to (14) and (16). This equality shows that the directional vector between O_1 and P_{user} is ($\cos\theta_1$, $\sin\theta_1$, $\tan\alpha_1$) in CCS 1. An angle ψ_1 is defined on the XZ-plane as illustrated in Fig. 5.7, which can be expressed as $\psi_1 = \tan^{-1}(\tan\alpha_1/\cos\theta_1)$. Similarly, the angle ψ_2 defined on the XZ-plane can be derived to be $\tan^{-1}(\tan\alpha_2/\cos\theta_2)$. Accordingly, we can derive β_1 to be

$$\beta_1 = \psi_1 - \left(\frac{\pi}{2} - \frac{\psi_2 - \psi_1 + \phi}{2} \right) = \frac{\psi_1 + \psi_2 + \phi}{2} - \frac{\pi}{2}, \tag{31}$$

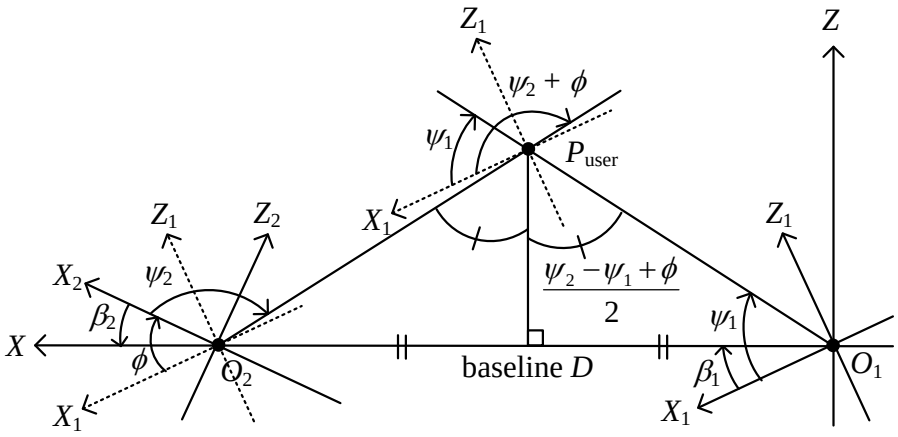


Fig. 5.7 A top-view of the coordinate systems. The baseline D, orientation angles

β_1 and β_2, and a point P_{user} on the user's body are also drawn.

and β_2 is just $\beta_2 = \beta_1 - \phi$. This completes the derivations of the orientation angles β_1 and β_2 of the two cameras.

To compute the baseline D, we make use of a fact about triangulation in binocular computer vision: the 3D data can be determined *up to a scale* without knowing the value of the baseline D [54]. Specifically, within the omni-images taken of the user standing in front of the two cameras as mentioned previously, we extract two points on the head and the feet of the user, respectively. Let P_{head} and P_{foot} denote their real 3D data, respectively. On the other hand, as stated previously, we can compute the 3D data up to a scale of the two points, which we denote as P'_{head} and P'_{foot}, respectively, using triangulation calculations [54] with the baseline D being *one* unit. Then, the relations between the data P_{head}, P_{foot}, P'_{head}, and P'_{foot} can be expressed as

$$P_{head} = D \cdot P'_{head}, \quad \text{and} \quad P_{foot} = D \cdot P'_{foot}, \tag{32}$$

where D is the *actual* baseline value. Let H' be the Euclidean distance between P'_{head} and P'_{foot}; and let H be the real distance between P_{head} and P_{foot}, which is just the known height of the user. Then, the baseline D can finally be computed as $D = H/H'$.

After finding the baseline D, the system parameters are now all adapted. To sum up, the three steps of the proposed adaptation method are briefly described as follows. First, the included angle ϕ between the two optical axes are determined using space line features as discussed in Section 5.3. Then, by asking the user to stand at the middle point in front of the two omni-cameras, the orientation angles β_1 and β_2 of the two cameras are calculated as described in this section. Finally, the baseline D is calculated using the height H of the user as described in this section.

5.5 Experimental Results

Some experimental results are given here to show the adaptation ability under different cameras and environments. Two types of cameras were used, which are perspective cameras and catadioptric omni-cameras, and three different environments were considered, which are a corridor, a hall, and a room, as shown in Figs. 5.8(a) through 5.8(c).

Four different experiments were conducted: Experiment 1 is conducted in the corridor with omni-cameras; Experiment 2 in the hall with omni-cameras; Experiment

3 in the room with omni-cameras; and Experiment 4 also in the room but with perspective cameras. In each experiment, the two cameras were oriented in different angles (i.e., –30°, –15°, 0°, 15°, and 30°). Fifty space line features were first extracted as proposed in Section 5.2. Then, the angle ϕ was automatically calculated using these lines as proposed in Section 5.3. The results are shown in Fig. 5.8(d). The *X*-axis specifies the ground truth of the angle ϕ, and the *Y*-axis specifies the absolute error of the calculated angle ϕ.

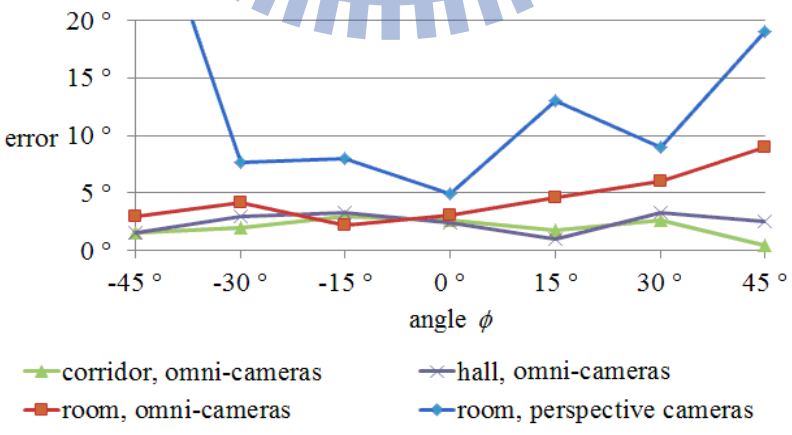
In Experiments 1 and 2, since the lines in the corridor and hall are relatively simple and obvious, the adaptation result is accurate with errors of about 2° as shown by the green and purple curves in Fig. 5.8(c). Also, since we use omni-cameras in these experiments, the lines can still be captured even when the two cameras were oriented with a large angle. Thus, the adaptation result remains accurate when the angle ϕ is large. In Experiment 3, since the space lines in the room are more complicated, the adaptation becomes more difficult. However, since the omni-cameras can capture a large field of view of the environment, a plenty number of space lines can be captured. Therefore, the adaptation result is accurate as well, with errors of about 4° as shown by the red curve in Fig. 5.8(c). In contrast, the adaptation errors are about 10° when perspective cameras were used, as shown by the blue curve in Fig. 5.8(c), and they become unacceptable (larger than 20°) when the included angle ϕ is large. These experimental results show the feasibility of the proposed adaptation methods, as well as the power of the omni-cameras in the automatic adaptation process.

Another series of experiments are conducted to test the *adaptation ability* and the *3D acquisition precision* in the *room* environment. In each experiment in this series, the two cameras were placed at a distance about 180cm to each other, and both were oriented *randomly* within the range of ±40°. After the cameras were set up, two omni-images of the environment were captured as shown, for example, in Figs. 5.9(a) and 5.9(d), respectively, and used to calculate the included angle ϕ according to Step 2 of Algorithm 5.1. Next, a user was asked to stand in the middle region in front of the two cameras, as shown in Figs. 5.9(b) and 5.9(e), to calculate the orientation angles β_1 and β_2 and the baseline *D* according to Step 3 of Algorithm 5.1. After these adaptation tasks were done, a board with 60 landmarks was held by the user, as shown in Figs. 5.9(c) and 5.9(f), to test the precision of the resulting 3D computation.

(a)

(b)

(c)



(d)

Fig. 5.8 Experimental results under different cameras and environments. (a) A corridor. (b) A hall. (c) A room. (d) Adaptation results of angle ϕ.

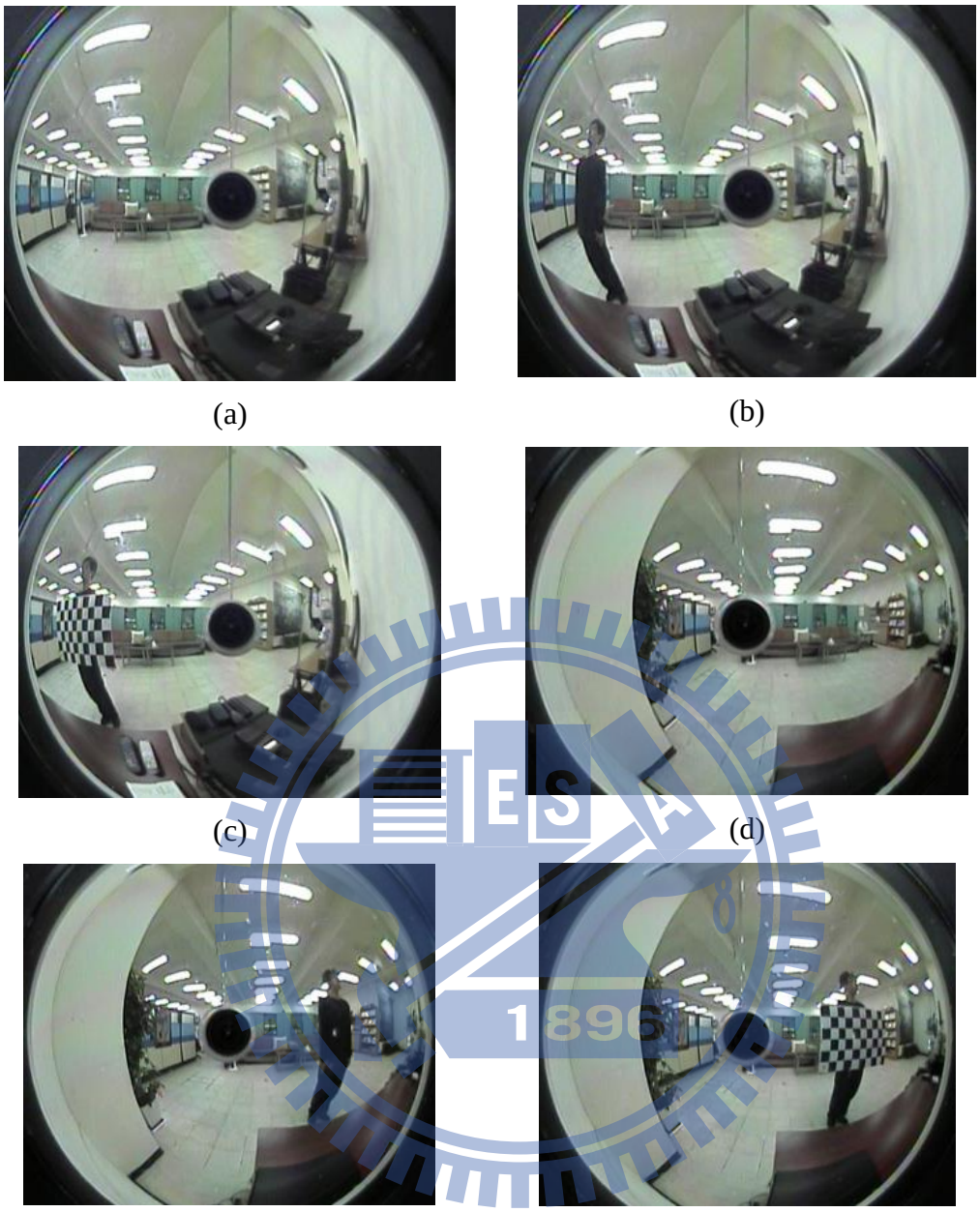


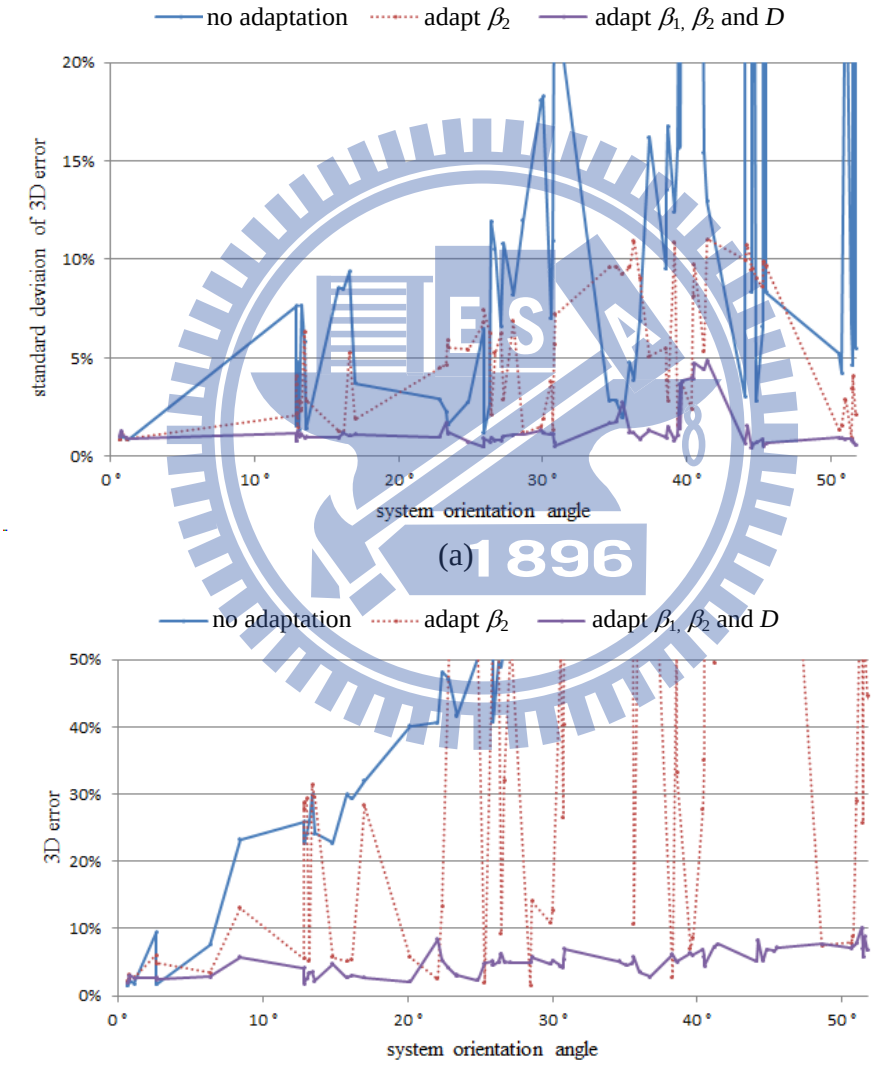
Fig. 5.9 Sample omni-images of an experiment. (a)(d) Taking a shot of the environment to calculate ϕ. (b)(e) A user standing in the middle region in front of the cameras to calculate baseline D and orientation angles β_1 and β_2. (c)(f) A board held by the user to test the 3D computation precision.

In these experiments, three different *degrees of adaptation* were implemented and the corresponding results compared: (1) no adaptation was conducted with the camera orientations and baseline set to be $\beta_1 = \beta_2 = 0°$ and $D = 180$ cm (D is the ground-truth value); (2) the left omni-camera was set up to face forward with the values $\beta_1 = 0°$, $D = 180$cm, and β_2 adapted to be $-\phi$; and (3) all the parameters β_1, β_2, and D were

adapted according to the proposed method. Denoting (X_i, Y_i, Z_i) as the ground-truth location of a landmark point, and (X_i', Y_i', Z_i') as the calculated location, we define the 3D error E of each landmark point as

$$E = \sqrt{(X_i - X_i')^2 + (Y_i - Y_i')^2 + (Z_i - Z_i')^2} \Big/ \sqrt{X_i^2 + Y_i^2 + Z_i^2}\,. \tag{33}$$

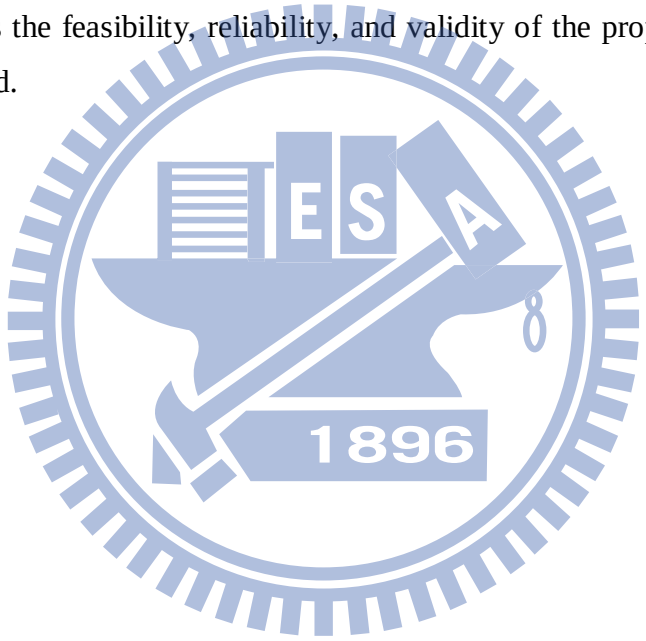
The comparison results are shown in Fig. 5.10 in which the vertical axis specifies the average of the 3D errors, and the horizontal axis specifies the *system orientation angle* which is defined as the *maximum* of the two orientation angles β_1 and β_2.



(b)

Fig. 5.10 Experimental results of three different degrees of adaptations. (a) The 3D errors. (b) The standard deviations of the 3D errors. The proposed adaptation methods yield the best results as shown by the purple curves.

As can be seen from Figs. 5.10(a) and 5.10(b), when no parameter is adapted with the results shown by the blue curve, the 3D errors are seen to become larger as the orientation angle becomes larger, showing the necessity of an automatic system adaptation process. When only the orientation β_2 of the right omni-camera is adapted with the result shown by the red curve, it is observed that the 3D errors are sometimes lower but vary largely. This results from the fact that the left omni-camera is *assumed* to face forward in this case. Thus, if the left omni-camera is *actually* placed to face forward in the experiment, the error measure is lowered; otherwise, the error is large as expected. Finally, when all the parameters β_1, β_2 and D are adapted with the results shown by the purple curve, the 3D errors are lower than 8% even when the system orientation angle is large. This shows the feasibility, reliability, and validity of the proposed system adaptation method.



Chapter 6 Optimal Design and Placement of Omni-cameras in Binocular Vision Systems for Accurate 3D Data Measurement

In this chapter, the problem of automatically designing an optimal stereo vision system using two omni-directional catadioptric cameras to yield the highest 3D data accuracy is studied. Factors of the system configuration considered in the design include camera pose, FOV, and mirror shape. To find the optimal vision system configuration, we derived analytic formula to model the 3D measurement error, which takes into consideration the variations of pixel-quantization precisions and angular resolutions in images by conducting error propagation analysis in the data computation process. The formula is then used in the proposed optimization framework to find the optimal system configurations for different shapes of system setup environments. For regular cases with rectangular cuboid-shaped 3D measurement and camera placement areas, the two proposed fast algorithms can be used to solve the problem, one being bisection-based and relatively slower for deriving the optimal solution; and the other faster using analytic formulas for deriving a sub-optimal solution which is proved to be close to the optimal one in precision. Experimental results of simulations and real application cases show the feasibility of the proposed method.

In the following, the proposed optimization framework to find the optimal system configuration is described in Section 6.1. The proposed formulas to model the catadioptric omni-cameras are derived in Section 6.2. The proposed formulas to *measure of 3D data accuracy* for use in the optimization process is derived in Section 6.3. The proposed three optimization algorithms for finding the optimal system configuration are presented in Sections 6.4 and 6.5. Finally, experimental results and conclusions are given in Sections 6.6 and 6.7, respectively.

6.1 Overview of the Optimization Framework

As illustrated in Fig. 6.1, to facilitate the determination of the *optimal system configuration* of a binocular omni-vision system, which includes the intrinsic parameters, locations, and orientations of the two omni-cameras of the system, for the

purpose of acquiring the most accurate 3D data. Some observable properties of this frame are: (1) if the two omni-cameras are close to each other, the length of the line segment L connecting the two cameras, namely, *the baseline*, will become small, and the computed 3D data accuracy will so be low; (2) if the baseline is very large, the space points in front of the cameras will become *relatively* close to L, and the resulting accuracy will so be low as well; and (3) since the distortion of an image taken by an omni-camera is significant, the resolution also varies significantly in the taken image so that, if a feature point is located in a higher resolution area, the computed 3D data accuracy will become higher, and vice versa. According to these observed facts, it can be seen that optimal system configurations do exist; therefore, it is meaningful to propose an optimization framework, as conducted in this study, for use in finding the optimal system configuration.

The proposed optimization framework, as depicted in Fig. 6.1, includes three main steps. First, an area where the 3D data are measured is specified, called the *3D measurement area*; and an area where the cameras can be placed is also specified, called the *camera placement area*. Then, the *optimal* locations, optical axes, and intrinsic parameters of the two omni-cameras are derived according to one of the three proposed system optimization algorithms presented in later sections. The three optimization algorithms are based on the use of an *analytic formula* indicating the *degree of accuracy* of the computed 3D data, which is derived according to error propagation analysis. The found optimal configuration is just the one with the highest degree of 3D data accuracy, which is shown to the user to tell him/her how to place (and/or design) the cameras.

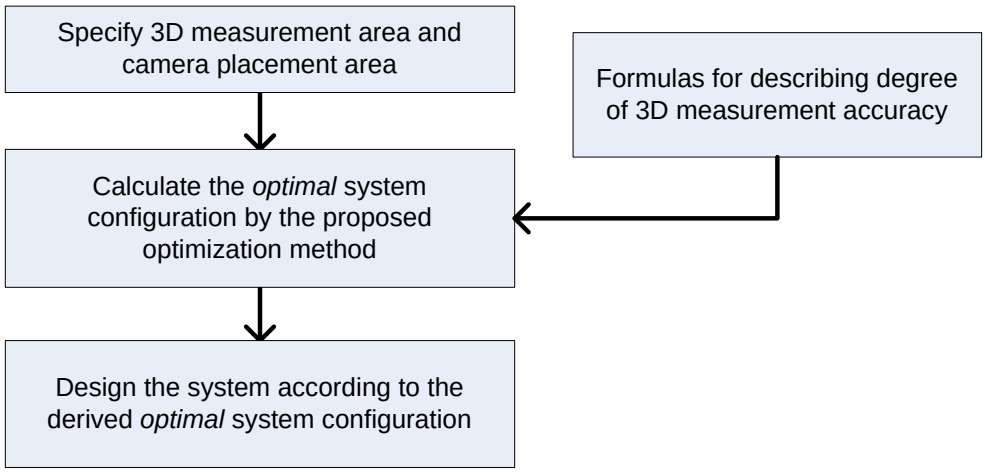


Fig. 6.1 Proposed optimization framework.

6.2 Related Formulas for Omni-cameras

By assuming the perspective camera in the omni-camera is well-calibrated and distortion-free, the only parameter of the perspective camera is its viewing angle $2\tau_{max}$. Also, as specified in the projection equation (2), the eccentricity ε describes all the distortion effect produced by the mirror. Therefore, the intrinsic parameters of an omni-camera to be determined are the viewing angle $2\tau_{max}$ of the perspective camera and the eccentricity ε of the mirror. In the following, we first derive the formula for the eccentricity ε under the assumption that the viewing angle $2\tau_{max}$ of the perspective camera is fixed. Then, we provide a guideline to determine the angle $2\tau_{max}$.

Theorem 6.1. If the viewing angle of the perspective camera is $2\tau_{max}$ and the viewing angle of the omni-camera is $2\phi_{max}$, then the eccentricity ε of the hyperboloidal-shaped mirror is

$$\varepsilon = \frac{\sin\phi_{max} + \sin\tau_{max}}{\sin(\phi_{max} - \tau_{max})}. \tag{34}$$

Proof. According to the projection equation (2), we have

$$(\tan\phi_{max} - \tan\tau_{max})\varepsilon^2 - 2\tan\phi_{max}\sec\tau_{max}\varepsilon + (\tan\phi_{max} + \tan\tau_{max}) = 0. \tag{35}$$

Accordingly, two solutions of ε can be obtained to be

$$\varepsilon = \frac{\tan\phi_{max}\sec\tau_{max} \pm \tan\tau_{max}\sec\phi_{max}}{\tan\phi_{max} - \tan\tau_{max}} = \frac{\sin\phi_{max} \pm \sin\tau_{max}}{\sin(\phi_{max} - \tau_{max})}. \tag{36}$$

The solution ε_1 with the minus sign is proved to be invalid as follows. First, ε_1, as an eccentricity, is larger than one, i.e.,

$$\varepsilon_1 = \frac{\sin\phi_{max} - \sin\tau_{max}}{\sin(\phi_{max} - \tau_{max})} > 1. \tag{37}$$

Next, since the viewing angle $2\phi_{max}$ of the omni-camera is larger than the viewing angle $2\tau_{max}$ of the perspective camera, we have $\pi > \phi_{max} > \tau_{max} > 0$, leading to $\cot(\phi_{max}/2) < \cot(\tau_{max}/2)$, which, according to the cotangent half-angle formula [55], results in

$$\cot\left(\frac{\phi_{\max}}{2}\right)=\frac{\sin\phi_{\max}}{1-\cos\phi_{\max}}<\frac{\sin\tau_{\max}}{1-\cos\tau_{\max}}=\cot\left(\frac{\tau_{\max}}{2}\right), \tag{38}$$

or equivalently,

$$\frac{\sin\phi_{\max}-\sin\tau_{\max}}{\sin(\phi_{\max}-\tau_{\max})}=\varepsilon_1<1, \tag{39}$$

which is a contradiction to (37). Therefore, the other solution described by (34) should be taken as the desired result. □

The effect of choosing different viewing angle $2\tau_{\max}$ of the perspective camera is shown by some images in Fig. 6.2 obtained from simulations with a checkerboard placed in front of the omni-camera, and the hyperboloidal mirror designed according to Theorem 6.1 in such a way that the entire checkerboard can be viewed. As seen from Fig. 6.2, the taken omni-images are not severely influenced by the magnitude of the viewing angle, implying that one may choose freely the viewing angle as long as the camera is distortion-free.

(a) (b) (c)

Fig. 6.2 Simulated omni-images using perspective cameras with different viewing angles: (a) 20°; (b) 40°; and (c) 60°.

To the resolution formula, Baker and Nayer [56] proposed a formula to calculate the resolutions at different pixels in an omni-image as follows. Let *dA* be an infinitesimal area on the image plane near a pixel *p*, which, as illustrated in Fig. 6.3, is the projection of an area in the space described by an infinitesimal solid angle $d\nu$ coming from a point *P*. The resolution of pixel *p* is formulated as

$$R(\varepsilon, f, \phi)=\frac{dA}{d\nu}=\frac{(\varepsilon^2-1)^2(\varepsilon^2+2\varepsilon\cos\phi+1)}{\left[2\varepsilon+(\varepsilon^2+1)\cos\phi\right]^3}f^2, \tag{40}$$

where ε is the eccentricity of the mirror, f is the focal length of the camera, and ϕ is the complementary elevation angle of P.

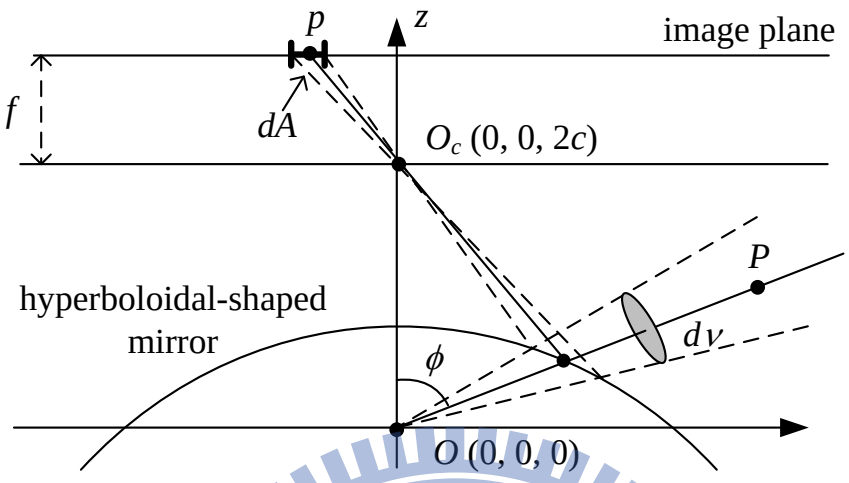


Fig. 6.3 Illustration for defining the resolution at a pixel.

6.3 Formula to Derive the Degree of Accuracy

A criterion function measuring the *degree of accuracy* of the computed 3D data is proposed in this section, for uses in the optimization algorithms proposed in the following sections.

A right-handed world coordinate system X-Y-Z is defined, as shown in Fig. 6.4, in such a way that the camera centers O_1 and O_2 are located at $(-D, 0, 0)$ and $(D, 0, 0)$, respectively, and the XY-plane contains the space point P. An assumption that, the two optical axes a_1 and a_2 lie on the XY-plane, is made in the following derivations, and this assumption is analyzed more thoroughly later in this section.

As depicted in Fig. 6.4, the two axes a_1 and a_2 are defined by the angles α_1 and α_2, respectively. To calculate the 3D data of P, two images are acquired first by the two omni-cameras, and a feature detection process is applied to extract the two pixels p_1 and p_2 corresponding to P in the two omni-images. Then, the complementary elevation angles ϕ_1 and ϕ_2 are derived by (2) using the coordinates of p_1 and p_2, respectively. Finally, the angles θ_1 and θ_2 as depicted in Fig. 6.4 are computed by:

$$\theta_1 = \alpha_1 - \phi_1, \text{ and } \theta_2 = \alpha_2 - \phi_2. \tag{41}$$

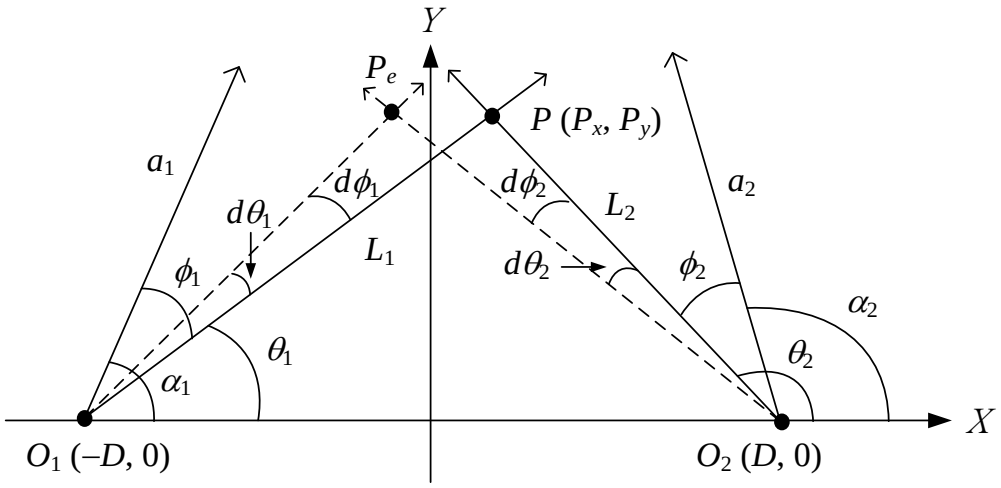


Fig. 6.4 Top-view illustration of triangulation process and error propagation.

Proposition 6.1. As depicted in Fig. 6.4, the position (P_x, P_y) of space point P can be calculated in terms of θ_1 and θ_2 as

$$T(\theta_1, \theta_2) = \left(P_x, P_y\right) = \left(D\frac{\sin(\theta_2+\theta_1)}{\sin(\theta_2-\theta_1)}, 2D\frac{\sin\theta_2\sin\theta_1}{\sin(\theta_2-\theta_1)} \right), \tag{42}$$

and the differentials of P_x and P_y are

$$\begin{bmatrix} dP_x \\ dP_y \end{bmatrix} = \frac{2D}{\sin^2(\theta_2-\theta_1)} \begin{bmatrix} \sin\theta_2\cos\theta_2 d\theta_1 - \sin\theta_1\cos\theta_1 d\theta_2 \\ \sin^2\theta_2 d\theta_1 - \sin^2\theta_1 d\theta_2 \end{bmatrix}. \tag{43}$$

Proof. As depicted in Fig. 6.4, the position of the feature point P can be calculated by two parametric equations

$$P = O_1 + s_1\cdot(\cos\theta_1, \sin\theta_1);\ \ P = O_2 + s_2\cdot(\cos\theta_2, \sin\theta_2) \tag{44}$$

where P, O_1, and O_2 are regarded as 2D *coordinate vectors*, and s_1 and s_2 are unknown parameters. Eq. (44) is equivalent to:

$$P_x = -D + s_1\cos\theta_1 = D + s_2\cos\theta_2;\ \ P_y = s_1\sin\theta_1 = s_2\sin\theta_2, \tag{45}$$

which may be solved to get s_1 and s_2, leading to the results

$$\begin{aligned} P_x &= D\frac{\sin\theta_2\cos\theta_1 + \cos\theta_2\sin\theta_1}{\sin\theta_2\cos\theta_1 - \cos\theta_2\sin\theta_1} = D\frac{\sin(\theta_2+\theta_1)}{\sin(\theta_2-\theta_1)}; \\ P_y &= 2D\frac{\sin\theta_1\sin\theta_2}{\sin\theta_2\cos\theta_1 - \cos\theta_2\sin\theta_1} = 2D\frac{\sin\theta_2\sin\theta_1}{\sin(\theta_2-\theta_1)}. \end{aligned} \tag{46}$$

And, the differentials of P_x and P_y can be derived accordingly to be those described by (43). □

The infinitesimals $d\theta_1$ and $d\theta_2$ are derived by considering the varying pixel-quantization precisions and angular resolutions as follows. As can be seen from Fig. 6.4, if the feature detection process to extract p_1 and p_2 in the two images is inaccurate so that the angles ϕ_1 and ϕ_2 are with errors $d\phi_1$ and $d\phi_2$, respectively, then an inaccurate triangulation result P_e, as depicted in Fig. 6.4, will be produced. Assume that the pixel quantization and feature detection process introduces an error within a small area dA, then, as depicted in Fig. 6.3, the measured angle ϕ will be with an error $d\nu$ related to the *angular resolution* function R. In the sense that $d\nu$ is a 2D solid angle and the angle $d\phi$ is the corresponding 1D angle, and under the assumption that the back-projected cone forming by $d\nu$ is circular, the value of $d\phi$ can be estimated by

$$d\phi = \pm\sqrt{d\nu} = \pm\sqrt{dA/R}\,, \tag{47}$$

in which the term $d\nu$ is expressed by the resolution formula (40). Also, by taking the differentiations of the equations in (41), one can get a relation between $d\phi$ and $d\theta$ as $d\theta = -d\phi$, which, after being combined with (47), leads to

$$d\theta = \pm\sqrt{dA/R}\,. \tag{48}$$

Accordingly, the errors $d\theta_1$ and $d\theta_2$ can be derived to be

$$d\theta_1 = \pm\sqrt{\frac{dA}{R(\varepsilon_1, f_1, \phi_1)}}\,, \text{ and } d\theta_2 = \pm\sqrt{\frac{dA}{R(\varepsilon_2, f_2, \phi_2)}}\,. \tag{49}$$

To assess the 3D measurement error of a feature point P, an error function $E(P)$ is proposed in the following, and the degree of accuracy of the point P is defined as $-E(P)$ in the sequel.

Theorem 6.2. With reference to Fig. 6.4, when the triangulation process yields an imprecise point P_e due to small errors $d\theta_1$ and $d\theta_2$, the 3D measurement error $E(P)$, which is the distance between the actual point P and the measured point P_e, is

$$E(P) = \max\left(E_1(P), E_2(P)\right) \tag{50}$$

where

$$E_1(P) = \frac{\sqrt{G_1^{\,2}(P) + 2G_1(P)G_2(P)\cos(\theta_2 - \theta_1) + G_2^{\,2}(P)}\sqrt{dA}}{\sin(\theta_2 - \theta_1)},$$

$$E_2(P) = \frac{\sqrt{G_1^{\,2}(P) - 2G_1(P)G_2(P)\cos(\theta_2 - \theta_1) + G_2^{\,2}(P)}\sqrt{dA}}{\sin(\theta_2 - \theta_1)},$$

$$G_1(P) = \frac{\left|\overline{O_1P}\right|}{\sqrt{R(\varepsilon_1, f_1, \phi_1)}}, \text{ and } \quad G_2(P) = \frac{\left|\overline{O_2P}\right|}{\sqrt{R(\varepsilon_2, f_2, \phi_2)}}. \tag{51}$$

Proof. The measurement error E, which by definition is the distance between P and P_e, may be computed from (43) to be

$$E(P) = \|dT\| = \sqrt{(dP_x)^2 + (dP_y)^2}$$

$$= \frac{\sqrt{\left(\frac{2D\sin\theta_2 d\theta_1}{\sin(\theta_2 - \theta_1)}\right)^2 - 2\cos(\theta_2 - \theta_1)\frac{2D\sin\theta_2 d\theta_1}{\sin(\theta_2 - \theta_1)}\frac{2D\sin\theta_1 d\theta_2}{\sin(\theta_2 - \theta_1)} + \left(\frac{2D\sin\theta_1 d\theta_2}{\sin(\theta_2 - \theta_1)}\right)^2}}{\sin(\theta_2 - \theta_1)}. \tag{52}$$

From Fig. 6.4, it can be derived, from the law of sines, that

$$\frac{2D}{\sin(\theta_2 - \theta_1)} = \frac{\left|\overline{O_1P}\right|}{\sin\theta_2} = \frac{\left|\overline{O_2P}\right|}{\sin\theta_1}. \tag{53}$$

By combining (52) and (53), we get

$$E(P) = \frac{\sqrt{(\left|\overline{O_1P}\right| d\theta_1)^2 - 2\cos(\theta_2 - \theta_1)(\left|\overline{O_1P}\right| d\theta_1)(\left|\overline{O_2P}\right| d\theta_2) + (\left|\overline{O_2P}\right| d\theta_2)^2}}{\sin(\theta_2 - \theta_1)}, \tag{54}$$

which, when combined with (49), leads to (50) and (51). □

6.4 Fast Configuration Optimization for Regular Cases

The optimization framework proposed previously can deal with general cases, in which the 3D measurement area and the camera placement area may both be of irregular shapes, and the two used perspective cameras may be different from each other. However, in regular indoor vision systems (called the *regular cases*), the two perspective cameras are of the same type, and the 3D measurement area and the camera placement area can be specified by two rectangular cuboids as illustrated in

Fig. 6.5. In the following, the formal definition of the optimization problem for such regular cases is first derived. Then, the derivation of the proposed optimization algorithm for generating the corresponding *optimal* system configuration is proposed, followed by another *sub-optimal* but *analytic* optimization method, which is shown additionally there to be a good approximation to the optimal solution.

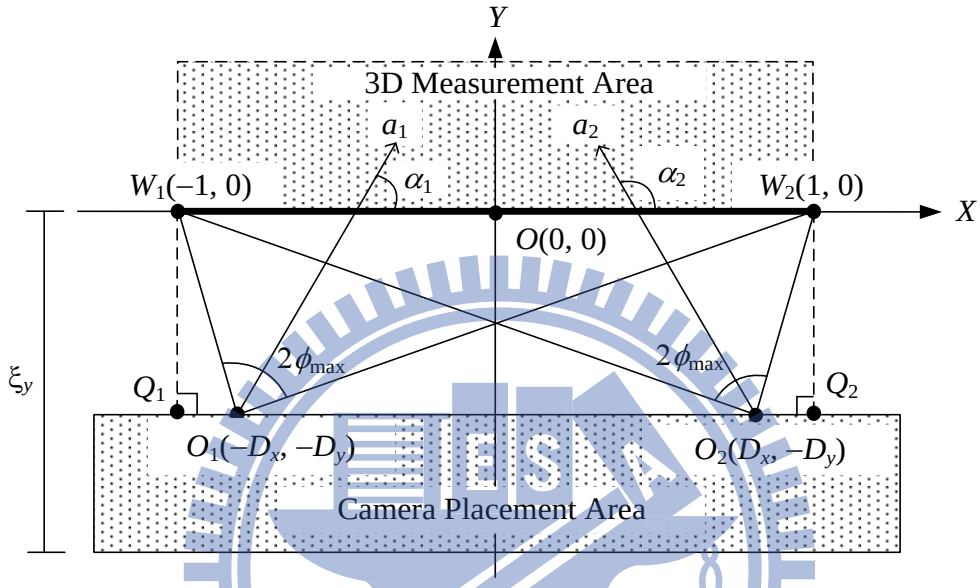


Fig. 6.5 An illustration of the regular cases.

As described in the optimization framework, a system configuration includes all the necessary parameters to design a vision system, and an optimization process needs to find the optimal system configuration which yields the best 3D measurement accuracy. A criterion function E_w for the optimization is defined in this study to be the *maximum* measurement error within the 3D measurement area W, i.e.,

$$E_w(W) = \max_{P \in W} \left(E(P) \right), \tag{55}$$

where $E(P)$ is the measurement error of a feature point P as derived by (50) in Theorem 6.2. By choosing the *maximum* value, *all* the 3D measurements errors are ensured to be lower than the value E_w. Next, as assumed, the two perspective cameras used in the omni-cameras are identical, so their focal lengths f_1 and f_2 are both equal to f, and their viewing angles $2\tau_{\max 1}$ and $2\tau_{\max 2}$ are both equal to $2\tau_{\max}$. The two omni-cameras are identical in structure and placed symmetrically, so the two optical axes a_1 and a_2 are coplanar so that the two optical axes can be defined by the two

angles α_1 and α_2 as shown in Fig. 6.5. Then, a system configuration can be defined to be the parameter set (D_x, D_y, α_1, α_2, ε_1, ε_2), where (1) the omni-cameras are placed, as seen from the top, at $O_1(-D_x, -D_y)$ and $O_2(D_x, -D_y)$, respectively; (2) the orientations of their optical axes are defined by α_1 and α_2, respectively; and (3) the eccentricities of the mirrors are ε_1 and ε_2, respectively. Hence, the optimization problem is just to find the optimal parameter set (D_x^*, D_y^*, α_1^*, α_2^*, ε_1^*, ε_2^*) derived in the following way:

$$(D_x^*, D_y^*, \alpha_1^*, \alpha_2^*, \varepsilon_1^*, \varepsilon_2^*) = \underset{(D_x, D_y, \alpha_1, \alpha_2, \varepsilon_1, \varepsilon_2)}{\arg\min} \left(E_w(W)\right). \tag{56}$$

Since it is desired that the captured image be fully filled up with the 3D measurement area, the cameras should be oriented to face the 3D measurement area. Accordingly, the optical axes a_1 and a_2 can be figured out to be just the bisectors of the angles spanned by the measurement area as depicted in Fig. 6.5, that is, the optical axis a_1 of the left omni-camera is the bisector of the viewing angle formed by $\overline{O_1W_1}$ and $\overline{O_1W_2}$, and the optical axis a_2 is the bisector of the viewing angle formed by $\overline{O_2W_1}$ and $\overline{O_2W_2}$. In view of these facts, the angle $\phi_{\max}$ and the optical-axis angles α_1 and α_2 can be calculated in terms of D_x and D_y as follows. First, the fact $\alpha_2 = \pi - \alpha_1$ can be derived from Fig. 6.5. Then, from the triangle formed by O_1, W_2, and Q_2, we have $D_y = (D_x + 1)\times\tan(\alpha_1 - \phi_{\max})$. Similarly, from the triangle formed by O_1, W_1, and Q_1, we have $D_y = (D_x - 1) \times \tan(\alpha_1 + \phi_{\max})$. Accordingly, the two unknowns α_1 and $\phi_{\max}$ can be solved respectively to be

$$\begin{cases} \alpha_1 = 0.5\left[\tan^{-1}(D_y/D_x - 1) + \tan^{-1}(D_y/D_x + 1)\right]; \\ \phi_{\max} = 0.5\left[\tan^{-1}(D_y/D_x - 1) - \tan^{-1}(D_y/D_x + 1)\right]. \end{cases} \tag{57}$$

Then, based on Theorem 6.1, the eccentricities ε_1 and ε_2 are

$$\varepsilon_1 = \varepsilon_2 = \varepsilon = \frac{\sin\phi_{\max} + \sin\tau_{\max}}{\sin(\phi_{\max} - \tau_{\max})}, \tag{58}$$

where $2\tau_{\max}$ is the viewing angle of the perspective cameras. To sum up, the optimization problem (56) is now reduced to include two parameters as follows:

$$\left(D_x^*, D_y^*\right) = \underset{(D_x, D_y)}{\arg\min}\left(E_w(W)\right) = \underset{(D_x, D_y)}{\arg\min}\left(\max_{P\in W}\left(E(P)\right)\right). \tag{59}$$

For further simplifications, two claims are given as follows.

Claim 6.1. The function E_w described by (55) can be rewritten as

$$E_w(W) = \max_{P \in W} \big(E(P)\big) = \max(E(O), E(W_2)), \tag{60}$$

if the two terms $R(\varepsilon, f, \phi_1)$ and $R(\varepsilon, f, \phi_2)$ are equal, where points O and W_2 are located at (0, 0) and (1, 0), respectively.

Proof. At first, by referring to Fig. 6.6(a), the measurement error of a point P at coordinates $(P_x, 0)$ can be derived using (50) and (51) with

$$\theta_1 = \cos^{-1}\frac{P_x + D_x}{\sqrt{(P_x + D_x)^2 + D_y^2}}, \quad \theta_2 = \cos^{-1}\frac{P_x - D_x}{\sqrt{(P_x - D_x)^2 + D_y^2}},$$

$$G_1(P) = \frac{\sqrt{(P_x + D_x)^2 + D_y^2}}{\sqrt{R(\varepsilon, f, \phi_1)}}, \text{ and } G_2(P) = \frac{\sqrt{(P_x - D_x)^2 + D_y^2}}{\sqrt{R(\varepsilon, f, \phi_2)}}. \tag{61}$$

Next, the function $E(P)$ is proved to be an even function as follows. From Fig. 6.6(a), the angles ϕ_1 and ϕ_2 can be seen to be

$$\phi_1 = |\alpha_1 - \theta_1|, \text{ and } \phi_2 = |\alpha_2 - \theta_2|. \tag{62}$$

From the resolution formula (40), we can get the equality $R(\varepsilon, f, \phi) = R(\varepsilon, f, -\phi)$ for any angle ϕ. Accordingly, we have

$$R(\varepsilon, f, \phi_1) = R(\varepsilon, f, \alpha_1 - \theta_1); R(\varepsilon, f, \phi_2) = R(\varepsilon, f, \alpha_2 - \theta_2). \tag{63}$$

Let P' be the point located at $(-P_x, 0)$, and let the related angles θ_1', θ_2', ϕ_1', and ϕ_2' be defined as those shown in Fig. 6.6(a). Since the triangles $\Delta O_1O_2P'$ and ΔO_1O_2P are similar, we get $\theta_1' = \pi - \theta_2$ and $\theta_2' = \pi - \theta_1$. Combining these facts with (62), we have

$$\begin{aligned}\phi_1' &= |\alpha_1 - \theta_1'| = |(\pi - \alpha_2) - (\pi - \theta_2)| = |-\alpha_2 + \theta_2| = \phi_2;\\ \phi_2' &= |\alpha_2 - \theta_2'| = |(\pi - \alpha_1) - (\pi - \theta_1)| = |-\alpha_1 + \theta_1| = \phi_1.\end{aligned} \tag{64}$$

Thus, the following equality can be derived:

$$G_1(P') = \frac{\sqrt{(-P_x + D_x)^2 + D_y^2}}{\sqrt{R(\varepsilon, f, \phi_1')}} = \frac{\sqrt{(D_x - P_x)^2 + D_y^2}}{\sqrt{R(\varepsilon, f, \phi_2)}} = G_2(P). \tag{65}$$

The function $E_1(P)$ can be proved accordingly to be an even function by

$$E_1(P)=\frac{\sqrt{G_1^{\,2}(P)+2G_1(P)G_2(P)\cos(\theta_2-\theta_1)+G_2^{\,2}(P)}\sqrt{dA}}{\sin(\theta_2-\theta_1)}$$

$$=\frac{\sqrt{G_2^{\,2}(P')+2G_2(P')G_1(P')\cos(\theta_2'-\theta_1')+G_1^{\,2}(P')}\sqrt{dA}}{\sin(\theta_2'-\theta_1')}=E_1(P')\,. \qquad (66)$$

Similarly, we can prove $E_2(P) = E_2(P')$, meaning that $E_2(P)$ is an even function, too.

Finally, we prove in the following the property that if $R(\varepsilon, f, \phi_1)$ and $R(\varepsilon, f, \phi_2)$ are equal, the maximum of $E(P)$ will occurs at $O(0, 0)$ or $W_2(1, 0)$. Firstly, let P_x^* be a value of P_x such that the case $\theta_2 - \theta_1 = 90°$ occurs. When $0 \le P_x \le P_x^*$, we get $90^o \le \theta_2 - \theta_1 \le 180^o$ so that $\cos(\theta_2 - \theta_2) \le 0$, implying that $E_2(P) \le E_1(P)$ according to (21), which leads to the following fact:

$$E(P)=E_2(P)\,. \qquad (67)$$

Furthermore, by applying the law of cosines and the assumption $R(\varepsilon, f, \phi_1) = R(\varepsilon, f, \phi_2)$, $E_2(P)$ can be reduced in the following way:

$$E_2(P)=\frac{\sqrt{|O_1P|^2-2|O_1P||O_2P|\cos(\theta_2-\theta_1)+|O_2P|^2}}{\sin(\theta_2-\theta_1)}=\frac{|O_1O_2|}{\sin(\theta_2-\theta_1)}\,. \qquad (68)$$

Accordingly, since the angle $\theta_2 - \theta_1$ decreases from 180^o to $90°$ when P_x increases from 0 to P_x^*, the maximum of $E(P)$ occurs at $P_x = 0$. For the other case that $P_x^* \le P_x \le 1$, we can get $0^o \le \theta_2 - \theta_1 \le 90^o$ so that $\cos(\theta_2 - \theta_1) \ge 0$, implying that $E_1(P) \le E_2(P)$, which leads to the following fact according to (68):

$$E(P)=E_1(P)\,. \qquad (69)$$

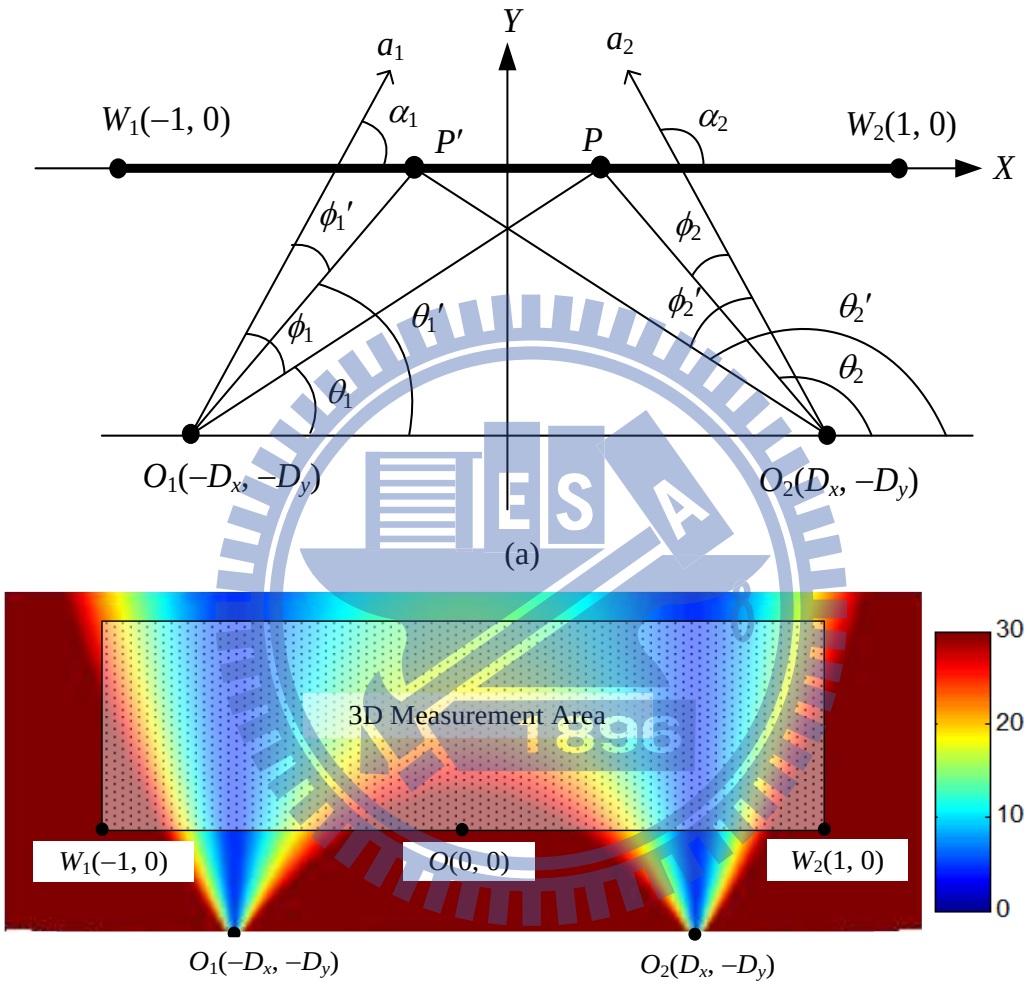
Furthermore, according to (61), $E_1(P)$ can be expressed as

$$E_1(P)=\frac{\sqrt{|O_1O_2|^2+4\overrightarrow{PO_1}\cdot\overrightarrow{PO_2}}}{\sin(\theta_2-\theta_1)}=\frac{\sqrt{|O_1O_2|^2+4(P_x+D_x,D_y)\cdot(P_x-D_x,D_y)}}{\sin(\theta_2-\theta_1)}\,. \qquad (70)$$

Accordingly, since the angle $\theta_2 - \theta_1$ decreases from $90°$ to 0^o when P_x increases, the maximum of $E(P)$ occurs at $P_x = 1$. Combine the results of the two cases, we get the conclusion that the maximum occurs at $O(0, 0)$ or $W_2(1, 0)$.

Finally, since both $E_1(P)$ and $E_2(P)$ are even functions, this conclusion may also be proved to hold for the "left-side" range $-1 \le P_x \le 0$. Therefore, the overall conclusion described by (60) may be drawn. □

In the above proof, the assumption $R(\varepsilon, f, \phi_1) = R(\varepsilon, f, \phi_2)$ is made, which is proved later by simulation results to be appropriate with very little affection on the 3D measurement precision of the derived system configuration (see the experimental results shown in a later section).



(b)

Fig. 6.6 Analysis of function E_w. (a) Illustrations of related terms. (b) Drawing of distribution of measurement errors E of a configuration.

Claim 6.2. A larger value of D_y always yields a smaller value of the criterion function E_w.

Proof. The *inscribed angle theorem* says that an angle θ inscribed in a circle is a half of the central angle 2θ that subtends the same arc on the circle [57]. That is, if the

viewing angle is $2\phi_{\max}$, the possible positions of the cameras can be figured out to be constrained on the dashed circle shown in Fig. 6.7(a), and the upper bound of D_y occurs at the bottom of the circle. Also, while recalling that the two cameras are *omni-directional*, we assume their viewing angle $2\phi_{\max}$ to be larger than 120°. So we have

$$\max(D_y) = \cot\phi_{\max} \le \cot(60°) \approx 0.6. \tag{71}$$

With this upper bound, the function E_w is plotted in Fig. 6.7(b), which shows that a larger value of D_y yields a smaller value of E_w. □

With Claim 6.1, Eq. (59) can be re-formulated as

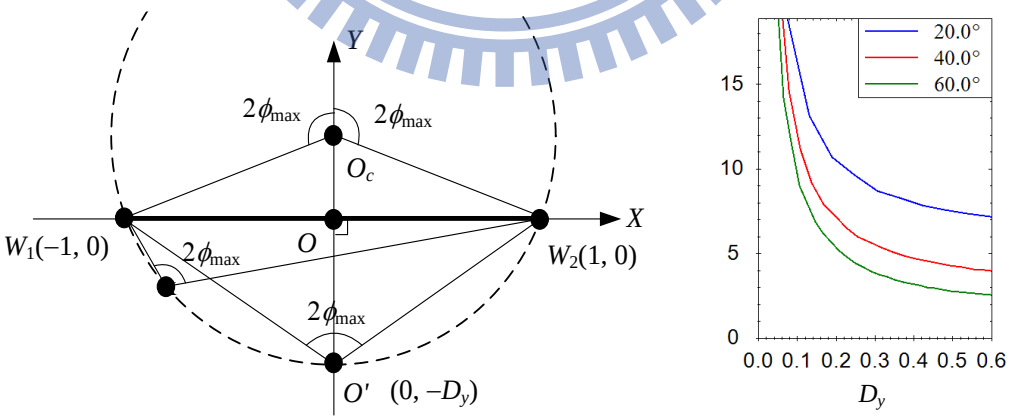
$$(D_x^*, D_y^*) = \arg\min_{(D_x, D_y)} \left(\max(E(O), E(W_2))\right). \tag{72}$$

Also, recall that the upper bound of D_y is limited by the camera deployment constraint, which we denoted as ξ_y in Fig. 6.5. With Claim 6.2, the optimal value D_y^* in (72) can be derived to be $\min(\xi_y, 0.6)$, leaving D_x^* to be the only parameter to be optimized according to the following constraint:

$$D_x^* = \arg\min_{D_x} \left(\max\left(E_{\text{mid}}(D_x, D_y^*), E_{\text{bound}}(D_x, D_y^*)\right)\right), \tag{73}$$

where E_{bound} and E_{mid} are defined as

$$E_{\text{bound}}(D_x, D_y^*) = E(W_2); \qquad E_{\text{mid}}(D_x, D_y^*) = E(O). \tag{74}$$



(a) (b)

Fig. 6.7 Finding the optimal value of D_y. (a) An illustration to find the upper bound. (b) A plot of E_w for different values of $2\tau_{\max}$: 20°, 40°, and 60°.

An optimization algorithm to solve D_x^* in (73) by a bisection scheme is proposed in the following. By referring to the plots of E_{mid} and E_{bound} depicted in Fig. 6.8(a), the optimal solution D_x^* is found at the intersection of the two functions of E_{mid} and E_{bound}. So, if the two functions intersect each other, the intersection point may be defined to be the optimal solution D_x^*; otherwise, the optimal solution D_x^* is defined to be the maximum value which will also be derived later in this section.

In more details, at first we define a new function E_{opt} as

$$E_{\text{opt}}(D_x, D_y^*) = E_{\text{mid}}(D_x, D_y^*) - E_{\text{bound}}(D_x, D_y^*). \tag{75}$$

Then, the optimal solution D_x^* is just the root of E_{opt}, which can be derived by a bisection scheme. Before the scheme is conducted, the initial range of the root must be determined. The lower bound $lower_{Dx}$ of D_x is obviously zero, and the upper bound $upper_{Dx}$ is derived as follows. From Fig. 6.8(b), we have

$$\left|\overline{O_cO_2}\right| = \left|\overline{O_cW_2}\right| = \csc(\pi - 2\phi_{\max}) = \csc(2\phi_{\max}). \tag{76}$$

And the coordinates of the circle center O_c is

$$O_c = (0, \cot(\pi - 2\phi_{\max})) = (0, -\cot(2\phi_{\max})). \tag{77}$$

According to the Pythagorean Theorem, we have

$$\begin{aligned} D_x^2 &= \left|\overline{O_cO_2}\right|^2 - \left|\overline{O_cO_2'}\right|^2 = \csc^2(2\phi_{\max}) - [D_y^* - \cot(2\phi_{\max})]^2 \\ &= 1 - D_y^*[D_y^* - 2\cot(2\phi_{\max})], \end{aligned} \tag{78}$$

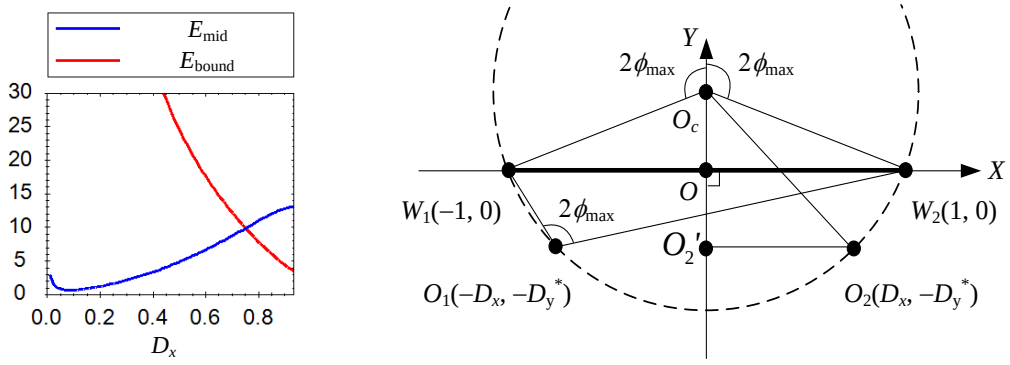
and the first derivative of D_x^2 with respect to $2\phi_{\max}$ is

$$\frac{\partial(D_x^2)}{\partial(2\phi_{\max})} = 2D_y^*(-\csc^2(2\phi_{\max})). \tag{79}$$

Since $D_y^* > 0$, the first derivative of D_x^2 is smaller than zero, meaning that D_x decreases as $2\phi_{\max}$ increases, or equivalently, that the maximum of D_x occurs when $\phi_{\max}$ is minimized. So, the upper bound of D_x can be derived from (78) to be

$$upper_{Dx} = \sqrt{1 - D_y^*[D_y^* - 2\cot(2\phi_{\max})]}. \tag{80}$$

A method to solve the optimization problem is proposed below.



(a) (b)

Fig. 6.8 Illustrations of finding the optimal solution D_x^* and its upper bound. (a) Plot of E_{mid} and E_{bound} versus D_x when $D_y^* = 0.1$ and $2\tau_{max} = 60.0°$. (b) Derivation of the upper bound of D_x.

Algorithm 6.1. *Finding the optimal configuration* (D_x^*, D_y^*).

Input: the viewing angle $2\tau_{max}$ and the focal length f of the cameras.

Output: the optimal system configuration (D_x^*, D_y^*), meaning that the omni-cameras are placed at $O_1(-D_x^*, D_y^*)$ and $O_2(D_x^*, D_y^*)$, and oriented as shown in Fig. 6.5.

Steps.

Step 1. Calculate ξ_y according to the deployment size as stated, and set $D_y^* = \min(\xi_y, 0.6)$.

Step 2. Set variable $lower_{Dx} = 0$ and calculate $E_{opt}(lower_{Dx}, D_y^*)$ as follows and assign the result to the variable *lower*.

2.1 Set $D_x = lower_{Dx}$ and calculate ϕ_{max} according to (57).

2.2 Calculate the eccentricity ε according to (58).

2.3 Calculate E_{mid} and E_{bound} by (74) with $O_1 = (-D_x, D_y^*)$ and $O_2 = (D_x, D_y^*)$, and calculate E_{opt} by (75).

Step 3. Calculate the upper bound $upper_{Dx}$ of D_x^* by (80).

Step 4. Calculate $E_{opt}(upper_{Dx}, D_y^*)$ in a way similar to Steps 2.1 through 2.3, and assign the result to the variable *upper*.

Step 5. If *lower* and *upper* are with opposite signs, then find the root in a bisecting fashion as follows.

5.1. Set variable $new_{Dx} = (lower_{Dx} + upper_{Dx})/2$.

5.2. Calculate $E_{opt}(new_{Dx}, D_y^*)$ in a way similar to Steps 2.1 through 2.3 and

assign the result to the variable *new*.

5.3. If (*new* < *lower*), then set *lower* = *new* and $lower_{Dx}$ = new_{Dx}; otherwise, set *upper* = *new* and $upper_{Dx}$ = new_{Dx}.

5.4. If ($upper_{Dx}$ – $lower_{Dx}$) < δ, where δ is a predefined precision threshold, then take (D_x^*, D_y^*) = (new_{Dx}, ξ_y) as the output and exit; otherwise, go to Step 5.1.

Step 6. If *lower* > 0 and *upper* > 0 or if *lower* < 0 and *upper* < 0, then choose D_x^* to be the upper bound $upper_{Dx}$, and take (D_x^*, D_y^*) = ($upper_{Dx}$, ξ_y) as the output.

The method proposed in Algorithm 6.1 is further simplified to get an analytic formula for deriving a sub-optimal solution. Let v_1 and v_2 be two vectors, and θ be the included angle. We have

$$|v_1 \pm v_2|^2 = (v_1 \pm v_2) \cdot (v_1 \pm v_2) = |v_1|^2 \pm 2|v_1||v_2|\cos\theta + |v_2|^2. \tag{81}$$

By referring to Fig. 6.9, the formula of the measurement error E_{mid} can be derived, using (74), (50), (51), and (81), to be

$$E_{\text{mid}}(D_x, D_y^*) = \frac{\max\left(\sqrt{\left|\overrightarrow{O_1O}\right|^2 \pm 2\left|\overrightarrow{O_1O}\right|\left|\overrightarrow{O_2O}\right|\cos(\theta_{\text{mid2}} - \theta_{\text{mid1}}) + \left|\overrightarrow{O_2O}\right|^2}\right)\sqrt{dA}}{\sin(\theta_{\text{mid2}} - \theta_{\text{mid1}})\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}}$$

$$= \frac{\max\left(\left|\overrightarrow{O_1O} + \overrightarrow{O_2O}\right|, \left|\overrightarrow{O_1O} - \overrightarrow{O_2O}\right|\right)\sqrt{dA}}{\sin(\theta_{\text{mid2}} - \theta_{\text{mid1}})\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}} = \frac{\max\left(2D_y^*, 2D_x\right)\sqrt{dA}}{\sin(\theta_{\text{mid2}} - \theta_{\text{mid1}})\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}}. \tag{82}$$

Referring to Fig. 6.9 and based on the double-angle formula of the sine function, we can get

$$\sin(\theta_{\text{mid2}} - \theta_{\text{mid1}}) = 2\sin\left(\frac{\theta_{\text{mid2}} - \theta_{\text{mid1}}}{2}\right)\cos\left(\frac{\theta_{\text{mid2}} - \theta_{\text{mid1}}}{2}\right) = \frac{2D_y^* D_x}{D_x^2 + (D_y^*)^2}. \tag{83}$$

Thus, the function E_{mid} in (82) can be rewritten as

$$E_{\text{mid}}(D_x, D_y^*) = \frac{\left[D_x^2 + (D_y^*)^2\right]\sqrt{dA}}{\frac{2D_y^* D_x\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}}{\max\left(2D_y^*, 2D_x\right)}} = \frac{\left[D_x^2 + (D_y^*)^2\right]\sqrt{dA}}{\min\left(D_x, D_y^*\right)\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}} \tag{84}$$

Similarly, the measurement error E_{bound} of the feature point $W_2(1, 0)$ can be simplified, using (50), (51), (74) and (81), to be

$$E_{\text{bound}}(D_x, D_y^*) = \frac{\max(\sqrt{\left|\overline{O_1W_2}\right|^2 \pm 2\left|\overline{O_1W_2}\right|\left|\overline{O_2W_2}\right|\cos(\theta_{\max 2} - \theta_{\max 1}) + \left|\overline{O_2W_2}\right|^2})\sqrt{dA}}{\sin(\theta_{\max 2} - \theta_{\max 1})\sqrt{R(\varepsilon, f, \phi_{\max})}}$$

$$= \frac{\max\left(2\sqrt{1+(D_y^*)^2}, 2D_x\right)\sqrt{dA}}{\sin(\theta_{\max 2} - \theta_{\max 1})\sqrt{R(\varepsilon, f, \phi_{\max})}}. \quad (85)$$

From the geometry shown in Fig. 6.9, we have

$$\sin(\theta_{\max 2} - \theta_{\max 1}) = \sqrt{1 - \left(\frac{\overrightarrow{O_1W_2} \cdot \overrightarrow{O_2W_2}}{\left|\overrightarrow{O_1W_2}\right|\left|\overrightarrow{O_2W_2}\right|}\right)^2} = \frac{2D_xD_y^*}{\sqrt{\left(D_x^2 + (D_y^*)^2 + 1\right)^2 - 4D_x^2}} \quad (86)$$

Thus, the function E_{bound} in (85) can be rewritten to be

$$E_{\text{bound}}(D_x, D_y^*) = \frac{\sqrt{\left(D_x^2 + (D_y^*)^2 + 1\right)^2 - 4D_x^2}\,\max\left(\sqrt{1+(D_y^*)^2}, D_x\right)\sqrt{dA}}{D_xD_y^*\sqrt{R(\varepsilon, f, \phi_{\max})}}. \quad (87)$$

By combining (84) and (87), the function E_{opt} can be re-formulated as

$$E_{\text{opt}}(D_x, D_y^*) = \frac{\left(D_x^2 + (D_y^*)^2\right)\sqrt{dA}}{\min\left(D_y^*, D_x\right)\sqrt{R(\varepsilon, f, \phi_{\text{mid}})}} - \frac{\max\left(\sqrt{1+(D_y^*)^2}, D_x\right)\sqrt{\left(D_x^2 + (D_y^*)^2 + 1\right)^2 - 4D_x^2}\,\sqrt{dA}}{D_xD_y^*\sqrt{R(\varepsilon, f, \phi_{\max})}}. \quad (88)$$

To calculate the optimal system configuration, the value D_y^* is firstly derived in the same way as stated in Algorithm 6.1. Then, the optimal solution D_x^* is just the root of the function E_{opt} described by (88). However, the root does not have an analytic formula since the involved terms ϕ_{mid}, $\phi_{\max}$ and ε are all with complicated formulas with respect to D_x. In order to derive an analytic solution for the root, we first propose a skillful approximation method to produce a new function E_{opt}' to approximate the original one E_{opt}, and then derive an analytic formula to compute a sub-optimal solution D_x'. This sub-optimal solution D_x' is a good approximation to the optimal one D_x^* as will be shown later in this section.

To simplify the function E_{opt} described by (88), we assume further

$$R(\varepsilon, f, \phi_{\text{mid}}) \approx R(\varepsilon, f, \phi_{\max}), \quad (89)$$

which is a special case of the assumption $R(\varepsilon, f, \phi_1) = R(\varepsilon, f, \phi_2)$ made before in the proof of Claim 1. This new assumption can be proved as well later by simulation

results to be proper with very little affection on the 3D measurement precision of the derived system configuration (see Fig. 6.10). Consequently, Equation (88) may now be simplified to be

$$E_{\text{opt}}'(D_x,\ D_y^*) = \left(\frac{\sqrt{dA}}{\sqrt{R(\varepsilon, f, \phi_{\max})}}\right) \times \left(\frac{D_x^{\,2} + (D_y^*)^2}{\min\left(D_y^*, D_x\right)} - \frac{\max\left(\sqrt{1+(D_y^*)^2}, D_x\right)\sqrt{[D_x^{\,2} + (D_y^*)^2 + 1]^2 - 4D_x^{\,2}}}{D_x D_y^*}\right). \quad (90)$$

Thus, the root D_x' of the equation $E_{\text{opt}}'(D_x, D_y^*) = 0$ satisfies

$$\frac{(D_x')^2 + (D_y^*)^2}{\min\left(D_y^*, D_x'\right)} = \max\left(\sqrt{1+(D_y^*)^2}, D_x'\right)\frac{\sqrt{[(D_x')^2 + (D_y^*)^2 + 1]^2 - 4(D_x')^2}}{D_x' D_y^*}. \quad (91)$$

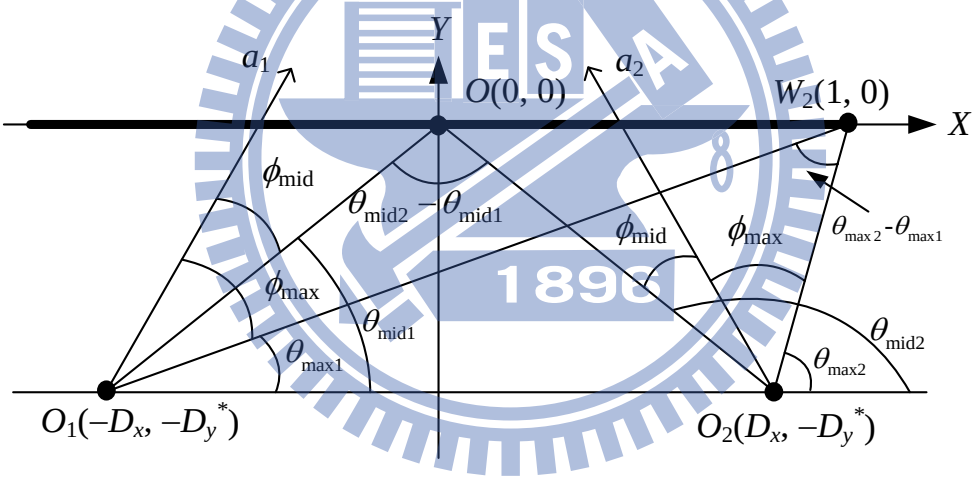


Fig. 6.9 Related parameters involved in E_{mid} and E_{bound}.

Theorem 6.3. The solution of D_x' in (91) is

$$D_x' = \sqrt{\frac{-4(D_y^*)^4 - (C-2)(D_y^*)^2 - C^2 + C + 5}{3C}}, \quad (92)$$

where

$$C = \sqrt[3]{0.5Q - 8(D_y^*)^6 - 48(D_y^*)^4 - 46.5(D_y^*)^2 - 5.5};$$

$$Q = \sqrt{27\left(1+(D_y^*)^2\right)\left(128(D_y^*)^8 + 352(D_y^*)^6 + 288(D_y^*)^4 + 75(D_y^*)^2 + 23\right)}. \quad (93)$$

Proof. To deal with the involved min/max function in (91), four cases are discussed separately, which are: (1) $D_x' < D_y^*$ and $D_x' \le (1+(D_y^*)^2)^{0.5}$; (2) $D_x' < D_y^*$ and $D_x' > (1+(D_y^*)^2)^{0.5}$; (3) $D_x' \ge D_y^*$ and $D_x' \le (1+(D_y^*)^2)^{0.5}$; and (4) $D_x' \ge D_y^*$ and $D_x' > (1+(D_y^*)^2)^{0.5}$. It is proved next that only Case (3) is valid.

For Case (1), we have the assumptions $D_x' < D_y^*$ and $D_x' \le (1+(D_y^*)^2)^{0.5}$, so that (91) can be reduced to be

$$\frac{(D_x')^2+(D_y^*)^2}{D_x'} = \sqrt{1+(D_y^*)^2}\,\frac{\sqrt{[(D_x')^2+(D_y^*)^2+1]^2-4(D_x')^2}}{D_x'D_y^*}, \tag{94}$$

or equivalently, to be

$$D_y^*[(D_x')^2+(D_y^*)^2] = \sqrt{1+(D_y^*)^2}\,\sqrt{[(D_x')^2+(D_y^*)^2+1]^2-4(D_x')^2}\,. \tag{95}$$

Defining $A = (D_x')^2$ and $B = (D_y^*)^2$, Eq. (95) can be expressed as

$$A^2 - 2A + (3B^2 + 3B + 1) = 0. \tag{96}$$

Since the discriminant of (96) is $4 - 4(3B^2 + 3B + 1) < 0$, the solution of A does not exist, or equivalently, D_x' does not exist. As a result, the assumptions made for Case (1) are invalid.

The assumptions made for Case (2) are $D_x' < D_y^*$ and $D_x' > (1+(D_y^*)^2)^{0.5}$. Since these two inequalities are contradictory to each other, Case (2) is also out of consideration.

For Case (4), the two assumptions are $D_x' \ge D_y^*$ and $D_x' > (1+(D_y^*)^2)^{0.5}$. Thus, Equation (91) can be rewritten to be

$$(D_x')^2+(D_y^*)^2 = \sqrt{[(D_x')^2+(D_y^*)^2+1]^2-4(D_x')^2}\,. \tag{97}$$

Defining $A = (D_x')^2$, $B = (D_y^*)^2$ and taking the squares of both sides of the above equation, we have $(A + B)^2 = (A + B + 1)^2 - 4A$, or equivalently, $2A = 2B + 1$. However, from the second assumption $D_x' > (1+(D_y^*)^2)^{0.5}$, we get $A > B + 1$, which is a contradiction to the equation $2A = 2B + 1$ derived previously. Therefore, the assumptions made for Case (4) are also invalid.

As a result, Case (3) is the only valid one, for which the two assumptions are $D_x' \ge D_y^*$ and $D_x' \le (1+(D_y^*)^2)^{0.5}$. Accordingly, Equation (91) can be rewritten to be

$$\frac{(D_x')^2+(D_y^*)^2}{D_y^*}=\sqrt{1+(D_y^*)^2}\,\frac{\sqrt{[(D_x')^2+(D_y^*)^2+1]^2-4(D_x')^2}}{D_x'D_y^*}, \tag{98}$$

or equivalently, to be

$$aA^3 + bA^2 + cA + d = 0 \tag{99}$$

where $A = (D_x')^2$, $B = (D_y^*)^2$, $a = 1$, $b = B - 1$, $c = 2 - B^2$, and $d = -(B + 1)^3$. To find the solution of A for the cubic function (99), we first calculate its discriminant Δ, according to [58], as

$$\begin{aligned}\Delta &= 18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2\\ &= -(B + 1)(128B^4 + 352B^3 + 288B^2 + 75B + 23),\end{aligned} \tag{100}$$

which is smaller than zero because $B = (D_y^*)^2 > 0$. Thus, we get to know that the cubic polynomial equation has only one real root which can be described by (92) according to [58]. □

The above-described process of generating the sub-optimal configuration (D_x', D_y^*) is summarized as an algorithm below.

Algorithm 6.2. *Finding a sub-optimal configuration* (D_x', D_y^*) *by analytic formulas.*

Input: the viewing angle $2\tau_{max}$ and the focal length f of the cameras.

Output: the sub-optimal configuration (D_x', D_y^*), meaning that the omni-cameras are placed at $O_1(-D_x', D_y^*)$ and $O_2(D_x', D_y^*)$, and oriented as shown in Fig. 6.5.

Steps.

Step 1. Calculate ξ_y according to the deployment size as stated previously in this section, and set $D_y^* = \min(\xi_y, 0.6)$.

Step 2. Calculate the upper bound $upper_{Dx}$ of D_x^* by (80).

Step 3. Calculate D_x' by (92) derived in Theorem 6.3.

Step 4. Set $D_x' = \min(D_x', upper_{Dx})$.

Step 5. Output the optimal system configuration (D_x', D_y^*).

The sub-optimal configuration (D_x', D_y^*) is shown to be *a good approximation* to the optimal one (D_x^*, D_y^*) as follows. Recalling that the goal of the optimization is to minimize the measurement error E_w defined by (73), we use the function E_w as a criterion to analyze the precision of the approximate one. In Fig. 6.10, we plot the curves of the measurement error values of the optimal and sub-optimal configurations

for all the possible values of D_y, from which we see that the measurement errors are very close to each other, meaning that the sub-optimal configuration also yields precise 3D measurement results as the optimal configuration does. Also, we use a *difference ratio* defined by

$$(E_w' - E_w^*) / E_w^* \tag{101}$$

to determine the *goodness* of the performance of the sub-optimal configuration, where E_w^* and E_w' denote the measurement errors using the optimal and sub-optimal configurations, respectively. As shown in the figure, the difference ratio is smaller than 0.4%, showing that the sub-optimal solution is indeed a good approximation.

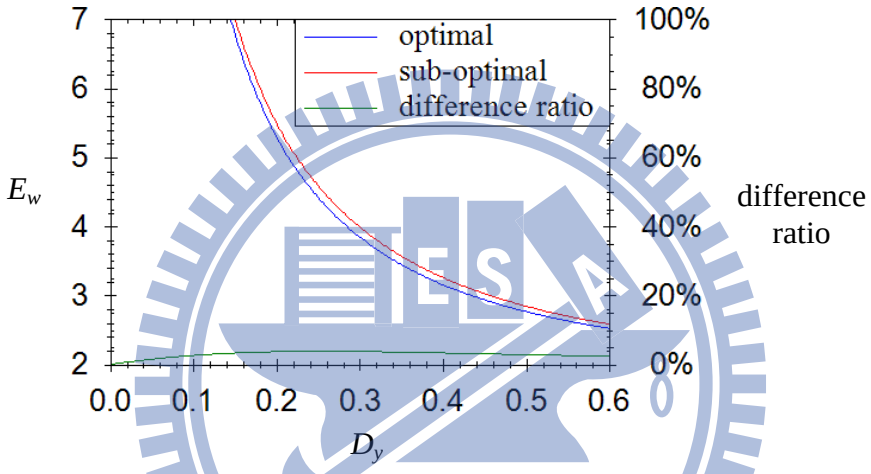


Fig. 6.10 Comparison of optimal configuration (D_x^*, D_y^*) and sub-optimal configuration (D_x', D_y^*) with viewing angle $2\tau_{max} = 60^o$.

6.5 Optimization for General Cases

To design a system configuration for the general case, the *3D measurement area* and the *camera placement area* are specified first. To make the descriptions of the possibly irregular shapes of the two areas easy, each area is described by multiple *sampled points*, called the *3D measurement locations* and *camera placement locations*, respectively. For example, a cuboid can be described by 10000 evenly-distributed points. The occlusion problem can be handled by just eliminating the camera locations where the 3D measurement locations will be partially occluded if the camera was placed there [59].

After the 3D measurement locations and the camera placement locations are both identified, the optimal system configuration can be found out as follows (see Fig. 6.11

for a flowchart). At First, two locations are chosen from the camera placement locations to be the positions of the left and right cameras, respectively. Then, the cameras are oriented to face the 3D measurement area to determine the extrinsic parameters of the omni-cameras. Also, their intrinsic parameters, including the mirror-shape parameter and the viewing angle of the perspective camera, are determined by the optimization method proposed in the previous section.

To decide which configuration yields the most precise 3D measurements, a measure indicating the degree of accuracy of the computed 3D data defined as $\min(-E(M_i))$ is calculated, where $-E(M_i)$ is the degree of accuracy of a 3D measurement location M_i as derived in Theorem 6.2. It is noted that, by choosing the *minimum* value, *all* the 3D measurement locations are ensured to be at least with this degree of accuracy. Then, to find the optimal system configuration, the previously-described steps are executed repeatedly for all possible camera locations as shown by the loop in Fig. 6.11, and the configuration with the highest degree of accuracy is picked out finally as the desired result.

After the positions of the two omni-cameras are decided, it is necessary to determine the optical axis and the viewing angle of each omni-camera. To solve this problem, the camera should face the 3D measurement locations, and this in turn determines the orientations of the camera. Specifically, let M_i denote the points in the 3D measurement locations, and O the chosen location of each omni-camera (i.e., the camera center). Then, the problem may be solved, as depicted in Fig. 6.12, by three steps: (1) find the smallest viewing cone containing all the 3D measurement locations M_i and with O as its apex; (2) set the optical axis as the viewing cone axis; and (3) take the viewing angle $2\phi_{\max}$ of the omni-camera to be the aperture of the cone.

To speed up the optimization method described in Fig. 6.11, three techniques are proposed as follows.

(1) *Longest baseline first*. When picking up the locations of the cameras in Step 3, the ones with large baselines (i.e., the distance between the cameras) are picked up first.

(2) *Farthest 3D point first*. When calculating the 3D data accuracy in Step 6, the 3D points which are farther from the two cameras are calculated first.

(3) *Early stopping*. If the computed 3D data accuracy is larger than that of the so-far best configuration, then the current configuration cannot be a better

one, and so the algorithm continues to perform Step 3 for the next configuration.

These techniques were implemented and tested by experiments, and the results show that they reduced the running time from 56.6 seconds down to 22.3, indicating a speedup of about 2.5 times.

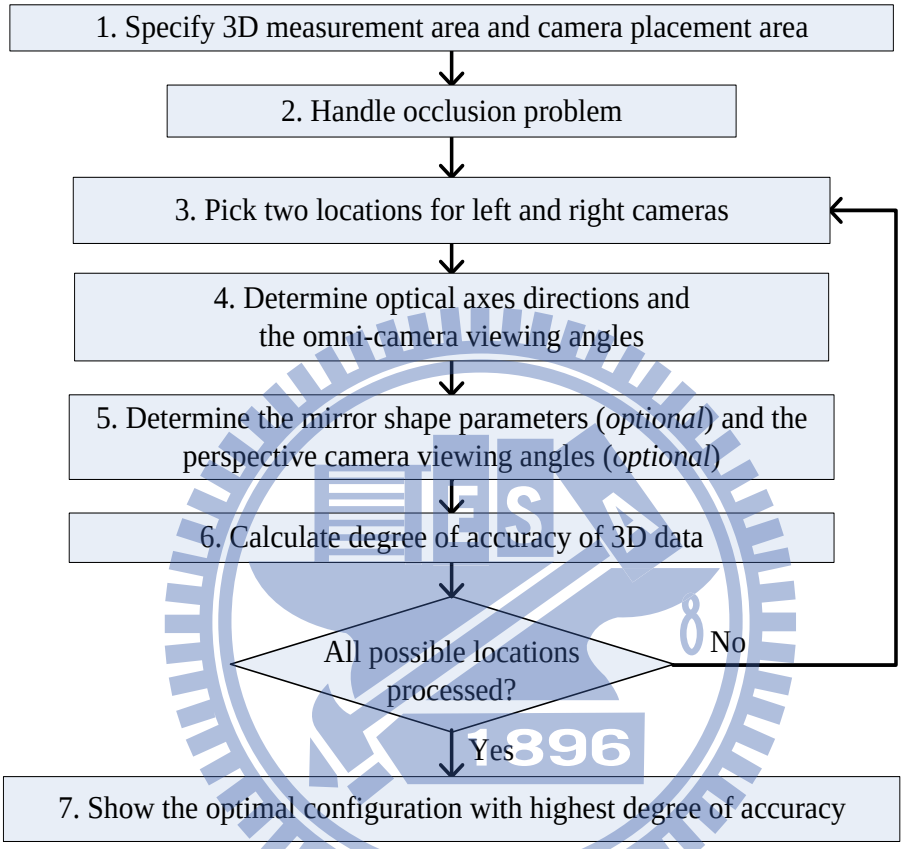


Fig. 6.11 Proposed optimization process to deal with general cases.

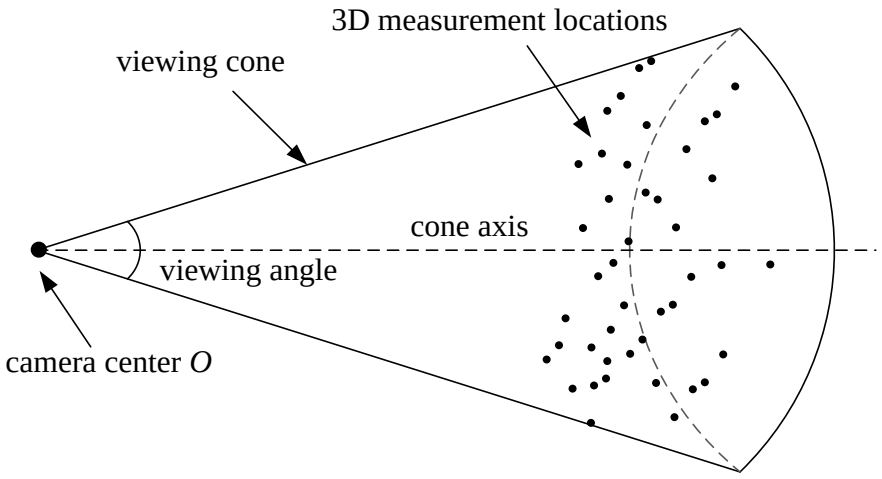


Fig. 6.12 An illustration of finding the optical axis and the viewing angle.

Three optimization methods have been proposed in the previous and this section, which are for the regular-optimal, regular-suboptimal, and general cases, respectively. These methods have their own advantages and disadvantages as described in the following.

(1) If the 3D measurement area and the camera placement area are both approximately of rectangular shapes, the first and second methods can be used; otherwise, the third one.

(2) If the optimization process can be done in an off-line fashion, the third method is suitable; otherwise, the first two methods should be used.

(3) If it requires a fast computation time, or the system has a low computation capability, the first two methods are suitable; and among the two, the second is the faster one but a little bit less accurate.

Some possible applications are listed next to demonstrate the uses of the three optimization methods. When designing a vision system for home entertainment, exhibitions, wide-area video surveillance, etc., since the camera positions can be derived in advance and the environment may be irregular-shaped and possibly with occlusions, the third optimization method should be used. However, if the cameras can be oriented automatically by computer, the first optimization method may be used in an online fashion to achieve better 3D measurement accuracies. On the other hand, if the cameras are mounted, for example, on unmanned vehicles to collect wide-area 3D information in realtime, the second optimization method should be used because fast computations according to analytic formulas can be conducted, and the saved computation power can be used for navigation, learning, event recognition, etc.

6.6 Experimental Results

In this section, the experimental results of a case study of finding the optimal system configuration in a simulated indoor environment are described first. And the proposed method is then compared with four existing methods by experimental results for a real laboratory environment.

A room with size 10m×2.5m×3m, as depicted in Fig. 6.13(a), was considered, and the 3D data of a user's body moving within the 3D measurement area were required to be calculated accurately. Also, the omni-cameras need be designed and placed in the

camera placement area. In the following, the optimal system configuration for this simulation case study is derived first using the proposed method. Then, comparisons of the 3D measurement accuracies yielded by the optimal system configuration and some non-optimal ones are presented.

To find the optimal system configuration, a coordinate system is defined first as depicted in Fig. 6.13(a) with the floor being taken to be the plane $Z = 0$, and the viewing angles $2\tau_{\max}$ of the perspective cameras chosen to be 60°. Next, about 100 evenly distributed points were generated in the 3D measurement area, and about 1000 similarly distributed points were generated in the camera placement area. Then, the optimization process proposed in the last section (as depicted in Fig. 6.11) was applied to such a *general* case of environment. And the generated configuration with the minimum 3D measurement error was chosen finally to be the desired optimal system configuration S_1^*, which, as illustrated in Fig. 6.13(a), includes: (1) the locations O_1 and O_2 of the two cameras at $(\mp 3.92m, -0.5m, 2.5m)$, respectively; (2) the optical axes a_1 and a_2 oriented in accordance with the vectors (±0.093, 0.996, 0.0), respectively; and (3) the eccentricities ε_1 and ε_2 of the mirrors both being 1.8967. The images taken by the two cameras were simulated by a ray tracing program POV-Ray, with two examples shown in Figs. 6.13(b) and 6.13(c).

Alternatively, the above problem of vision system design can be seen as a 2D one, in which only the *XY*-plane is considered, and the two cameras are installed at the middle height of 2.5m (recalling that the room is with a height of 5m). As depicted in Fig. 6.14, let W be the boundary line $\overline{w_1w_2}$ of the 3D measurement area which is nearer to the camera placement area, and let C be the boundary line of the camera placement area which is farther to the workspace. The 2D coordinate system is defined in such a way that the coordinates of W_1 and W_2 are (–1, 0) and (1, 0), respectively. In this sense, a unit in the coordinate system represents 5m in real space, so that the distance ξ_y between the line segments W and C, as depicted in Fig. 6.14, can be derived to be 0.1. Then, with the use of ξ_y, the two proposed optimization schemes described by Algorithms 6.1 and 6.2 were performed to such a *regular* case of environment to derive the optimal two camera locations.

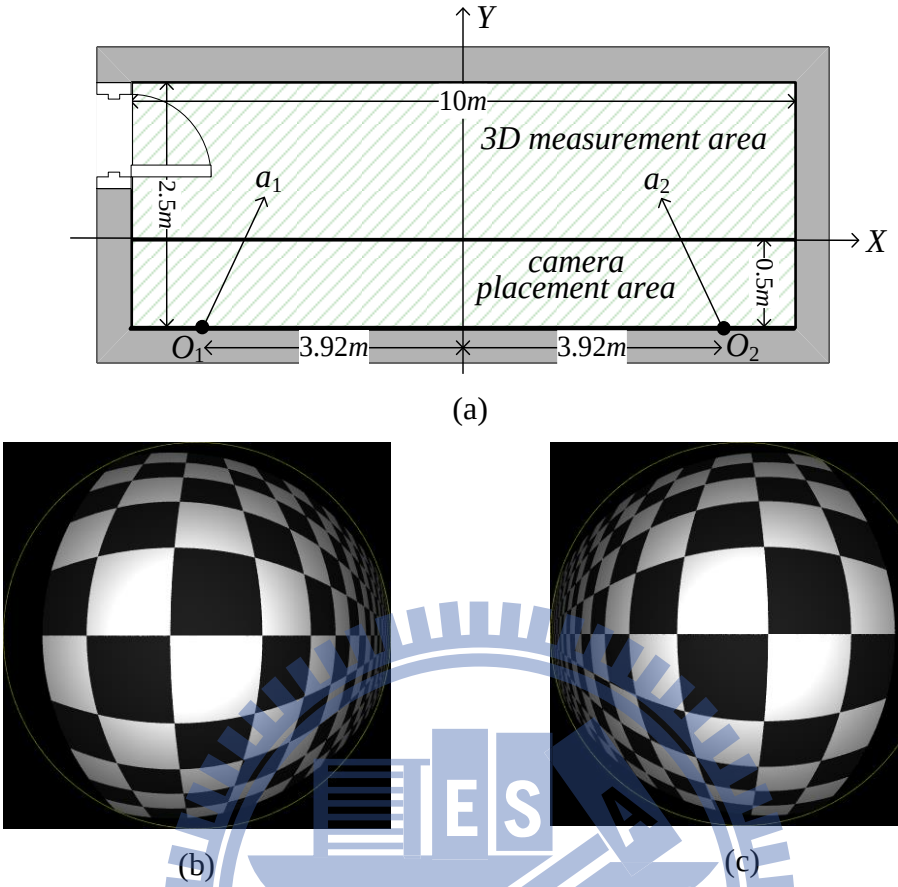


Fig. 6.13 Optimal system configuration for the general case derived by the proposed optimization process. (a) An illustration of the optimal system configuration. (b)(c) Simulated images taken by the two cameras, with 3D measurement area drawn as a checkerboard cube.

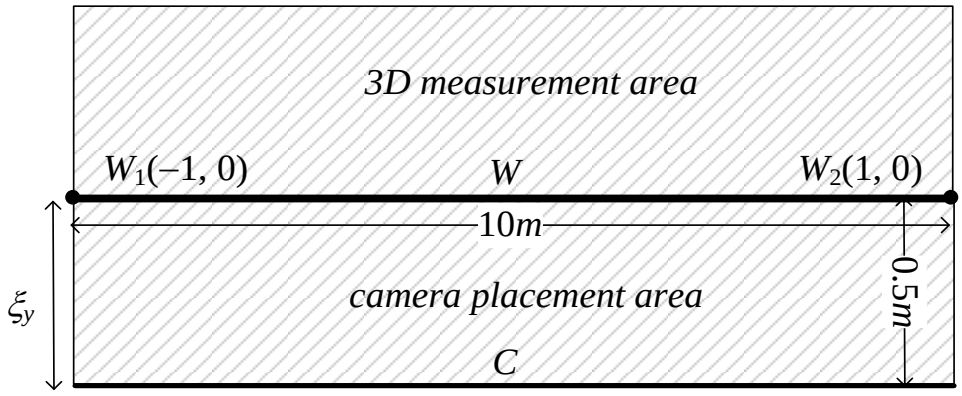


Fig. 6.14 The corresponding 2D problem of the case study.

The *optimal* camera locations derived by Algorithm 6.1 are ($\mp$0.753, –0.1) in the 2D coordinate system, which then were mapped back to the world coordinate system as ($\mp$3.766*m*, –0.5*m*). Subsequently, the proposed methods were applied to find the optical axes directions and eccentricities, yielding the optimal system configuration S_2^*, which includes: (1) the locations O_1 and O_2 of the two cameras at ($\mp$3.766*m*, –0.5*m*, 2.5*m*), respectively; (2) the optical axes a_1 and a_2 oriented in accordance with the vectors (±0.163, 0.987, 0.0), respectively; and (3) the eccentricities ε_1 and ε_2 of the mirrors both being 2.0067. In a similar way, the *sub-optimal* system configuration S_3^* were derived by applying Algorithm 6.2 and the related proposed methods, which includes: (1) the locations O_1 and O_2 of the two cameras at ($\mp$3.822*m*, –0.5*m*, 2.5*m*); (2) the optical axes a_1 and a_2 oriented in accordance with the vectors (±0.172, 0.985, 0.0), respectively; and (3) the eccentricities ε_1 and ε_2 both being 2.0194. It can be seen that the data of the two configurations S_2^* and S_3^* are quite close as expected.

To compare the 3D measurement accuracies yielded by the three different optimal system configurations S_1^*, S_2^*, and S_3^* derived previously, three additional system configurations S_1, S_2 and S_3 were chosen arbitrarily, in which the two cameras are located at (±2.5*m*, –0.5*m*, 2.5*m*) in configuration S_1, at (±1.67*m*, –0.5*m*, 2.5*m*) in S_2, and at (±3.33*m*, –0.5*m*, 2.5*m*) in S_3. Their optical axes' directions, and the eccentricities of the two mirrors were calculated similarly by the proposed methods.

Similar to the experiments described in 1) of this section, about 10,000 points were uniformly generated in the 3D measurement area. Each of the 10,000 point was firstly back-projected onto the two omni-images with coordinates (u_1, v_1) and (u_2, v_2). Then, Gaussian noise with zero means and standard deviations 1.0 (pixel) were applied to the four coordinate values (two for each omni-image), and the estimated location of the 3D point was calculated accordingly by mid-point triangulation [54]. The distance between the ground-truth 3D point and the estimated 3D point was then computed as the 3D measurement error. The minimum, maximum, and standard deviation of the 3D measurement errors resulting from the 10,000 points are listed in Table 6.1. Also, the proposed function E was used to estimate the maximum 3D measurement errors, whose minimum, maximum, and standard deviation values are also listed in Table 6.1 for comparison. Note that, since the values calculated by the use of E are unitless, they were scaled in such a way that the standard deviations are the same as the one derived with Gaussian noise added.

Recall that a good system configuration is one with a small *maximum* 3D measurement error (listed in the 3rd column with bold fonts in Table 6.1). Accordingly, we can see that the three optimal system configurations S_1^*, S_2^*, and S_3^* are better. Also, recall that the function E is proposed to estimate these maximum values for use as a criterion to find the optimal system configuration. The effectiveness of E in this aspect can be seen from the similarity of the maximum values **max** listed in the 3rd columns of Table 6.1 to those listed in the 6th column.

Table 6.1 3D Measurement Errors

System config.	3D measurement errors			Estimated 3D measurement error (by proposed method)		
	min [*cm*]	**max [*cm*]**	std. [*cm*]	min[a]	**max[a]**	std.[a]
S_1^*	0.395	**12.287**	1.965	4.127	**12.499**	1.965
S_2^*	0.393	**12.673**	2.000	3.860	**11.799**	2.000
S_3^*	0.385	**12.379**	1.985	4.004	**11.579**	1.985
S_1	0.342	**22.773**	3.235	1.655	**17.159**	3.235
S_2	0.314	**36.685**	5.326	1.078	**27.236**	5.326
S_3	0.362	**15.196**	2.223	2.704	**12.950**	2.223

[a]The values are scaled such that the standard deviations are the same as the ones derived with Gaussian noise added.

6.7 Comparisons with Existing Methods

Recalling that the proposed method to assess the 3D measurement error is based on analytic error propagation analysis, another approach found popular in the literature is to use the covariance matrix to assess the 3D measurement error [37]-[42] as surveyed previously. Four different methods of this approach were implemented by programs in this study. They are briefly introduced first here and then compared with the proposed method by experimental results in this section.

When using the covariance matrix to assess the 3D measurement error yielded by a binocular vision system, let P be a feature point in the space, (u_1, v_1) be the coordinates of the pixel P' corresponding to P in the left-camera image, and (u_2, v_2) be those of P' in the right-camera image. By mid-point triangulation [54], if the 3D location of P is calculated by a function $f(u_1, v_1, u_2, v_2)$, then, according to [60], the covariance matrix Σ_P of the measured 3D location data of P can be assessed by

$$\Sigma_P = (\partial f / \partial p)\Sigma_p (\partial f / \partial p)^T, \quad (102)$$

where p is the vector (u_1, v_1, u_2, v_2), Σ_p is the covariance matrix of p, and T denotes the operation of matrix transpose. The covariance matrix Σ_p can be estimated in complicated ways [37]-[41] or by constant values [38][42]. For simplicity, the matrix was estimated by an identity matrix in our implementations, and the first-order derivatives of f were obtained by a finite difference approach [61] with the difference taken to be 10^{-10}. After the covariance matrix Σ_P is derived, the four implemented methods use the following data of Σ_P to assess the 3D measurement error: (1) the determinant [37][38], (2) the trace [37][39], (3) the maximum eigenvalue [37][40], and (4) the maximum diagonal element [41]. These four different methods are named *DET*, *TR*, *MAXEIG*, and *MAXDIAG*, respectively, subsequently.

A simulation environment for comparisons of the proposed error model with others is constructed as follows. The 3D measurement area is defined to be rectangular-shaped with two corners located at (−1, 0, −1) and (1, 0, 1), including about 100,000 equidistant points for use as the 3D measurement locations. The two omni-cameras were placed at (±0.7, −0.1, 0), and the viewing angles of the used perspective cameras are 60° and the resolution of acquired images is 600×600. In each simulation, two omni-images were taken firstly, and the projections of each 3D measurement location L_i were extracted as two pixels l_i and r_i in the left and right omni-images, respectively. To simulate the imprecision introduced by the feature detection process, noise values within the range from −1.0 to 1.0 were introduced into the coordinates of the extracted pixels l_i and r_i. The mid-point triangulation process [54] was then conducted to compute the 3D position L_i' of each landmark point located at L_i using the coordinates of image pixels l_i and r_i. Since the coordinate values were interfered with noise, the calculated 3D position L_i' is slightly different from the ground truth L_i. With the recall that the measurement error is defined as the distance between the actual point and the measured one, the measurement error yielded by the simulation was computed accordingly to be $\|L_i - L_i'\|$.

The above simulation was conducted several times, and an average measurement error was calculated for each landmark point as plotted in Fig. 6.15(a). These average measurement errors are considered as ground-truth values, and were compared with the measurement errors calculated by the proposed error model and those proposed by

the four implemented existing methods shown in Figs. 6.15(b) through 6.15(f), respectively. The peak signal-to-noise ratio (PSNR) values and the running times of the five methods are listed in Table 6.2, from which one can see that the proposed error model yields the highest PSNR, and the *TR* method is the best of the four existing methods in this aspect but worse than the proposed model by a factor of $10^{(2.367 - 1.760)} = 10^{0.607} \approx 4.04$, and the running time of the proposed method is smaller than that of the *TR* method by a factor of 273.97/8.56 ≈ 32.01.

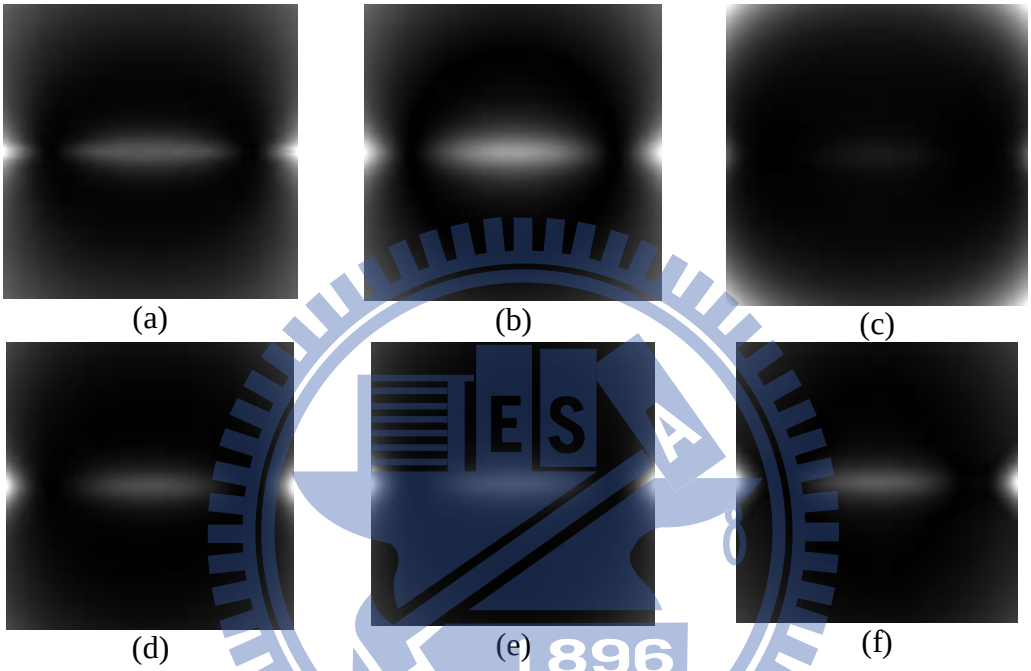


Fig. 6.15 Images for 3D measurement errors (with darker colors indicating smaller errors) calculated by (a) simulations, (b) proposed method, (c) method *DET* [37][38], (d) method *TR* [37][39], (e) method *MAXEIG* [37][40], and (f) method *MAXDIAG* [41].

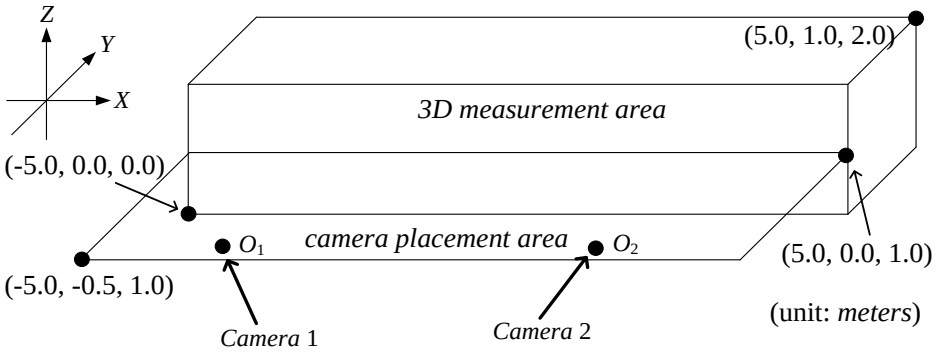
Table 6.2 PSNR Values and Running Times in the Simulation

Method	PSNR	Running Time
Proposed error model	23.67dB	8.56 milliseconds
Method *DET* [37][38]	12.07dB	273.22 milliseconds
Method *TR* [37][39]	17.60dB	273.97 milliseconds
Method *MAXEIG* [37][40]	15.32dB	344.03 milliseconds
Method *MAXDIAG* [41]	15.24dB	277.24 milliseconds

Next, we describe the derivation of the optimal configuration of a vision system for a real laboratory environment to compare the proposed optimization methods with the conventional ones. In the environment, a user was allowed to move freely in a specified 3D measurement area, and the two omni-cameras were placed within a specified camera placement area. The optimal positions and orientations of the two omni-cameras were computed for this environment by the proposed method and the four existing methods mentioned previously.

Specifically, as shown in Fig. 6.16(a), the floor of the environment is the *XY*-plane, the 3D measurement area is the cuboid with two diagonal points being (5.0, 2.0, 2.0) and (–5.0, 0.0, 0.0), and the camera placement area is the rectangle on the plane $Z = 0$ with two diagonal points being (5.0, 0.0, 1.0) and (–5.0, –0.5, 1.0). The goal of the optimization algorithm is to find the optimal positions O_1 and O_2 of the two omni-cameras, and their optical axes directions a_1 and a_2, such that the 3D measurements of an object located in the 3D measurement area are as accurate as possible. The two cameras are hyperboloidal catadioptric ones with eccentricity ε being 1.6571, the viewing angle of the perspective camera is $2\tau_{\max} = 38°$, and the omni-image size is 600×600.

The optimization process was conducted firstly for a general case. At first, some points, for use as 3D measurement locations, were sampled within the 3D measurement area with a fixed interval 10*cm*, and some points, for use as the camera placement locations, were sampled within the camera placement area with a fixed interval 1*cm*. Then, for each possible positions O_1 and O_2 of the cameras, the directions of the optical axes were derived according to the proposed method, and the 3D measurement error were assessed by the error model proposed previously. Finally, the poses of the two omni-cameras which yield the minimum 3D measurement error were taken to be the parameters of the best system configuration. The result of this process says that two cameras should be placed at (±3.78, –0.5, 1.0), respectively, and the optical axes be oriented in accordance with the vectors ($\mp$0.14, 0.99, 0.0), respectively.



(a)

(b) (c)

Fig. 6.16 Environment where experiments were conducted. (a) An illustration. (b)(c) Two omni-images captured in the environment.

To test the 3D measurement accuracy using such a system configuration, a checker board were placed on the planes with Y = 0.0, 1.0, and 2.0 in the 3D measurement area, as shown in Figs. 6.17(a) and 6.17(b). The image pixels corresponding to all the *cross points* were manually picked out from the captured omni-images, and the obtained coordinates of these pixels were disturbed with additive noise within the range [–5, 5] to simulate errors introduced by the feature detection process. Then, by mid-point triangulation, the 3D data of the cross points were derived, called the *measured data*. Finally, the 3D measurement errors were taken to be the distances between the measured data and the ground-truth data, the latter being measured manually in advance. The 3D measurement errors of the cross points on the calibration board at plane Y = 0.0 are drawn in Fig. 6.17(c), whose shape, as can be found, is consistent with that of Fig. 6.6(b) or Fig. 6.15(b), though depicted in different ways. Also, these results of the proposed method are listed in Table 6.3 for comparison with those obtained similarly of the aforementioned four existing

methods. As can be seen from the table, the proposed method yields the minimum measurement errors, and runs faster than the others for about 20 times.

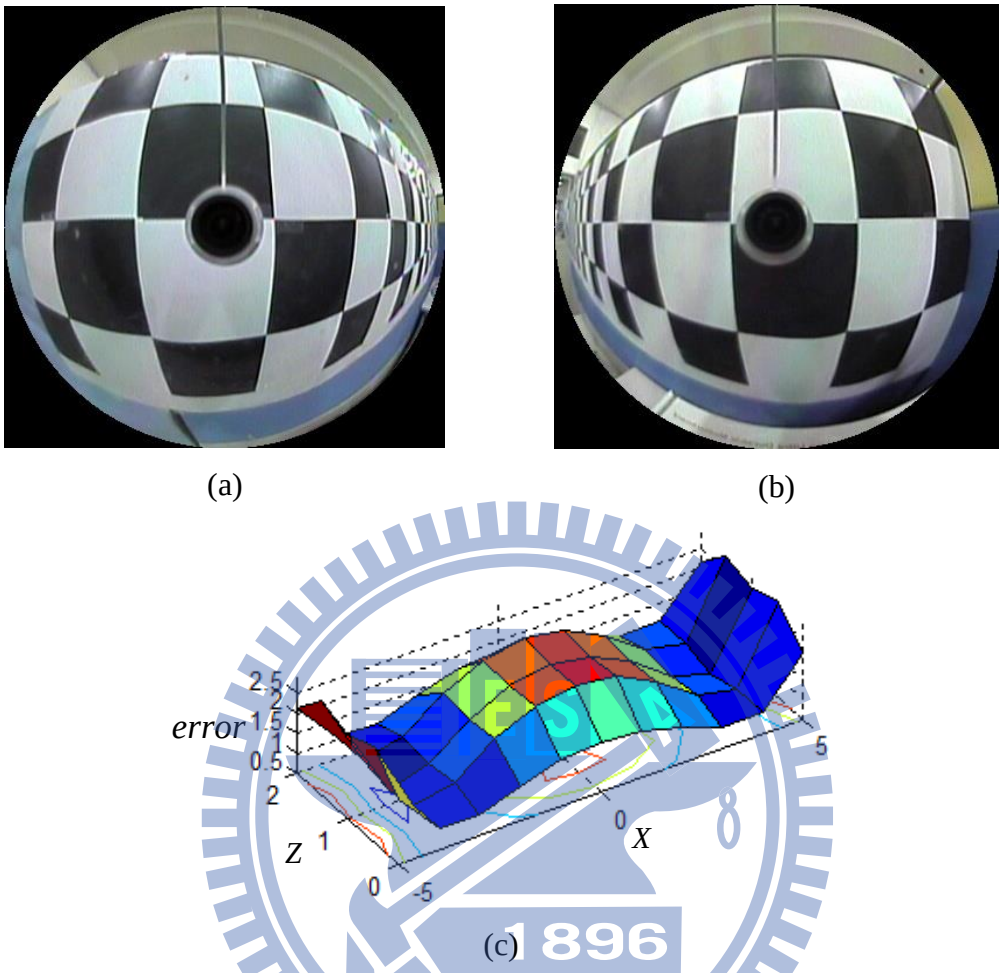


Fig. 6.17 Testing the 3D measurement accuracy of the derived best system configuration. (a)(b) A checkerboard is placed at $Y = 0$ to test the accuracy. (c) The 3D measurement errors of the points on the board.

Table 6.3 The Optimization Results of the Methods

Method	Camera Positions (unit: *meter*)	Maximum 3D Measurement Error (unit: *cm*)	Run Time (unit: *sec*)
Proposed method	(±3.78, –0.5, 1.0)	26.290	11.413
Method *DET* [37][38]	(±4.02, –0.5, 1.0)	29.185	216.927
Method *TR* [37][39]	(±3.95, –0.5, 1.0)	28.007	217.241
Method *MAXEIG* [37][40]	(±3.94, –0.5, 1.0)	27.843	254.599
Method *MAXDIAG* [41]	(±3.88, –0.5, 1.0)	26.877	220.423

6.8 Conclusion

The issue of designing the optimal configuration of a stereo vision system with two catadioptric omni-cameras to compute 3D data with minimum errors is investigated in this study. The solution includes the poses and the mirror-shape parameters of the omni-cameras. An analytic formula is derived to model the 3D measurement error, which takes into consideration the error propagation in the data computation process. Two fast and elegant optimization algorithms have been designed accordingly for "regular" environments with rectangular cuboid-shaped 3D measurement and camera placement areas. One of them, based on a bisection scheme, is optimal but relatively slower, which may be used for off-line applications. And the other, using analytic formulas to calculate approximate solutions, is faster for realtime applications with the computed precision being sub-optimal but close to that of the former. An algorithm for dealing with general environments with irregular-shaped 3D measurement and camera placement areas has also been developed for general uses. Experimental results show the feasibility of the proposed method.

In real applications, a manufacturer may produce omni-cameras according to the derived optimal mirror shape. Then, a consumer may bring them back and deploy them in the optimal or nearly-optimal pose using the proposed algorithms. As a result, a stereo vision system which yield precise 3D measurement results can be set up. Future studies may be directed to generalizing the proposed optimization method to a stereo vision system with more than two omni-cameras.

Chapter 7 A Convenient Vision-based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-angle Cameras

The details of the proposed convenient indoor vision-based parking lot system using wide-angle fisheye-lens or catadioptric cameras is described in this chapter, which is easy to set up by a user with no technical background. Easiness in the system setup comes mainly from the use of a new camera model which can be calibrated using only one space line without knowing its position and direction, as well as from the allowance of convenient changes of the detected parking space boundaries. After camera calibration based on the new camera model is completed, parking-space boundary lines are extracted automatically from input wide-angle images by a modified Hough transform with a new cell accumulation scheme, which can generates more accurate equal-width curves using the geometric relations of line positions and directions. Also, the user may easily add or remove the boundary lines by single clicks on images, and parking spaces can be segmented out by region growing by the use of the boundary lines. Finally, vacant parking spaces can be detected by a background subtraction scheme. A real vision-based parking lot has been established and relevant experiments conducted. Good experimental results show the correctness, feasibility, and robustness of the proposed methods.

In the following, an overview of the proposed method is first described. Then, the details of the proposed camera model and calibration scheme are described next, followed by the proposed space line detection method and the proposed parking space segmentation and vacancy detection techniques. Three different series of experiments are also stated and discussed, followed by some conclusions provided in the last section.

7.1 Overview of Proposed Method

The indoor parking lot system proposed in this study utilizes multiple wide-angle cameras affixed on the ceiling and looking downward vertically. The system operations can be divided into four stages as shown in Fig. 7.1: camera calibration,

space line detection, parking space segmentation, and vacancy detection. In the first stage, a new camera model is proposed, leading to the uses of *fewer* parameters than conventional models and a *single* space line for calibrating the model. This is why the proposed calibration process can be done by a user with no technical background. In the second stage, space lines appearing in the captured image are detected by an edge detection process, followed by a Hough transform based on a *new* cell accumulation scheme. *Equal-width* curves are so generated, leading to more precise space line detection results for use as the boundary lines of the parking spaces.

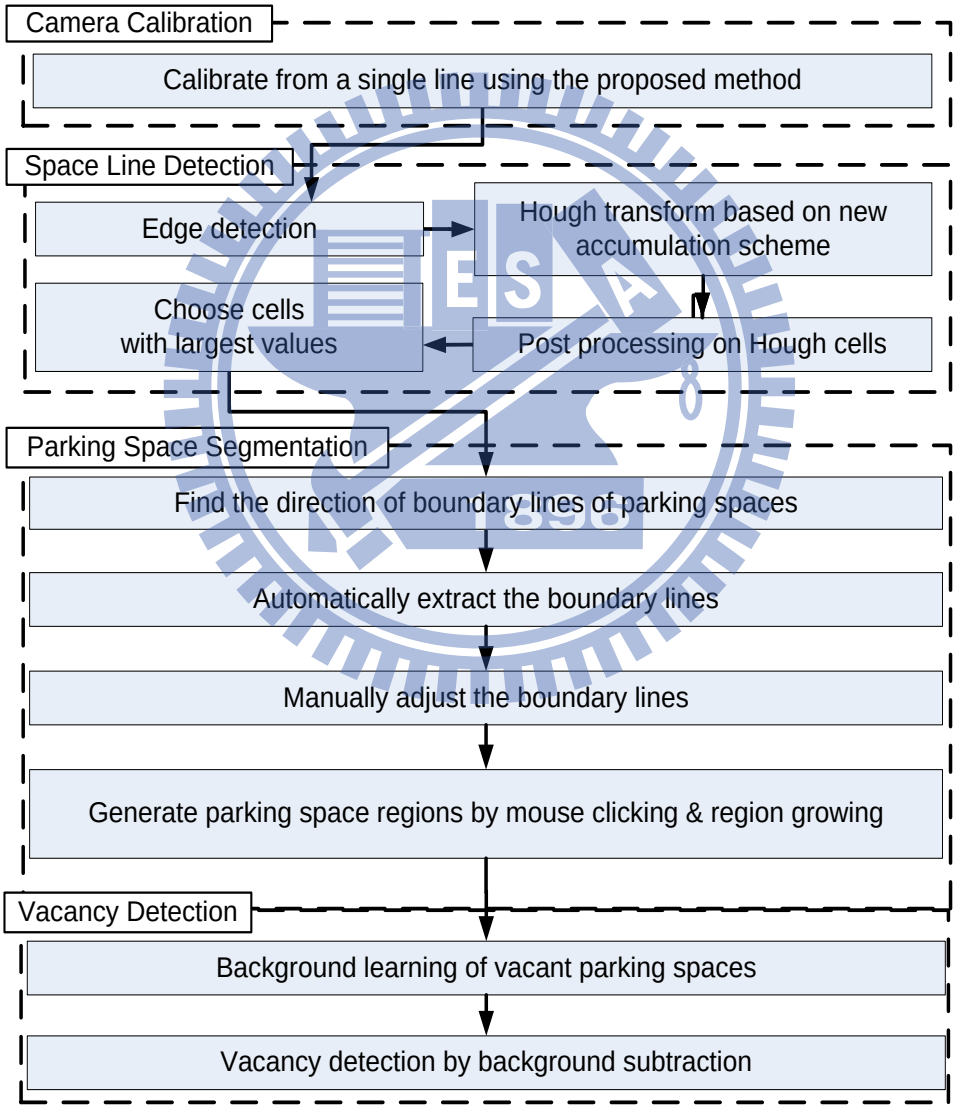


Fig. 7.1 The proposed parking lot system.

In the third stage, the detected boundary lines are analyzed and displayed automatically, and the user can then simply click on the image to add or remove the boundary lines. In addition, by clicking on any position within each parking space, the space region is generated by a region growing process. Finally, after "learning" the background parameters of the vacant parking spaces for various environment conditions, vacant spaces can be found by background subtraction for uses in parking management.

The proposed vision-based parking lot system has at least the following merits: 1) wide-angle cameras are used so that *less* cameras are needed to cover the area of a given parking lot; 2) the camera system can be calibrated using *only one line* in the environment, so that the system can be set up easily by a user with no technical background; 3) the proposed Hough transform with the new cell accumulation scheme generates equal-width curves, so that the proposed system is capable of dealing directly with distorted images captured by wide-angle cameras and the space line detection results are more precise than those yielded by conventional methods; 4) unlike many previous studies which specify parking spaces *manually*, the proposed method detects them *automatically* for convenient system setup as well.

7.2 Proposed Calibration Method using Only One Space Line

To design an easy-to-setup vision-based parking lot system, the camera calibration process must be easy to carry out by normal users with no technical knowledge. In this section, a simplified camera model is proposed for this aim. A calibration method is proposed accordingly which makes use of *only one* space line in the environment without knowing its position and direction.

The proposed modified unifying camera model is based on the use of an optimal approximation value of the parameter l which is the distance from the effective viewpoint O to the pinhole point O_c as shown in Fig. 7.2. The model has two merits: 1) it reserves important characteristics of space lines as shown in this section; and 2) it can be calibrated easily by the use of a single space line as described later in this dissertation. These merits make the corresponding system setup process easy to conduct.

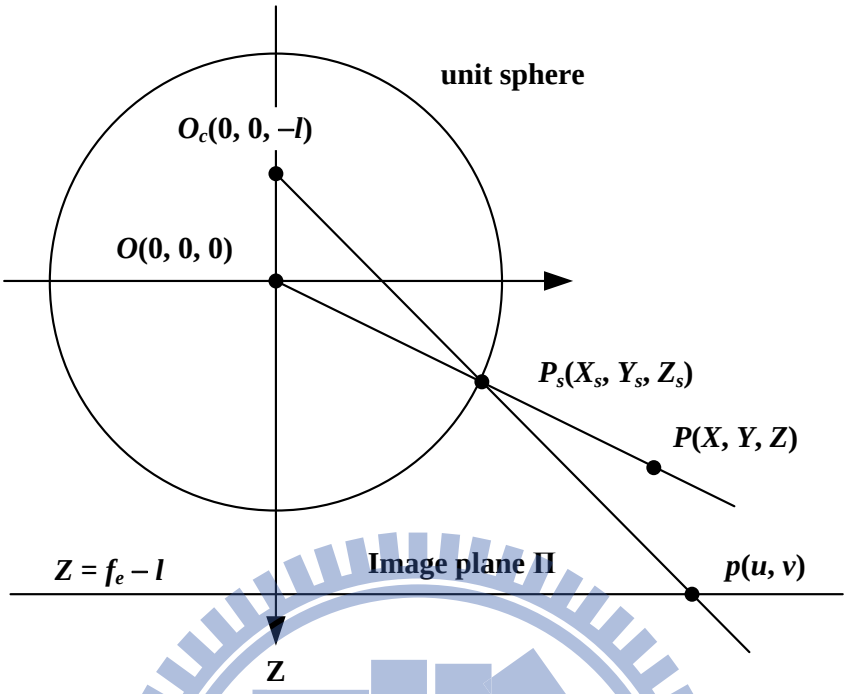
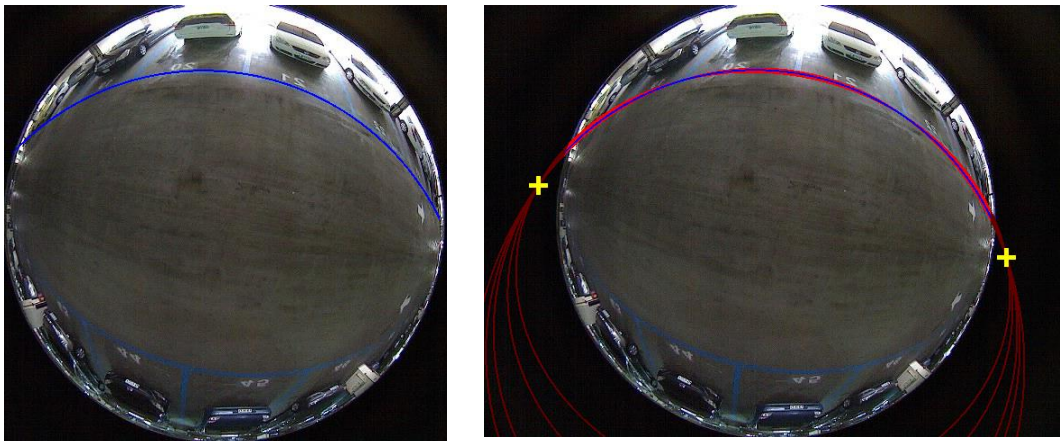


Fig. 7.2 An illustration of a two-step spherical mapping.

The rationale of finding a fixed optimal value of parameter *l* can be explained as follows. In Fig. 7.3(a), the image of a space line, called *line image* hereafter, is marked as a blue curve; and in Fig. 7.3(b), this line image is shown to be fit well enough by conic sections with different values of the parameter *l* while the two vanishing points are fixed (marked in yellow in the figure). This phenomenon leads to two conclusions: 1) the parameter *l cannot* be well calibrated from line images; and 2) reversely, the value of the parameter *l did not* affect the space line detection process. The first conclusion is consistent with some previous studies [46][48]. Specifically, the parameter *l* was fixed in the simulation experiments described in Geyer and Daniilidis [46], so the parameter *l* was not derived in the calibration process; and as seen in Deng et al. [48], the parameter *l* is assumed to be known before the calibration. The second conclusion makes it possible to find an optimal value of the parameter *l* to approximate that of any kind of wide-angle camera, without affecting the space line detection process.



(a) (b)

Fig. 7.3 Fitting a space line using different values of l. (a) Space line to be fit (marked as blue). (b) Fitting results using l = 2.0, 1.5, 1.0, 0.8, and 0.5 with larger ellipses corresponding to smaller values of l, and the two yellow crosses indicate the fixed vanishing points.

To find the optimal value of l, we define first the range of parameter l of each kind of wide-angle camera. For parabolic catadioptric cameras, the value of l is known to be 1.0 [44]; for hyperbolic catadioptric cameras, the value of l is smaller than 1.0 and larger than 0.0 [44]; and for fisheye-lens cameras, the value of l is larger than 1.0 [47]. In this study, we define the interesting range of the parameter l to be $0.5 < l < 2.0$, which includes the commonly used values of l. For example, the values of l derived in [46][48] are 0.8, 0.966 and 1.0, that derived in [49] is 0.9663, and that derived in [50] is 1.07.

The optimal value l^* found by a simulation process as described in the following.

1) Generate simulated line images I_i with size 1000×1000 for a set of sampled values of l in the interesting range $0.5 < l < 2.0$ and for a set of sampled positions and directions of space lines;
2) For each sample value l_j^* in the range $0.5 < l_j^* < 2.0$, do the following steps
   2.1) find the best-fitting curve I_{ij}^* to each line image I_i, with $l = l_j^*$ by a Levenberg–Marquardt process;
   2.2) compute the distance $dist_{ij}$ between I_i and I_{ij}^* as

$$dist_{ij} = \frac{1}{n}\sum_{k=1}^{n}\left\| p_k - p_k^* \right\| \tag{103}$$

where n denotes the number of pixels on the line image I_i, p_k a pixel on I_i, p_k^* the nearest pixel on I_{ij}^* to p_k, and $\|p_k - p_k^*\|$ the distance between p_k and p_k^*;

2.3) calculate the average distance $dist_j$ of all $dist_{ij}$ of all I_i as a measure of optimality of the sampled value l_j^*, with a smaller $dist_j$ meaning a better fit of l_j^* to all I_i.

3) Choose as the desired optimal value l^* the l_j^* with the smallest $dist_j$ which is called also the *average fitting error* and denoted as *dist* subsequently.

An experimental result of the above process is shown in Fig. 7.4, where Fig. 7.4(a) shows a line image I_i marked in blue and a best-fitting curve I_{ij}^* marked in red; Fig. 7.4(b) shows the trend of the value of $dist_j$ for different l_j^* values, from which it can be seen that an optimal value of l_i^* does exist and is located at 1.24 for choice as l^*, and that the line images can be well approximated by Eq. (2) with $l = l^*$ resulting in an average fitting error $dist \approx 1.1941$ pixels.

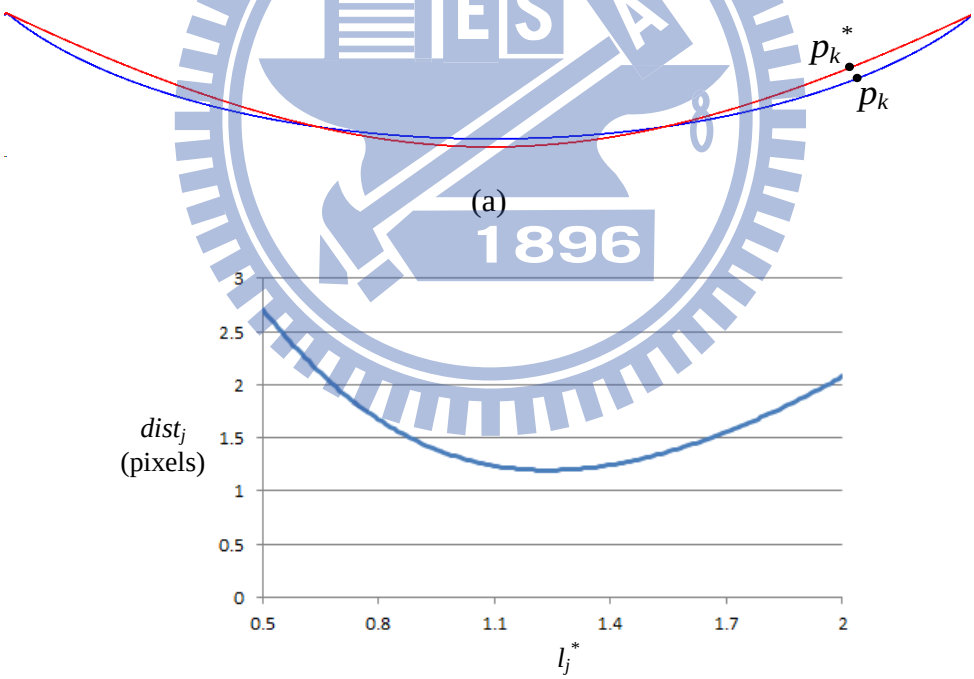


Fig. 7.4 Finding the optimal value l^*. (a) A line image I_i (marked in blue) and its best-fitting curve I_{ij}^* (marked in red). (b) The trend of the average $dist_j$ of the distances between the best-fitting curve and the line images. The optimal value of l_j^* is 1.24, with $dist_j = dist$ being about 1.1941 pixels.

As a result of utilizing the aforementioned second conclusion, we propose a camera model which is identical to the unifying one proposed in [44] but with its parameter l fixed to be the optimal value 1.24 derived previously. One merit of this model is that it leads to the possibility of calibrating wide-angle cameras using only one line image. This property is a great advantage over the conventional models [45][44], by which the camera cannot be calibrated reliably from line images as proved previously. It also facilitates a non-technical user to conduct the calibration process without difficulty as mentioned previously.

Based on the proposed camera model using the fixed parameter l = 1.24, the idea of the proposed calibration process using a single line can be divided into three steps. First, a space line is chosen with its line image (in the shape of a conic-section curve) marked manually. Then, the best-fitting ellipse to this line image is computed, from which the unknown camera parameters and the space line are estimated roughly. Finally, a Levenberg–Marquardt algorithm is conducted to find the precise values of the camera parameters.

In more detail, let L be a chosen space line, I_L its line image, and E_L the best-fitting ellipse to I_L. As derived in [45][46], I_L can be expressed as

$$F(u,v)=\begin{bmatrix}u\\v\\1\end{bmatrix}^T\begin{bmatrix}\left(l^2-1\right)n_x{}^2+l^2n_z{}^2 & \left(l^2-1\right)n_xn_y & -f_en_zn_x\\ \left(l^2-1\right)n_xn_y & \left(l^2-1\right)n_y{}^2+l^2n_z{}^2 & -f_en_zn_y\\ -f_en_zn_x & -f_en_zn_y & -f_e{}^2n_z{}^2\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}=0, \quad (104)$$

where (u, v) are coordinates of pixels on I_L, (n_x, n_y, n_z) is the normal vector of the plane Π formed by line L and the origin of the camera's lens center, and (l, f_e) are the parameters used in the unifying camera model as described in Sec. II.A. Also, let the ellipse E_L be described by

$$\begin{bmatrix}u\\v\\1\end{bmatrix}^T\begin{bmatrix}a & b & d\\ b & c & e\\ d & e & f\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix}=0. \quad (105)$$

Note that when $n_z = 0$, the line image is a straight one going through the image center so that the parameter f_e, which is the *effective focal length* of the camera, cannot be calibrated [46][62]. Ignoring this, we may rewrite Eq. (104), after dividing it by $n_z{}^2 \neq 0$, to be

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}^T \begin{bmatrix} (l^2-1)G^2+l^2 & (l^2-1)GH & -f_eG \\ (l^2-1)GH & (l^2-1)H^2+l^2 & -f_eH \\ -f_eG & -f_eH & -f_e^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0, \tag{106}$$

where $G = n_x / n_z$, and $H = n_y / n_z$.

Now, the problem is to estimate *roughly* the values of the parameters (l, f_e, G, H) such that Eqs. (105) and (106) are close to each other. Since only rough estimation is needed, we first simplify (106) by assuming l = 1.0. Accordingly, the problem is reduced to finding the parameters (f_e, G, H) which satisfy

$$\begin{bmatrix} 1 & 0 & -f_eG \\ 0 & 1 & -f_eH \\ -f_eG & -f_eH & -f_e^2 \end{bmatrix} \sim \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}, \tag{107}$$

where "~" means "equals up to a scale." Let λ be the hidden unknown parameter for this scaling. Then, we have

$$\begin{bmatrix} \lambda & 0 & -\lambda f_eG \\ 0 & \lambda & -\lambda f_eH \\ -\lambda f_eG & -\lambda f_eH & -\lambda f_e^2 \end{bmatrix} = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}. \tag{108}$$

One solution to the above equation for use as rough estimates of the parameters is:

$$\lambda = \frac{a+c}{2};\ f_e = \sqrt{-\frac{f}{\lambda}};\ G = \frac{-d}{\lambda f_e};\ \text{and}\ H = \frac{-e}{\lambda f_e}. \tag{109}$$

A Levenberg–Marquardt process is conducted finally to derive the precise values of (f_e, G, H), with the initial values being specified by (109) and the criterion being to minimize the value of

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}^T \begin{bmatrix} (l^2-1)G^2+l^2 & (l^2-1)GH & -f_eG \\ (l^2-1)GH & (l^2-1)H^2+l^2 & -f_eH \\ -f_eG & -f_eH & -f_e^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{110}$$

with respect to all the pixels (u, v) on the line image I_L. After this optimization process is done, the parameter f_e of the camera model is derived, completing the calibration process (as shown in Fig. 7.5).

Four results of this calibration process are shown in Fig. 7.5. The calibrated values f_e for Figs. 7.5(a) and 7.5(b) are 319.90 and 319.57, respectively, and those for

Figs. 7.5(c) and 7.5(d) are 266.73 and 269.53, respectively. The validity of the proposed calibration method can be shown by the good fitness of the best-fitting ellipse to the manually-marked line image in each case, and the closeness of the calibrated values f_e in the first two cases using a fisheye-lens camera and in the remaining two cases using a hyperbolic catadioptric camera.

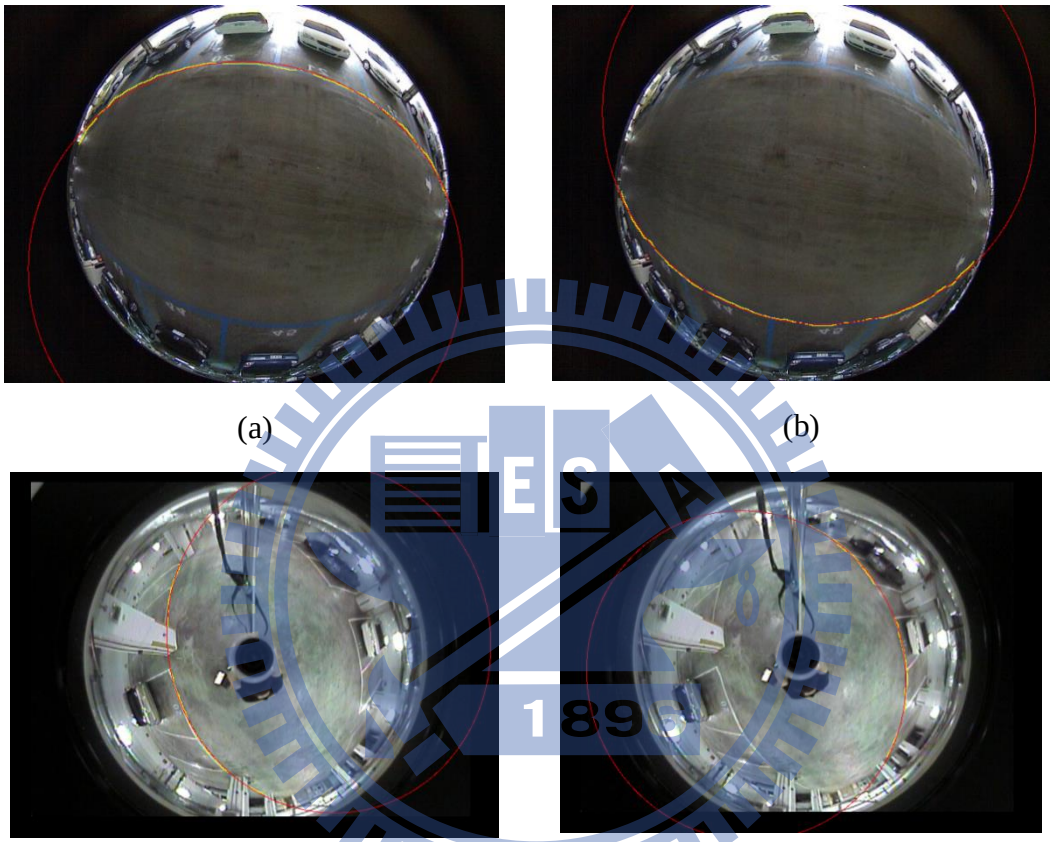


Fig. 7.5 Calibration results with yellow curves indicating manually-marked pixels, and red ellipses being the best-fitting results. (a)(b) Results using a fisheye-lens camera with calibrated values f_e being 319.90 and 319.57, respectively. (c)(d) Results using a hyperbolic catadioptric camera with calibrated values f_e being 266.73 and 269.53, respectivley.

7.3 Review of the Proposed Space Line Detection Method

One of the important features in man-made environments is straight line, and as formulated by (104), these space lines are projected by wide-angle cameras to form conic sections in the resulting image, called the *line images* so far. In this section, the

proposed method for detecting the conic-section curve is described, which, differently from conventional methods, generates *equal-width regions* along a curve, as shown in Fig. 7.6. As a result, the proposed method can detect *thick curves* in images more reliably, overcome the noise produced by edge detection, and be utilized in accordance with the proposed camera model (described in a previous chapter) for more accurate detection of parking-space boundary lines.

The proposed space line detection method is based on the Hough transform with a new cell accumulation scheme to achieve the ability to detect equal-width thick curves. As described by (104), a line image, which appears to be a conic section in the acquired wide-angle image, can be parameterized by the normal vector (n_x, n_y, n_z) of the plane Π formed by the corresponding space line and the origin of the camera. The normal vector can be expressed as

$$\left(n_x, n_y, n_z\right) = \left(A, B, \sqrt{1-A^2-B^2}\right), \tag{111}$$

where

$$A = \frac{n_x}{\sqrt{n_x^{\,2} + n_y^{\,2} + n_z^{\,2}}}, \quad B = \frac{n_y}{\sqrt{n_x^{\,2} + n_y^{\,2} + n_z^{\,2}}}, \tag{112}$$

and $0 \le A, B \le 1$. The parameters (A, B) are quantized into $n \times n$ values to form *Hough cells* in this study, with the line images (conic-section curves) corresponding to the cells being described by the equation $F(u, v) = 0$ in (104), and the values (n_x, n_y, n_z) in (104) defined by (111) and (112).

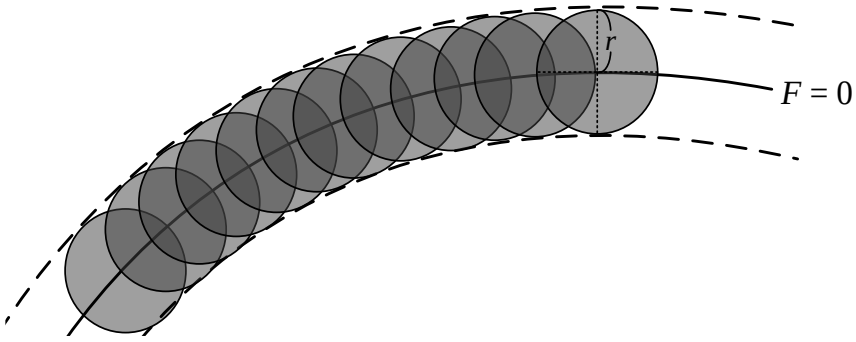


Fig. 7.6 An illustration of an equal-width curve along a curve $F = 0$ with width r, defined to be the regions of all the gray circles (not all drawn), or equivalently, the thick area bounded by the two dashed curves.

The first step in the proposed detection method is to obtain a binary version I_b of the input wide-angle image by edge detection. Then, for each Hough cell at parameters (A, B) and for each 'black' pixel p in B with coordinates (u, v), if p is in the equal-width curve region of the line specified by the parameter (A, B), then the cell value at (A, B) is incremented by one.

In more detail, by referring to Fig. 7.7, it can be figured out that a pixel (u, v) is in the equal-width region with width r if and only if the curve $F = 0$ intersects the circle C centered at (u, v) with radius r. Furthermore, according to the intermediate value theorem [63], the curve $F = 0$ intersects the circle C if there exists a point $(u', v') = (u + \Delta u, v + \Delta v)$ on C such that $F(u', v')$ and $F(u, v)$ are with opposite signs. Accordingly, if $F(u, v)$ is *positive*, we try to find the point (u', v') on C with the *minimum* value of $F(u', v')$, and then determine whether $F(u', v')$ and $F(u, v)$ are with opposite signs. Reversely, if $F(u, v)$ is *negative*, we try to find the point (u', v') on C with the *maximum* value of $F(u', v')$, and then determine whether $F(u', v')$ and $F(u, v)$ are with opposite signs. For the first case with $F(u, v) \geq 0$, since $F(u', v')$ should be the *minimum* value and since the gradient ∇F specifies the direction of *increasing* the function value $F(u, v)$, the vector $(\Delta u, \Delta v)$ should be in the *negated* direction of the gradient ∇F. So, under the constraint that the length of $(\Delta u, \Delta v)$ is the radius r, the vector $(\Delta u, \Delta v)$ may be expressed as $[-\nabla F(u, v)/\|\nabla F(u, v)\|]\times r$. For the second case with $F(u, v) < 0$, since $F(u', v')$ should be the *maximum* value, it can be derived by a similar reasoning that the vector $(\Delta u, \Delta v)$ should be with the *same* direction of the gradient ∇F so that it can be expressed as $[\nabla F(u, v)/\|\nabla F(u, v)\|]\times r$. As a summary, we have

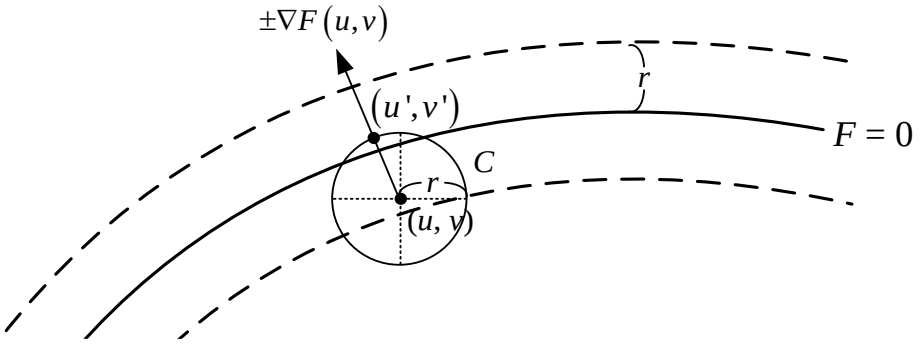


Fig. 7.7 An illustration for determining whether a pixel at (u, v) lies inside an equal-width curve or not.

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \text{ and } \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{cases} \dfrac{-\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|} r & \text{if } F(u,v) \geq 0 \\ \dfrac{\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|} r & \text{if } F(u,v) < 0 \end{cases}. \tag{113}$$

Accordingly, the function value $F(u', v')$ can now be derived by the 2nd-order Taylor expansion to be:

$$F(u',v') = F(u,v) + \nabla F(u,v)^T \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}^T HF(u,v) \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \tag{114}$$

where $HF(u, v)$ is the Hessian matrix of function F described by

$$HF = \begin{bmatrix} \dfrac{\partial^2 F}{\partial u^2} & \dfrac{\partial^2 F}{\partial u \partial v} \\ \dfrac{\partial^2 F}{\partial u \partial v} & \dfrac{\partial^2 F}{\partial v^2} \end{bmatrix} \tag{115}$$

whose details may be derived from (104) and are omitted here.

In conclusion, to detect an equal-width line in a binarized wide-angle image I_b by the proposed Hough transform, we conduct the following process.

1) For each pixel p with coordinates (u, v) in I_b and for each Hough cell with parameters (A, B), calculate $(\Delta u, \Delta v)$ and (u', v') by (113), and $F(u, v)$ and $F(u', v')$ by (104) and (114), respectively.
2) If $F(u, v)$ and $F(u', v')$ are with opposite signs, decide the pixel p at (u, v) to be in the equal-width region of the line specified by the parameters (A, B), and increment the value of the Hough cell with (A, B) by one.
3) Find out the parameters (A, B) of the desired line images (conic-section curves) from the resulting Hough space by the following steps.
   3.1) Enhance the local maximums of the Hough space by applying the following sharpening filter:

$$\frac{1}{25} \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}. \tag{116}$$

   3.2) Pick out all the Hough cells with values larger than a pre-selected threshold δ and take the cells' parameters (A, B) for use as those of the

detected line images.

4) Use the derived parameters (A, B) to draw the conic sections of the detected line images on the input image as the output.

By the way, the curve width r used in (113) is defined by considering the error introduced in the edge detection process as well as the error introduced by the approximation of the parameter l. The former is approximated to be 1 pixel, and the latter 1.2 pixels, according to the data shown in Fig. 7.4. As a result, the width r used in this study is taken to be $r = 1.0 + 1.2 = 2.2$ pixels.

A result of the above line image detection method is shown in Fig. 7.8. The input image captured by a hyperbolic catadioptric camera is shown in Fig. 7.8(a), with the Canny edge detection result obtained from it shown in Fig. 7.8(b). The generated Hough space is shown in Fig. 7.8(c), and the result of applying the filter (116) in Fig. 7.8(d). Finally, some cells with larger values were picked out, and the line images corresponding to the cells' parameters (A_i, B_i) were drawn as shown in Figs. 116(e) and (f), where the numbers of drawn lines are 30 and 50, respectively.

Next, the effectiveness of the proposed cell accumulation scheme is analyzed more thorough as follows. As stated previously, each Hough cell is specified by two parameters A and B, and corresponds to a conic-section curve $F(u, v) = 0$ in the input image. Since the parameters A and B and the image coordinates u and v are all quantized to be discrete, the function value $F(u, v)$ for a certain coordinate pair (u, v) may *not* be exactly zero. A conventional way to deal with this problem is to define a threshold T and consider pixels with coordinates (u, v) satisfying the following inequality as being on the conic section described by $F(u, v) = 0$:

$$\left|F\left(u,v\right)\right| \le T \,. \tag{117}$$

Two results yielded by this method are shown in Figs. 7.9(a) and (b) with different thresholds $T = 40$ and 150, respectively. As can be seen, the generated conic sections are *not* with equal widths. This phenomenon results from some undesirable operations which cause some edge pixels not belonging to the conic section to be accumulated into wrong Hough cells, and vice versa. In contrast, the proposed method described in Sec. III.A yields equal-width curves as shown in Figs. 7.9(c) and (d) for different curve widths $r = 1$ and 2 pixels, respectively. These results show the capability of the proposed method for detecting equal-width curves, which is superior

to conventional methods and suitable for applications requiring higher line detection accuracy.

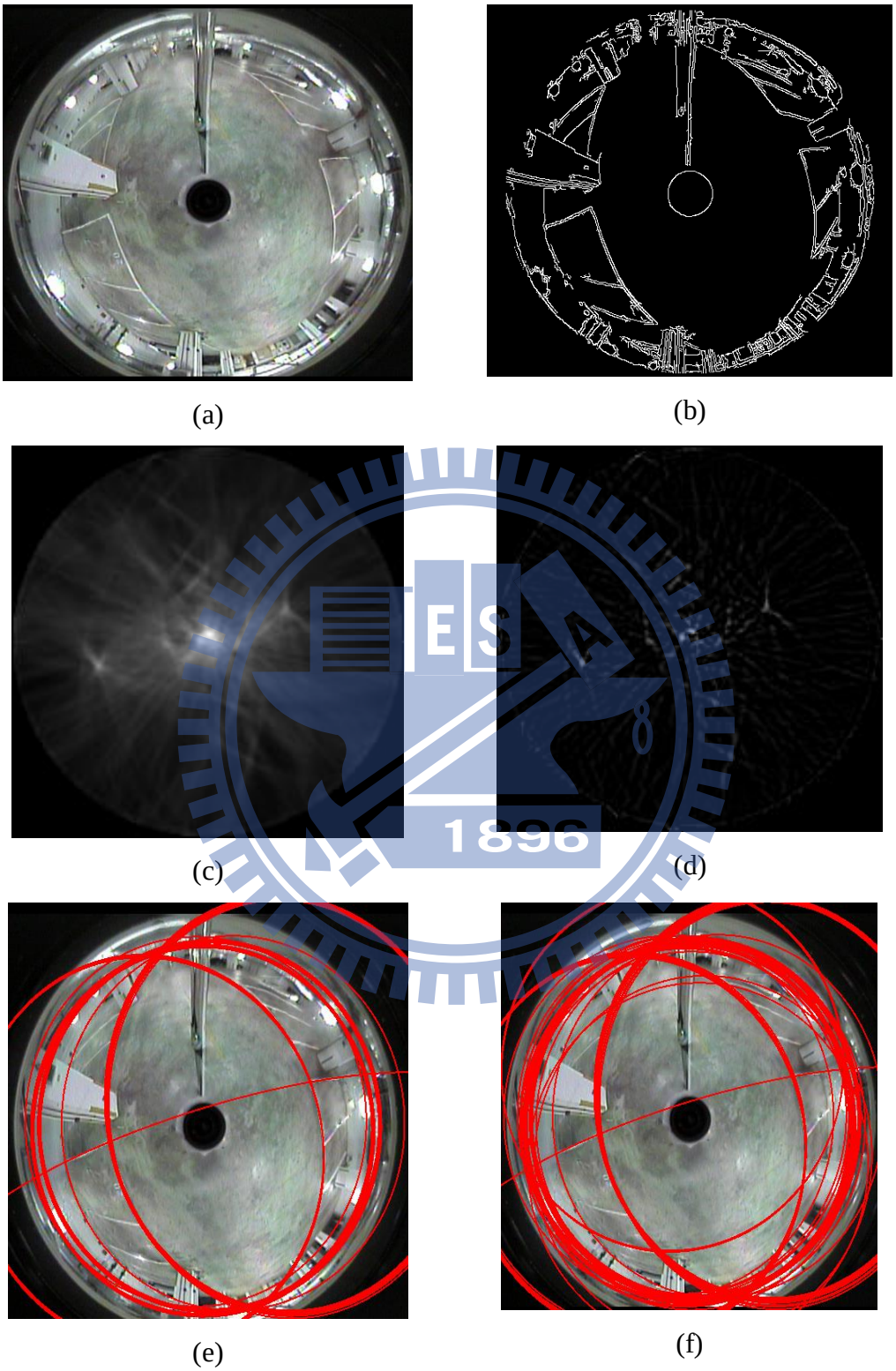


Fig. 7.8 Results of the proposed space line detection method. (a) The input image. (b) Canny edge result. (c) The Hough space. (d) The Hough space after applied the filter (116). (e)(f) Results of 30 and 50 lines, resepctively.

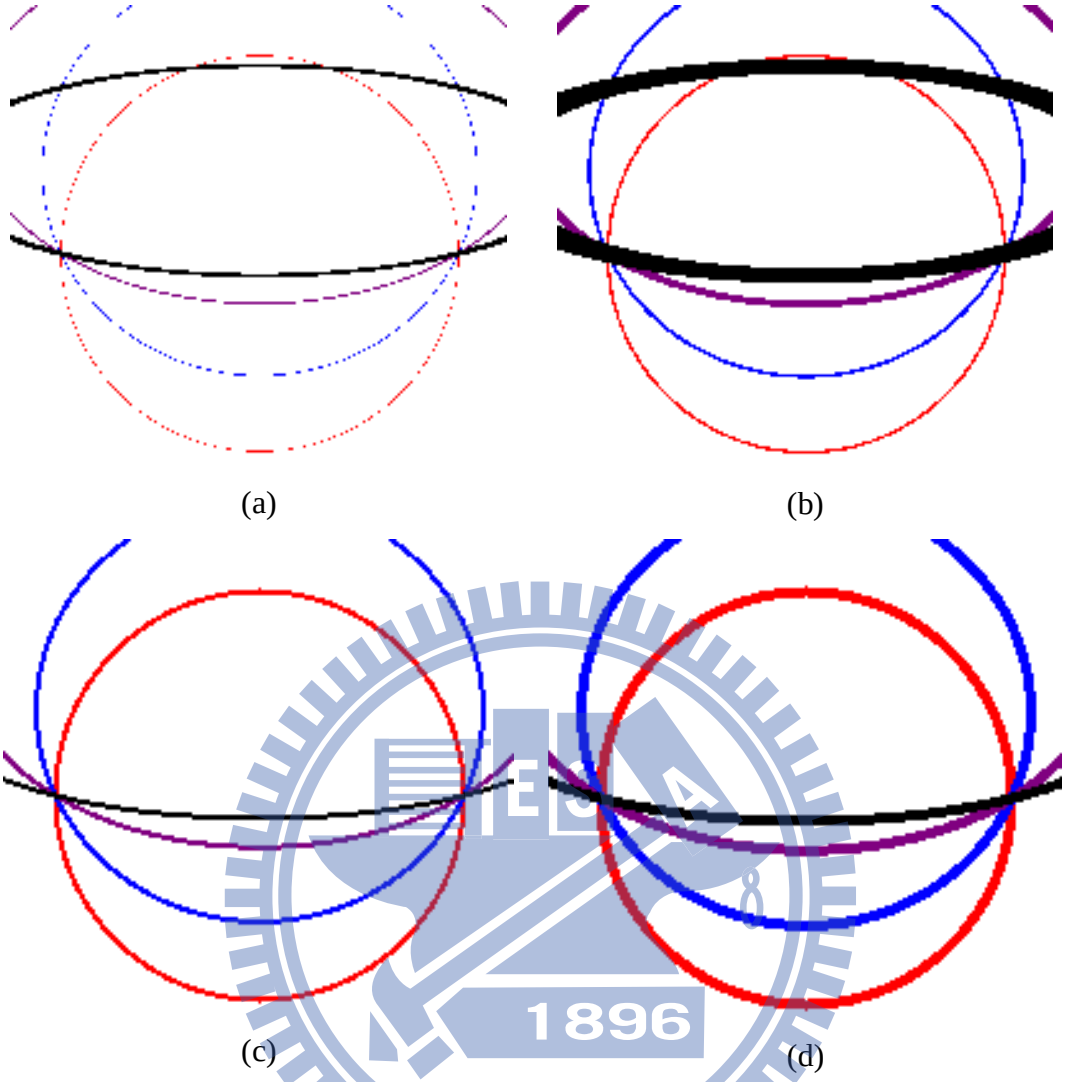


Fig. 7.9 Comparison of conventional and proposed accumulation method. (a)(b) Non-equal-width results yielded by conventional method with thresholds T = 40 and 150, respectively. (c)(d) Equal-width results yielded by proposed method with width r = 1 and 2, respectively.

Some experiments were also conducted to compare the space line detection results yielded by the conventional Hough cell accumulation scheme and those by the proposed method. An input image and the corresponding Canny edge detection result are shown in Figs. 7.10(a) and (b), respectively. The 10 and 30 detected space line images yielded by the conventional scheme are drawn in Figs. 7.10(c) and (d), respectively, and those yielded by the proposed method are shown in Figs. 7.10(e) and (f), respectively. As can be found from the figures, the curves detected by the conventional scheme mostly go through the image center, which are not good results.

This undesirable phenomenon may be explained from the simulation results shown in Fig. 7.9. Specifically, as can be seen from Fig. 7.9(b), the conventional scheme generates a thicker curve when the real curve appears nearer to the image center, and vice versa. This means that each Hough cell corresponding to a curve nearer to the image center will collect "votes" from a *thicker line band* in the input image (i.e., *more* edge pixels will be "accumulated" into such a cell), and vice versa. Therefore, when using the conventional scheme to detect curves by picking up Hough cells with *larger* votes, those curves nearer to the image center are *more* likely to be chosen first, yielding the results as shown in Figs. 7.10(c) and (d). In contrast, since the proposed method generates equal-width curves *everywhere*, this phenomenon does not appear in the detection results yielded by the proposed method as shown in Figs. 7.10(e) and (f).

7.4 Parking Space Segmentation and Vacancy Detection

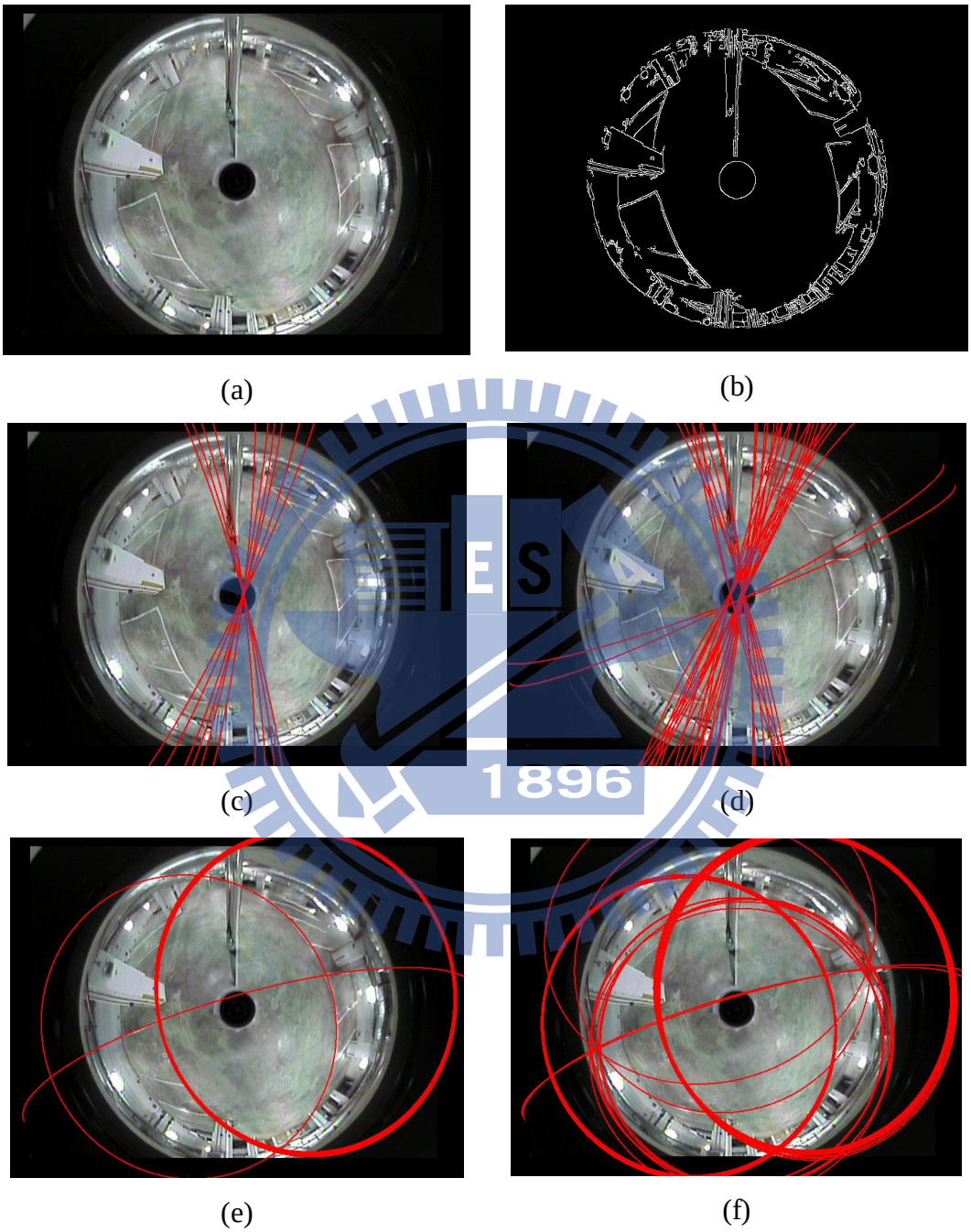
Each parking space is usually marked by some colored boundary lines. In this section, some properties of such boundary lines are first derived. Then, a method to detect such boundary lines is proposed. Finally, a method to mark the regions of parking spaces using boundary lines, and a simple vacancy detection method based on *background subtraction* are proposed, by which whether a car is parked in a parking space can be decided.

Three properties of the boundary lines of parking spaces can be identified: (1) lying on the ground; (2) being either parallel or perpendicular to one another; and (3) being ignorable if too far away from the camera. These properties are discussed in more detail as follows.

As stated previously, the wide-angle cameras used in the proposed parking lot system are affixed on the ceiling to look right downward; so, the optical axis of the camera, namely, the Z-axis of the camera coordinate system as shown in Fig. 7.2, is vertical with respect to the ground specified as the XY-plane. Given a boundary line L_1 on the ground, its directional vector (d_{1x}, d_{1y}, d_{1z}) is parallel to the XY-plane, so we have

$$d_{1z} = 0. \tag{118}$$

Let (A_1, B_1) be the Hough parameters of line L_1 found by the space line detection method, and (n_{1x}, n_{1y}, n_{1z}) be the normal vector of the plane formed by L_1 and the camera origin. Then, from (111) we have



(a) (b)

(c) (d)

(e) (f)

Fig. 7.10 Comparisons of space line detection results yielded by conventional and proposed method. (a) Input image. (b) Edge detection result. (c)(d) 10 and 30 detected curves using conventional method. (e)(f) 10 and 30 detected curves using proposed method.

$$\left(n_{1x}, n_{1y}, n_{1z}\right) = \left(A_1, B_1, \sqrt{1 - A_1^2 - B_1^2}\right). \tag{119}$$

In addition, since (n_{1x}, n_{1y}, n_{1z}) is perpendicular to the directional vector (d_{1x}, d_{1y}, d_{1z}) of L_1, we have

$$\left(n_{1x}, n_{1y}, n_{1z}\right) \cdot \left(d_{1x}, d_{1y}, d_{1z}\right) = 0, \tag{120}$$

which, when combined with (118) and (119), leads to

$$A_1 d_{1x} + B_1 d_{1y} = 0, \tag{121}$$

so that the directional vector (d_{1x}, d_{1y}, d_{1z}) can be expressed in terms of A_1 and B_1 as:

$$\left(d_{1x}, d_{1y}, d_{1z}\right) = \left(B_1, -A_1, 0\right). \tag{122}$$

Let (A_2, B_2) be the parameters of another boundary line L_2. Then, the directional vector (d_{2x}, d_{2y}, d_{2z}) of L_2 can be derived similarly to be

$$\left(d_{2x}, d_{2y}, d_{2z}\right) = \left(B_2, -A_2, 0\right). \tag{123}$$

Now, since the two boundary lines L_1 and L_2 are either parallel or perpendicular to each other, two cases can be identified. For the first case where the two boundary lines are parallel, their directional vectors satisfy

$$\left(d_{1x}, d_{1y}, d_{1z}\right) = \lambda\left(d_{2x}, d_{2y}, d_{2z}\right), \tag{124}$$

which, when combined with (122) and (123), leads to

$$\left(A_1, B_1\right) = \lambda\left(A_2, B_2\right), \tag{125}$$

where λ is a scalar value. For the other case where the two lines are perpendicular, the inner product of their directional vectors satisfy

$$\left(d_{1x}, d_{1y}, d_{1z}\right) \cdot \left(d_{2x}, d_{2y}, d_{2z}\right) = 0, \tag{126}$$

which, with the use of (122) and (123), leads to

$$\left(A_1, B_1\right) \cdot \left(A_2, B_2\right) = 0. \tag{127}$$

Moreover, in order to ignore boundary lines which are too far away, we check the *elevation angle* ψ of each boundary line, which is defined in a way as illustrated in Fig. 7.11 so that if the larger the angle Ψ is, the farther the boundary line is from the camera. According to our experimental experience, if the camera is mounted at 2.5

meters high, the farthest usable boundary line is at a distance about 10.0 meters from the camera. As a result, the maximum value of Ψ is $\tan^{-1}(10.0/2.5) = 75.96°$, which is equivalent to $n_{1z} \le \sin(\Psi) = 0.970$. Using this result with (119), we can derive another constraint on the parameters (A, B) as $n_{1z} = \sqrt{1 - A_1^2 - B_1^2} \le 0.970$, or equivalently, as

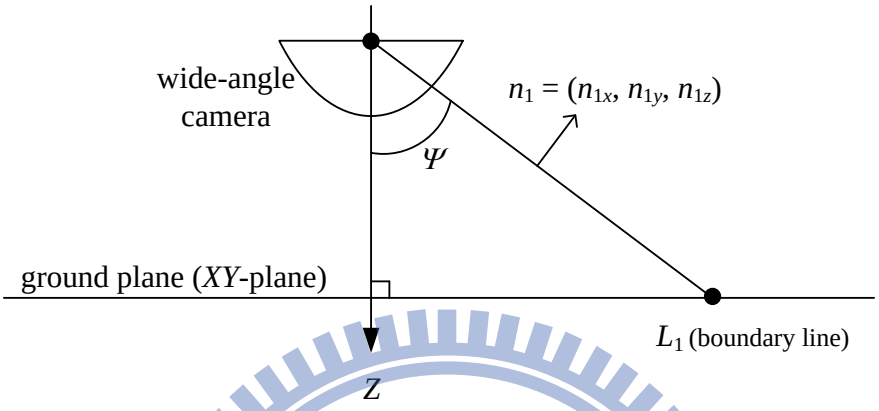


Fig. 7.11 Definition of the elevation angle Ψ of a boundary line L_1.

$$A_1^2 + B_1^2 \ge 0.059 . \tag{128}$$

As can be seen from Figs. 7.8(e) and (f), the results of the line image detection method are too noisy to be used directly. Next, a method is proposed to make use of the geometric properties mentioned previously to generate a better segmentation result.

As defined above, let L_1 and L_2 be two boundary lines with parameter pairs (A_1, B_1) and (A_2, B_2), respectively. If (A_1, B_1) and (A_2, B_2) are treated as two vectors, then if the two lines are parallel to each other, (125) can be derived. Similarly, if the two lines are perpendicular to each other, (127) can be derived. Recalling that the boundary lines are either mutually parallel or perpendicular, we define the *direction θ of the boundary lines* in such a way that the parameters (A, B) of each boundary line are either "parallel" or "perpendicular" to $(\cos\theta, \sin\theta)$, i.e., are equal, up to a scalar, to either $(\cos\theta, \sin\theta)$ or $(\cos(90^o+\theta), \sin(90^o+\theta))$.

To find the correct value of the direction θ, we first generate the Hough space and apply the filter as stated in Sec. III. From this Hough space, about 0.3% of the Hough cells with the largest values and satisfying (128) are chosen and put into a set S. Also, we create 90 bins in the range 0°~90° for different values of θ. For each cell with

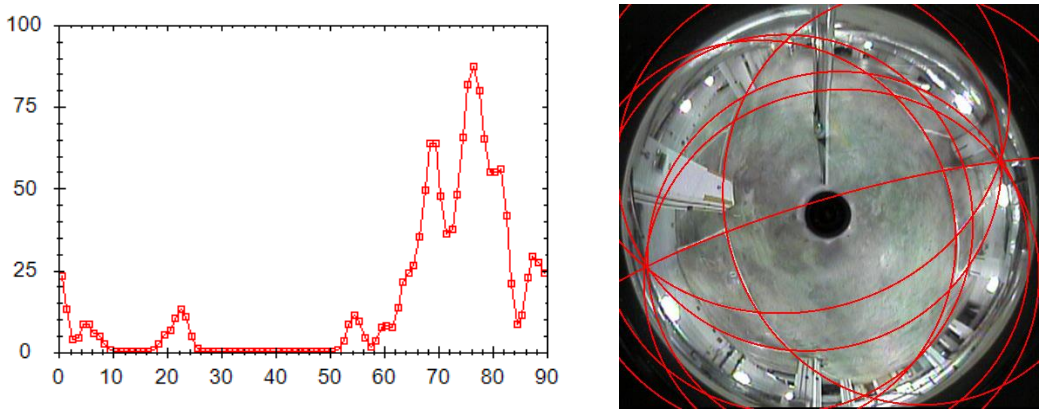
parameters (A_i, B_i) in the set S, we find the *closest* bin b_i with value θ_i such that (A_i, B_i) is either parallel or perpendicular to ($\cos\theta_i$, $\sin\theta_i$), and increment bin b_i by the value of the Hough cell with (A_i, B_i). After applying a Gaussian filter with standard deviation $\sigma = 1$, the bin with the largest accumulation value is chosen to be the direction of the boundary lines θ.

An experimental result of the above process is described in the following. An image captured by a hyperbolic catadioptric camera for use as the input is shown in Fig. 7.8(a). By following the line image detection method proposed in Sec. III, the 2D Hough space, after applying the filter, becomes as that shown in Fig. 7.8(d). The results of the accumulated bin values of θ are shown in Fig. 7.12(a), where the largest bin is with value $\theta = 76.5°$, which is taken finally as the direction of the boundary lines.

After the direction θ of the boundary lines is derived, the boundary lines which do not satisfy the direction θ are removed as follows. For each Hough cell in the set S with parameters (A_i, B_i), if there exists a real number θ_{offset} within the range $-10° \leq \theta_{\text{offset}} \leq 10°$ such that the two vectors (A_i, B_i) and ($\cos(\theta + \theta_{\text{offset}})$, $\sin(\theta + \theta_{\text{offset}})$) are parallel or perpendicular to each other, then the cell is considered to satisfy the direction θ and so kept in the set S; otherwise, it is removed. Afterwards, the connected components S_i in the set S are found according to the criterion that two cells with parameter values (A_1, B_1) and (A_2, B_2) are connected if

$$|A_1 - A_2| \leq 0.05 \text{ and } |B_1 - B_2| \leq 0.05. \quad (129)$$

Finally, the center of each connected component S_i is calculated and their parameters (A_i, B_i) are then used to specify the boundary lines. By this process, the result corresponding to Fig. 7.12(a) is shown in Fig. 7.12(b), in which one can see that all the boundary line of the parking spaces have been detected.



(a) (b)

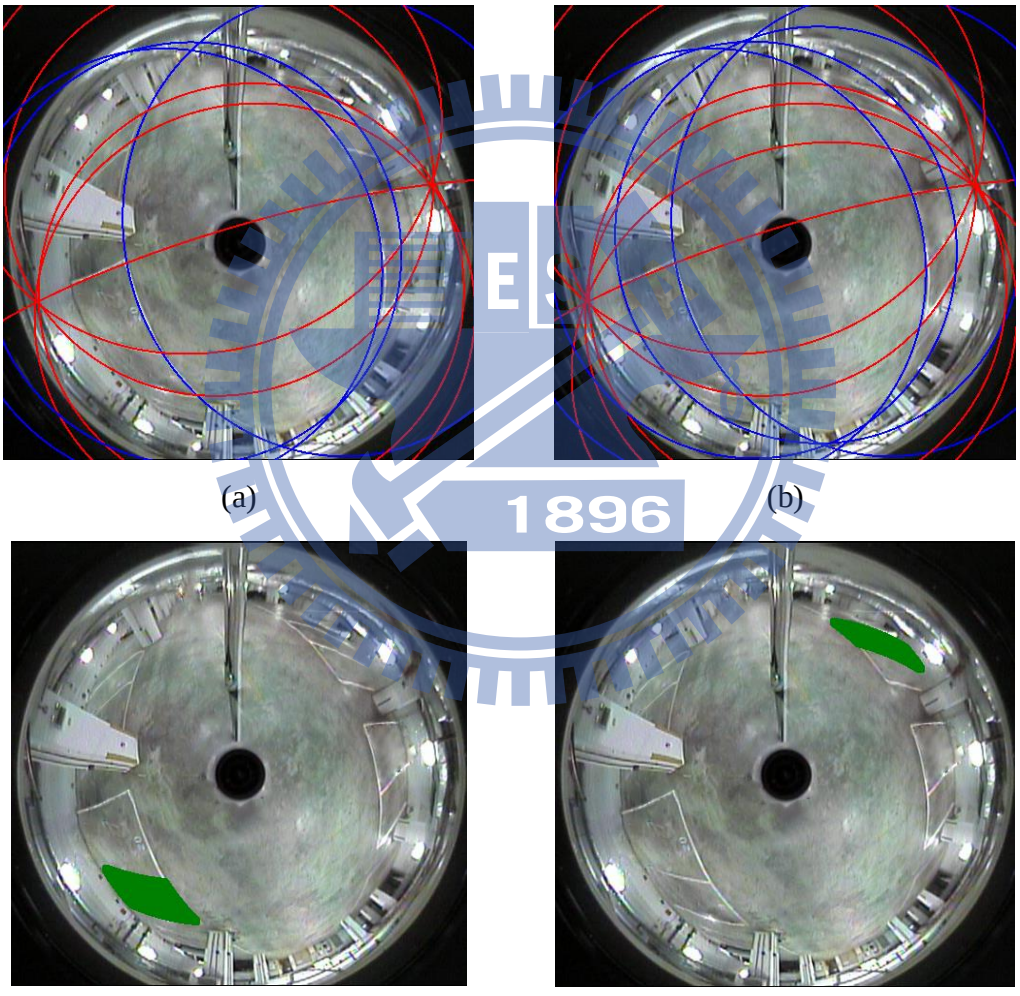
Fig. 7.12 An example of parking space detection results with input image shown in Fig. 7.8(a). (a) Cell accumulation result of the direction θ of boundary lines. (b) Results of boundary lines found by proposed method, showing improvement over line detection results shown in Figs. 7.8(e) and (f).

Two more steps are proposed next to segment out the regions of the parking spaces in the acquired wide-angle image. Then, a vacancy detection scheme based on background subtraction is adopted. Recall that, these three steps are designed to be performed easily by a user with no technical background, so that the vision-based parking lot system can be set up and utilized conveniently as claimed.

The first step is to adjust the boundary lines manually by clicking on the image to add and remove boundary line, one line a time. The boundary lines are first divided into two "mutually" perpendicular groups, as shown in Fig. 7.13(a) and distinguished by colors. The user may select either group, and then simply click on an existing boundary line to remove it, or click on any other position to add a new boundary line going through the mouse position. It is important to note that, with the direction θ found previously, the boundary line can now be uniquely determined with only *one* point marked by the user; without θ, there will be an infinite number of lines going through the point marked by the user. For example, from the detection results shown in Fig. 7.13(a), the result after two lines added and one line removed manually is shown in Fig. 7.13(b).

The second step is to find parking-space regions. In this step, the user may simply click on any position of a parking space in the image, and a region growing algorithm is performed immediately to find the region of the parking space using the boundary lines. Two results are shown in Figs. 7.13(c) and (d).

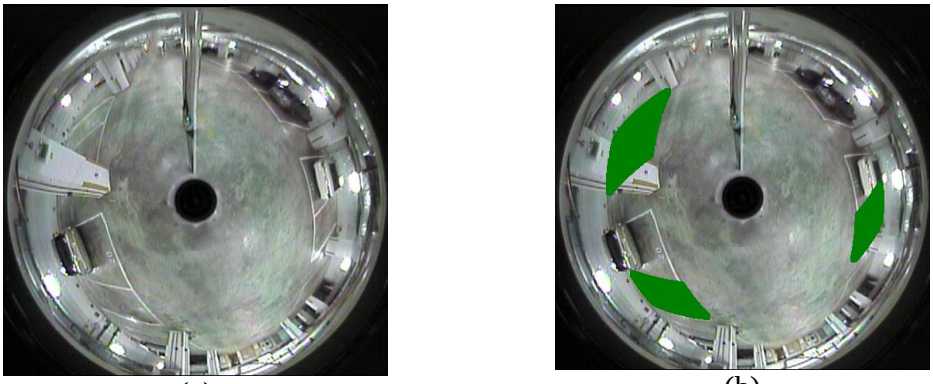
The final step is to find vacant parking spaces. To implement this, the user has to capture beforehand an image of each parking space when it is vacant, and consider the image as the *background* of the parking space. Subsequently, to determine the vacancy of the parking space, each pixel in the acquired wide-angle image is considered as a *foreground pixel* if the absolute difference between the pixel's value and that of the background pixel is larger than a threshold value in at least one of the R, G, and B channels. And then, if the number of foreground pixels is larger than 20% of that of the pixels in the parking-space region, the parking space is considered to be *occupied*; otherwise, *vacant*. An experimental result is shown in Fig. 7.14.



(a) (b)

(c) (d)

Fig. 7.13 Parking space segmentation. (a) Boundary lines detected by proposed method are divided into two groups marked by red and blue, respectively. (b) Boundary lines after manually adjusted. (c)(d) Regions of two green-marked parking spaces found automatically by proposed scheme.



(a) (b)

Fig. 7.14 Parking space vacancy detection. (a) Input image. (b) Vacant parking spaces marked as green regions.

7.5 Experimental Results of Proposed Calibration Method

The proposed camera calibration method described in Sec. II uses the image of only a space line, without knowing its position or direction, to derive the parameters of the proposed general model for fisheye-lens, parabolic catadioptric, and hyperbolic catadioptric cameras. By assuming that the image center has already been calibrated by the use of the circular boundary of the omni-image as done in [48], the effective focal length value f_e is the only remaining one to be calibrated. For this, a series of experiments were conducted using space lines lying on the ground at different positions, with the camera affixed on the ceiling and looking downward. Specifically, as illustrated in Fig. 7.15, one of the parameters of a space line L is its elevation angle ψ, so several simulated space lines with different elevation angles ψ in the range of 0°~90° were generated for uses in the experiments. The parameters f_e of these simulated lines were all set to be $f_e^* = 500$, and their line images generated to be of the size 800×800 identically. Also, the coordinates of the pixels on each line image were disturbed by Gaussian noise with zero mean and varying standard deviations σ. Then, the proposed calibration method was applied to derive the parameter f_e using these simulated line images. This process was repeated 100 times, and the mean and variances of the derived values of f_e are plotted in Figs. 7.16(a), (b), and (c) for different noise levels σ = 1.0, 3.0, and 5.0 pixels, respectively. Recalling that the actual value of the parameter f_e is $f_e^* = 500$, one can see from the figures that the parameter f_e can be robustly calibrated under a large noise level when the elevation angle $\Psi > 15°$. Note also that a space line with an elevation angle Ψ near 0° will appear to be a short and nearly *straight* line segment going through the image center,

so that the parameter f_e cannot be calibrated reliably using it, as expected according to the theory of the proposed calibration method.

The main objective of the proposed methods is the ease for a non-technical user to set up the system. In this section, achievement of this goal is demonstrated — the user only has to select a single space line with its position and direction unknown. Images like those shown in Fig. 7.17 were captured by a hyperbolic catadioptric camera with MapCam MRC530N manufactured by EeRise, Inc. Four experimental results are shown in Figs. 7.17(a)-(d). In each experiment, the user only has to choose arbitrarily an obvious space line in the captured image, and marked *roughly* the pixels on the line as yellow points by mouse dragging. Then, the proposed calibration method was applied to derive the value of the effective focal length f_e using the yellow points. The values of f_e so derived for the cases shown in Figs. 7.17(a)-(d) are 265.34, 258.14, 276.84, and 272.36, respectively, and the best-fitting ellipses to the marked space lines are shown in red. With parameter f_e = 265.34, the boundary lines of the parking spaces were finally found and drawn in Fig. 7.17(e) using the proposed detection method. Also conducted were similar experiments with a fisheye-lens camera manufactured by Hunt Electronic, Inc. with model No. HLC-1NAD, and the results are shown in Fig. 7.18. The results of both experiment series show the feasibility and robustness of the proposed calibration method by the fitness of the drawn red ellipses to the marked yellow points, the closeness of the calibrated values f_e to the real value, and the nearly perfect overlapping of the drawn boundary lines on the real ones.

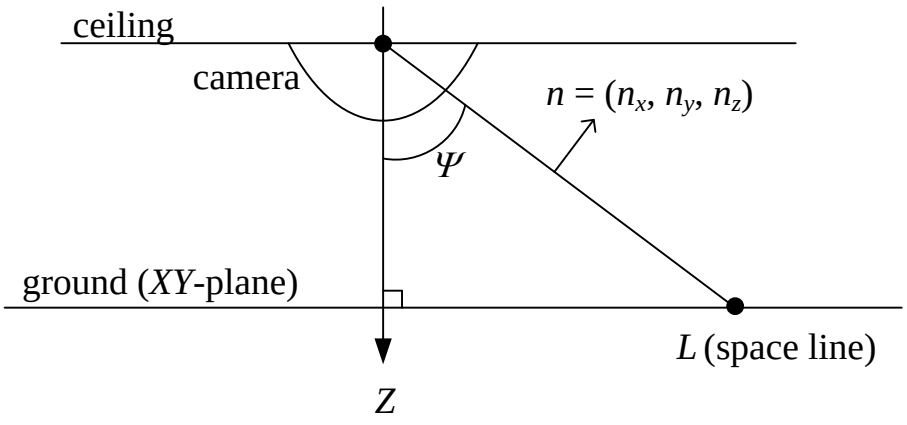
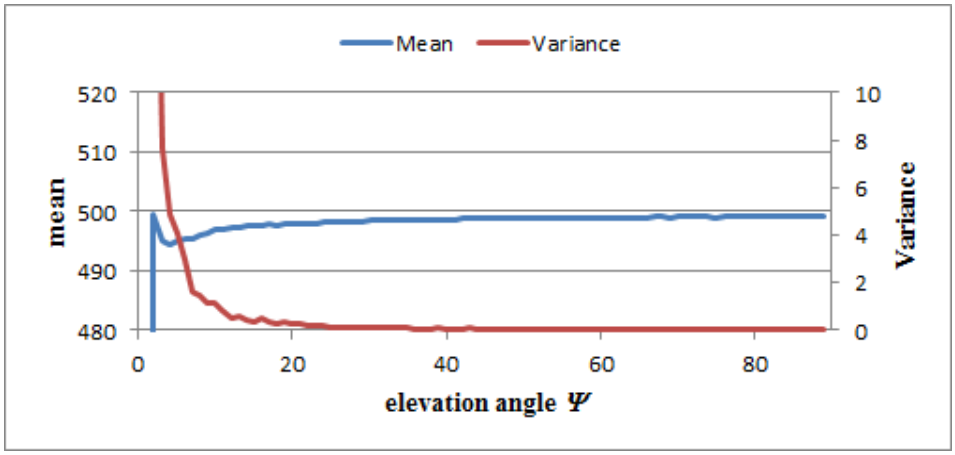
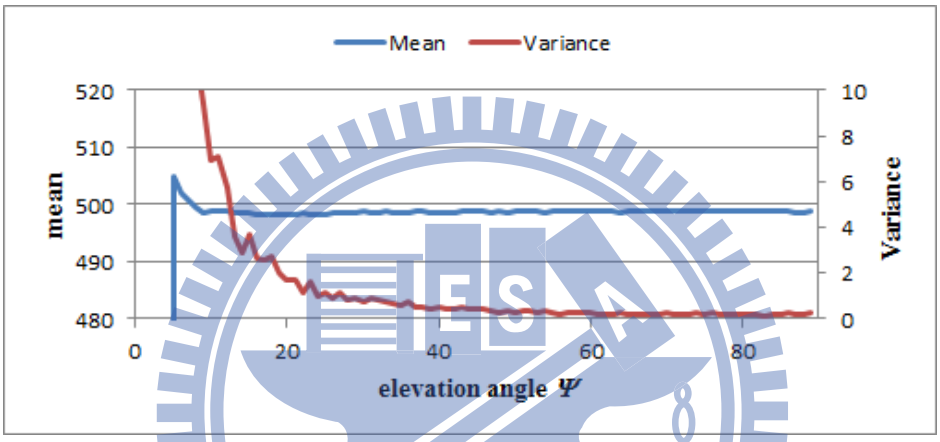


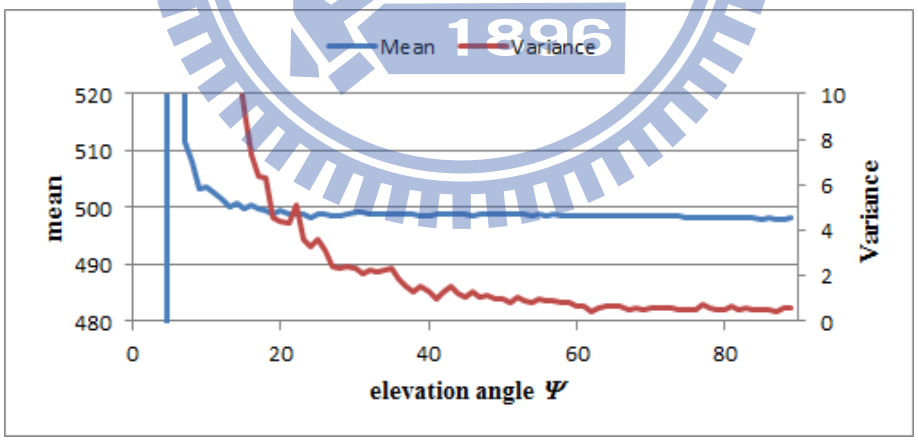
Fig. 7.15 Definition of the elevation angle Ψ of a space line L on the ground.



(a)

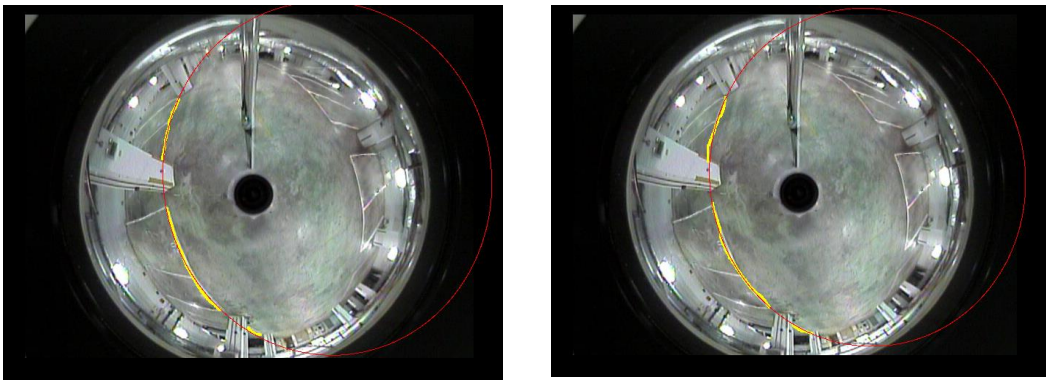


(b)



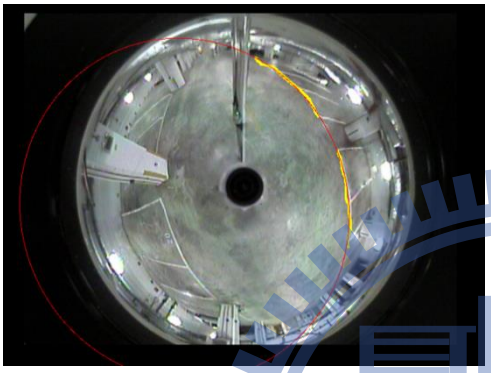
(c)

Fig. 7.16 Computed means and variances of calibrated values f_e for simulated space lines with different elevation angles Ψ where standard deviaion of Gaussian noise is 1.0 pixel for (a), 3.0 pixels for (b), and 5.0 pixels for (c).

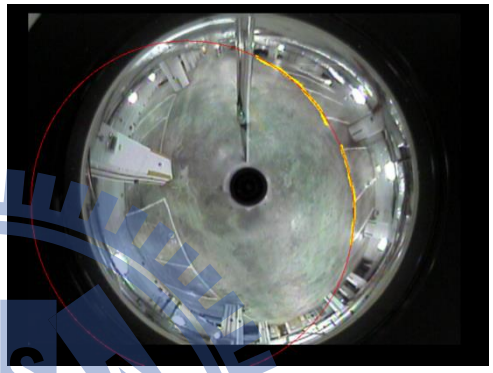


(a)

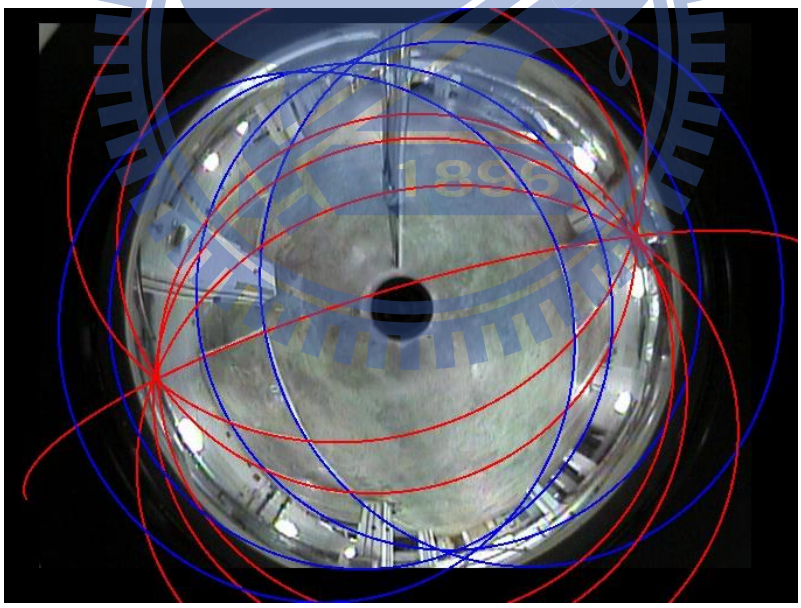
(b)



(c)

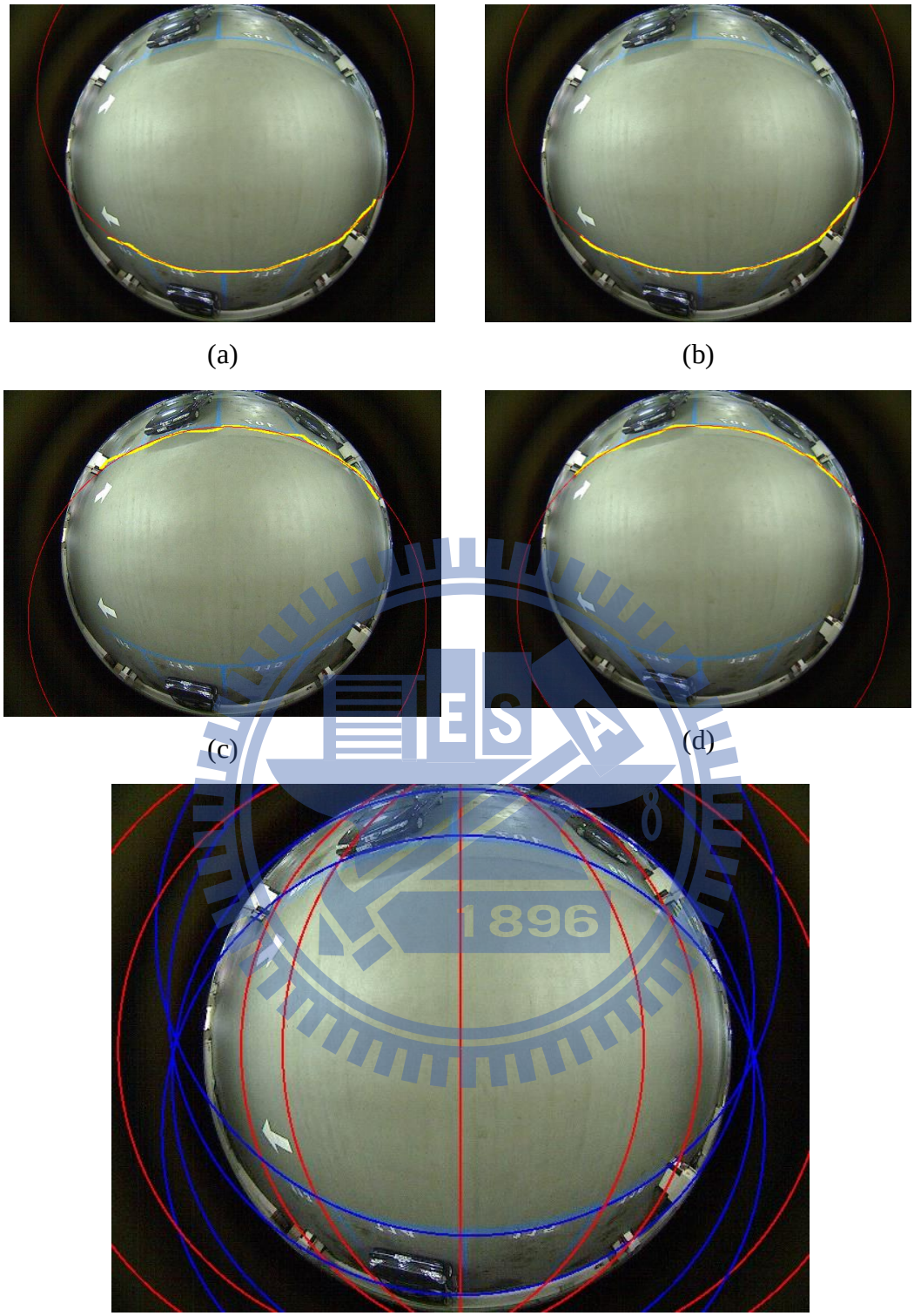


(d)



(e)

Fig. 7.17 Applying proposed calibrate method to derive the parameter f_e of a hyperbolic catadioptric camera. (a)-(d) Four results with calibrated values f_e = 265.34, 258.14, 276.84, and 272.36, respectively. (e) Boundary lines found by the proposed detection method with f_e = 265.34.

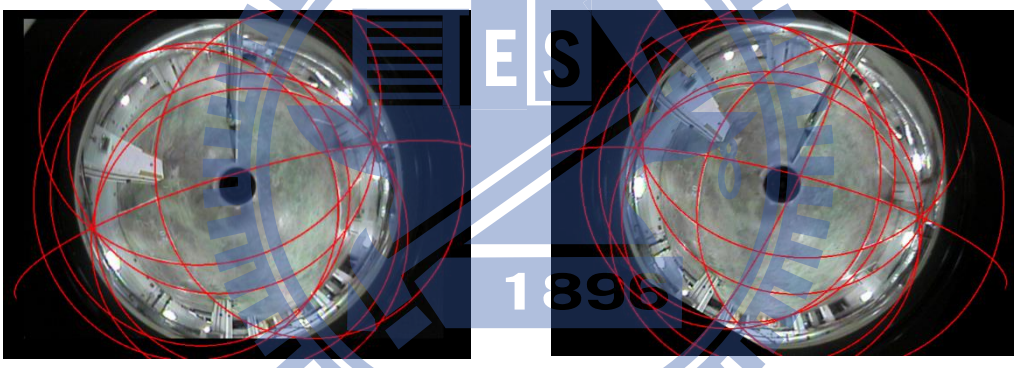


(a) (b) (c) (d) (e)

Fig. 7.18 Use of proposed calibration method to calibrate a fisheye-lens camera. (a)-(d) Four results with calibrated values f_e = 323.74, 331.12, 339.74, and 328.57, respectively. (e) The boundary lines found by proposed detection method with f_e = 331.12.

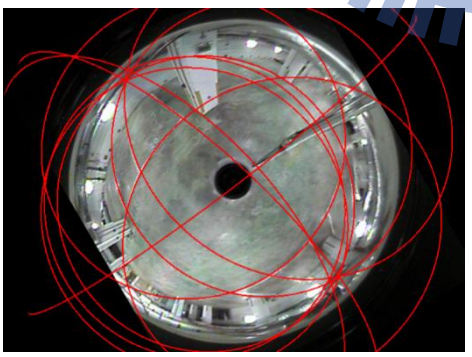
7.6 Experimental Results of Parking Space Segmentation

In the proposed boundary line detection method, the direction of the boundary lines, denoted as θ, is computed first and the boundary lines are derived accordingly. Some experiments were conducted to test the capability of this scheme of finding θ as described next. First, an image of an indoor parking lot was captured by a hyperbolic catadioptric camera, as shown in Fig. 7.19(a). Then, this image was rotated through the angles 30°, 60°, and 90° as shown in Figs. 7.19(b), (c) and (d), respectively. The boundary line directions θ in these four images were firstly derived. Then, the boundary lines were detected automatically and superimposed on the original images as shown in Figs. 7.19(a)-(d), respectively. From the figures, one can see that the boundary lines fit well to the real ones appearing in the captured images, meaning that the directions of the boundary lines were found correctly by the proposed method.

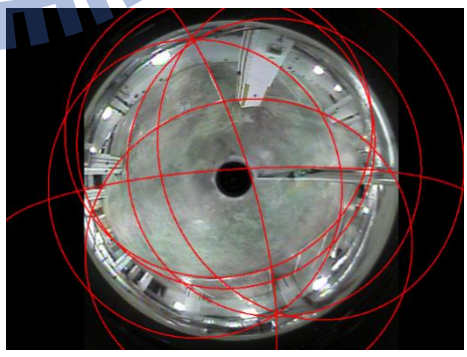


(a)

(b)



(c)



(d)

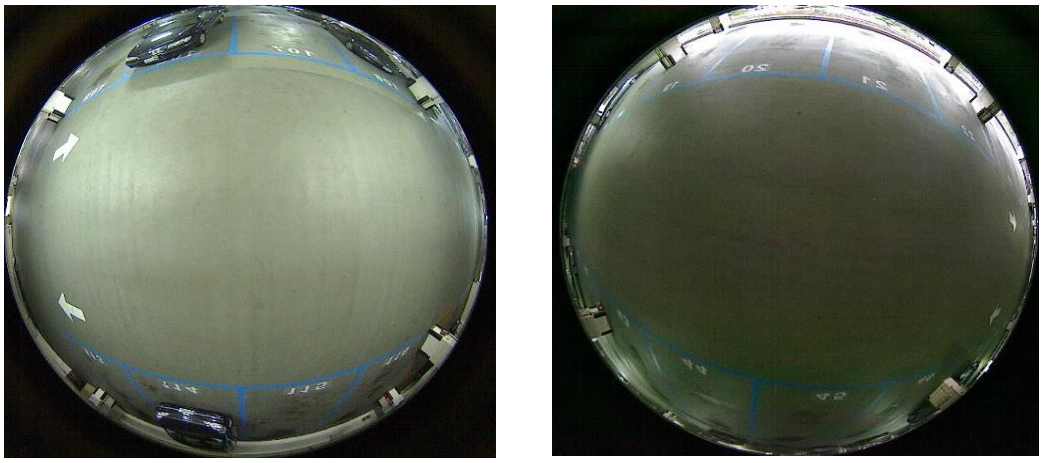
Fig. 7.19 Automatic detection results of boundary lines. Rotation angles of these images are (a) 0°, (b) 30°, (c) 60°, and (d) 90°, respectively. The detected boundary lines are drawn and superimposed on the images.

To demonstrate that the direction θ of the boundary lines can be robustly found even under bad conditions, two more experiments were conducted with input images shown in Figs. 7.20(a) and (b), in which noise was present, cars were parked in the spaces, and the lighting situations were poor. The results of the first experiment are shown in Figs. 7.20(c), (e), and (g), and those of the second are shown in Figs. 7.20(d), (f), and (h). Specifically, the edge detection results are shown in Figs. 7.20(c) and (d), respectively, in which one can see that the boundary lines were poorly detected; Also, some of the boundary lines are missing, as shown in Figs. 7.20(e) and (f). However, the directions of the detected and drawn boundary lines are correct, meaning that the direction θ can be derived precisely even under bad conditions. With this correct value of θ, the boundary lines can be adjusted easily by the proposed method as demonstrated by the results shown in Figs. 7.20(g) and (h).

7.7 An Example of Setting up an Indoor Parking Lot System

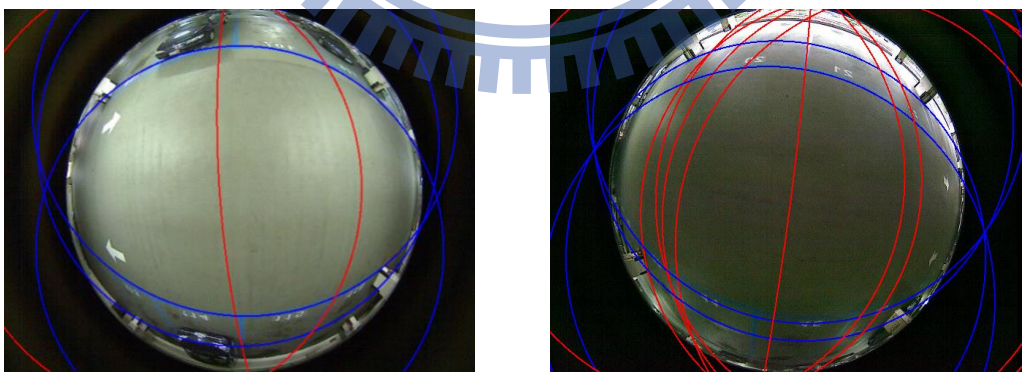
A real indoor vision-based parking lot system was established and its use for this study is introduced in this section. A fisheye-lens camera manufactured by Hunt Electronic with model No. HLC-1NAD was affixed on the ceiling of the parking lot. An image as shown in Fig. 7.21(a) was acquired at midnight firstly and used both for calibration and for detecting the boundary lines of the parking spaces. Specifically, a space line in the image was chosen arbitrarily according to a rule of thumb that the line should be *long enough and not straight*. The line pixels were traced and marked in yellow as shown in Fig. 7.21(b), with calibrated parameter f_e = 331.1.

Afterwards, the direction θ of the boundary lines was derived, and the boundary lines detected automatically, with the results drawn in Fig. 7.21(c). Subsequently, boundary lines were added or removed by *single* clicks on the image as described previously. A result of such adjustments is shown in Fig. 7.21(d). At last, the parking spaces were segmented out by clicking on any point within the region of each space and applying a region growing algorithm. The regions of two parking spaces so found and drawn in green are shown in Figs. 7.21(e) and (f), respectively. After using the captured image shown in Fig. 7.21(a) as background, the system setup process was completed, and parking space vacancy detection was started. In one of the experiments we conducted, this system was maintained to run for 24 hours to detect vacant parking spaces every minute. The resulting detection accuracy was 99.67%, which is good enough for real applications.



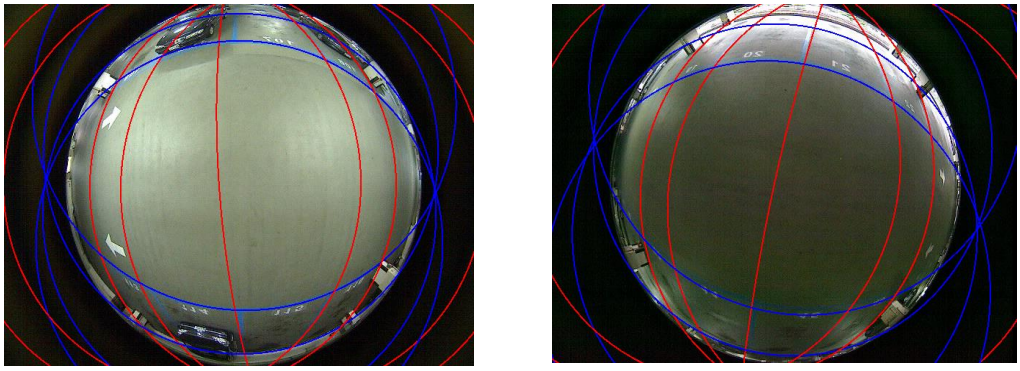
(a) (b)

(c) (d)



(e) (f)

Fig. 7.20 Results of two experiments under bad conditions with those of the first shown in (a)(c)(e)(g), and those of the second shown in (b)(d)(f)(h). (a)(b) Input images. (c)(d) Edge detection results. (e)(f) Automatic detection results of boundary lines. (g)(h) Adjusted boundary lines. *(continued on next page)*



(g) (h)

Fig. 7.20 *(continued from previous page)*

7.8 Conclusions

A convenient indoor vision-based parking lot system has been proposed in this study, which is easy to set up by a typical user with no technical background and can detect vacant parking spaces automatically. The system uses wide-angle cameras, like fisheye-lens or catadioptric ones and analyzes parking-space boundary lines based on a new camera model proposed in this study. This model approximates optimally one of the parameters used in a conventional model (the distance l from the effective viewpoint O to the pinhole point O_c) while reserving some important characteristics of line images, including the shape of the curve and the locations of the vanishing points. A new line-based calibration method has also been proposed to calibrate the camera model using only one space line without knowing its location and direction, so that the calibration process can be done easily by a user without any technical background. A new Hough transform has been proposed as well to detect space lines, in which a skillful cell accumulation scheme is used to generate equal-width curves, yielding more robust and accurate detection result of parking-space boundary lines. A convenient adjustment method has also been developed such that an user can add or remove boundary lines by simple clicks on input images.

Currently, the background images should be learned for various lighting conditions in order to conduct parking-space vacancy detection based on background subtraction. More intelligent methods may be developed to remove this weakness in the future study.

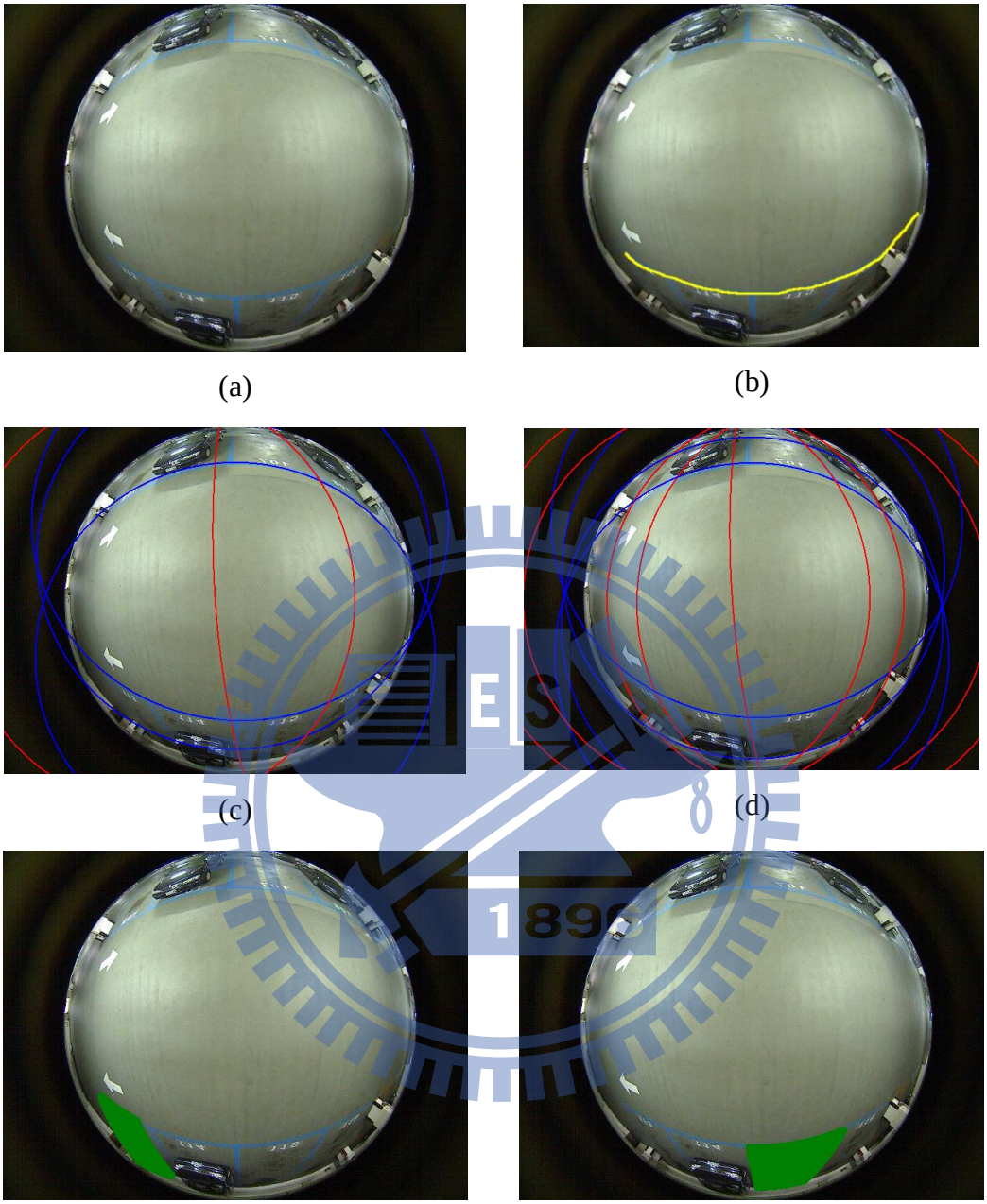


Fig. 7.21 Steps of setting up proposed system. (a) An image captured at midnight. (b) Manually-marked pixels (in yellow) of a space line for calibration task. (c) Results of automatic detection of boundary lines. (d) Result of boundary line adjustment. (e) and (f) Regions of two found parking spaces (marked in green).

Chapter 8 Conclusions and Suggestions for Future Works

8.1 Conclusions

In this dissertation, several methods for applications using omni-directional cameras have been proposed. A new camera model for omni-cameras has been proposed, which can be used to model various kinds of omni-cameras, and a convenient calibration method for the new model has also been proposed with the use of only one straight line without its location and direction unknown in advance. Also, an improved space line detection method has been proposed to detect straight lines from images captured by omni-cameras without unwarping them. From the viewpoint of consumers, a new binocular omni-vision system has been proposed as well along with an adaptation algorithm which can automatically adapt the system parameters to *any* system setups, so that the proposed vision system provides a convenient setting up process without any restriction on the locations or orientations of the two cameras. Furthermore, to design an optimal vision system, the issue of optimally designing the optimal configuration of a stereo vision system with two catadioptric omni-cameras has been investigated, and the proposed optimization methods yield the optimal poses of the cameras and the optimal mirror-shape parameters of the mirrors used in the omni-cameras. Finally, a convenient indoor vision-based parking lot system has been proposed in this study, which can be easily set up by a typical user with no technical background and can detect vacant parking spaces automatically. In the following, the conclusion of each method and suggestions for future researches are given as follows.

(1) A new camera model for omni-cameras has been proposed, which can be used to model various kinds of omni-cameras, including single viewpoint catadioptric cameras, fisheye-lens cameras, etc. One of the main features of the proposed model is that it can be easily calibrated from one straight line without knowing the location and direction of the line. More specifically, in the proposed modified model, we have investigated an important invariant property in the projections of straight lines via omni-directional cameras, and have used this invariant property to eliminate one of the parameters, i.e., the parameter l, in the original unifying

model proposed by Geyer and Daniilidis [44]. In more detail, a series of experiments have been designed and conducted to find the optimal approximate value of parameter *l*, leaving only one parameter in the proposed modified model. Since there is only one parameter left in the camera model, the model can be easily calibrated, and it has been showed that it can be calibrated more robustly using only straight lines. A new calibration method has also been proposed using only one straight line to calibrate an omni-camera, which is a great advantage since it facilitates a non-technical user to conduct the calibration process without difficulty because it requires no extra object or measurement in the process.

(2) An improved space line detection method has been proposed to detect straight lines from the images captured by omni-cameras without unwarping them. Three different conventional approaches has been first identified and analyzed, and it has been found out that these conventional methods have problems when detecting space lines from omni-images, yielding imprecise and unrobust detection results. From some more thorough analyses, it turns out that the main reason of the imprecision and unrobustness is because the conventional methods cannot generate equal-width curves when detecting space lines. To overcome this problem, a new method to generate equal-width curves has been proposed in this study with the use of total differentials, and this method has also been combined with the Hough Transform to yield the proposed space line detection method. From the experimental results, the proposed method yields much more precise and robust detection results when detecting space lines from omni-images, and it only requires a little more computation time when comparing with the conventional methods. Also, the proposed detection method along with the equal-width curve extraction techniques can be easily generalized to detect more complex analytical shapes.

(3) A new binocular omni-vision system has been proposed along with an adaptation algorithm which can automatically adapt the system parameters to *any* system setups. The proposed vision system is aimed to provide a convenient setting up process without any restriction on the locations or orientations of the two cameras, so that it is suitable for the use in the area of consumer electronics. Firstly, the proposed vision system is composed of two omni-cameras facing the user's activity area. Each omni-camera is affixed firmly to the top of a rod, forming an

omni-camera stand, with the camera's optical axis adjusted to be horizontal (i.e., parallel to the ground). To deploy the vision system, the user can just bring the two camera stands home, and place them freely at the desired locations and with arbitrary orientations. After the two cameras are placed, the proposed adaptation process can be used to automatically derive the orientations of the two cameras with the use of the straight lines in the environment. During this step, the process also takes the advantage of the property that, the straight lines are mostly parallel or perpendicular to each other, to improve the correctness and robustness, and make the process require no line correspondence algorithm. After deriving the orientations, the system will ask the user to stand at the middle region in front of the two cameras, and derive the distance between the two cameras (i.e., the baseline) with the use of the user's height. Accordingly, a coordinate system can be defined with no ambiguity, so the 3D data can be computed correctly. The proposed vision systems has two main advantages over the traditional ones. First, the proposed vision system can be set up freely, meaning that the user can place the two cameras at the locations he or she wants, and orient them freely. Second, the proposed system is with a very large viewing field by the use of omni-directional cameras, so the users can move much more freely within the environment.

(4) The issue of designing the optimal configuration of a stereo vision system with two catadioptric omni-cameras to compute 3D data with minimum errors has been investigated. The solution includes the poses of the cameras and the mirror-shape parameters of the hyperboloidal mirrors used in the omni-cameras. An analytic formula is derived to model the 3D measurement error, which takes into consideration the variations of pixel-quantization precisions and angular resolutions in images by conducting error propagation analysis in the data computation process. Two fast and elegant optimization algorithms have been designed accordingly for "regular" environments with rectangular cuboid-shaped 3D measurement and camera placement areas. One of them, based on a bisection scheme, is optimal but relatively slower, which may be used for off-line applications. And the other, using analytic formulas to calculate approximate solutions, is faster for realtime applications with the computed precision being sub-optimal but close to that of the former. An algorithm for dealing with general

environments with irregular-shaped 3D measurement and camera placement areas has also been developed for general uses. Experimental results show the feasibility of the proposed method. In real applications, a manufacturer may produce omni-cameras according to the derived optimal mirror shape. Then, a consumer may bring them back and deploy them in the optimal or nearly-optimal pose using the proposed algorithms. As a result, a stereo vision system which yield precise 3D measurement results can be set up.

(5) A convenient indoor vision-based parking lot system has been proposed in this study, which is easy to set up by a typical user with no technical background and can detect vacant parking spaces automatically. The system uses omni-cameras mounted on the ceiling, like fisheye-lens or catadioptric ones, and analyzes parking-space boundary lines based on a new camera model. The calibration method uses only one space line without knowing its location and direction, so that the calibration process can be done easily by a user without any technical background. A new Hough transform has been proposed as well to detect space lines, in which a skillful cell accumulation scheme is used to generate equal-width curves, yielding more robust and accurate detection result of parking-space boundary lines. A convenient adjustment method has also been developed such that a user can add or remove boundary lines by simple clicks on input images. Currently, the background images should be learned for various lighting conditions in order to conduct parking-space vacancy detection based on background subtraction.

8.2 Suggestions for Future Works

The following topics may be investigated in the future to further improve the proposed omni-vision techniques:

(1) Omni-camera calibration using multiple lines —

It is desired to utilize more than one space lines to calibrate omni-cameras. The geometry relation between these multiple lines should be utilized in the calibration process, such as the properties of parallel, perpendicular, distance, length, directions, etc. In this way, omni-cameras can be easily calibrated and

will be suitable for point localization and/or curve detections of general shapes.

(2) Thick curve detections in omni-images —

Future studies may be directed to improve the proposed method by focusing on extracting equal-width curves regarding to the 3D world, rather than the 2D image space. Also, it is also desired to combine the proposed equal-width extraction techniques with advanced binary thresholding techniques. In this way, the region of a thick curve can be segmented out, so hopefully the curve can be extracted accurately and robustly by the proposed method.

(3) Automatic adaptation to any system setups —

Future works may focus on eliminating the coplanrity constraint of the optical axes. Also, it is also desired to develop automatic adaptation techniques for multiple omni-camera vision systems. In this way, the user can place the omni-cameras freely in the environment, and then the vision system can automatically adapt the system setup using features in the environment.

(4) Derivation of optimal system configuration —

It is possible to generalize the proposed optimization method to a stereo vision system with more than two omni-cameras. Also, other optimization methods (e.g., evolutionary algorithms) could be developed and compared to existing methods to come up with a good optimization method.

(5) Parking lot management system using omni-cameras —

More intelligent methods may be developed to remove this weakness in the future study. Also, future studies may be directed to develop a more convenient system by developing a method to simultaneously calibrate omni-cameras and detect parking spaces. In this way, the system can further ease the setting up phase by elimination the calibration process.

References

[1] Z. Y. Zhou, A. D. Cheok, Y. Qiu, and X. Yang, "The Role of 3-D sound in human reaction and performance in augmented reality environments," *IEEE Trans. on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 37, no. 2, pp. 262–272, 2007.

[2] B. J. Tippetts, D. J. Lee, J. K. Archibald, and K. D. Lillywhite, "Dense disparity real-time stereo vision algorithm for resource-limited systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1547–1555, 2011.

[3] Y. Sun, X. Chen, M. Rosato, and L. Yin, "Tracking vertex flow and model adaptation for three-dimensional spatiotemporal face analysis," *IEEE Trans.on Systems, Man and Cubernetics–Part A: Systems and Humans*, vol. 40, no. 3, pp. 461–474, 2010.

[4] K. Li, Q. Dai, W. Xu, J. Yang, and J. Jiang, "Three-dimensional motion esimtation via matrix completion," *IEEE Trans. on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 42, no. 2, pp. 539–551, 2012.

[5] Wikipedia contributors, "Kinect," *Wikipedia, The Free Encyclopedi*a, http://en.wikipedia.org/wiki/Kinect.

[6] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no.2, pp. 251–263, 1997.

[7] Y. Yagi, Y. Nishizawa, and M. Yachida, "Map-based navigation for a mobile robot with omni-directional image sensor copis," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, pp. 634–648, 1995.

[8] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omni-directional camera," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 890–898, 2000.

[9] E. Menegatti, T. Maeda and H. Ishiguro, "Image-based memory for robot navigation using properties of the omni-directional images," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.

[10] H. Koyasu, J. Miura, and Y. Shirai, "Recognizing moving obstacles for robot navigation using real-time omni-directional stereo vision," *Journal of Robotics and Mechatronnics*, vol. 14, no. 2, pp. 147–156, June 2002.

[11] C. Cauchois, E. Brassart, B. Marhic, and C. Drocourt, "An absolute localization method using a synthetic panoramic image base," *Proceedings of IEEE Workshop on Omnidirectional Vision, Copenhagen, Denmark*, pp. 128–135, 2002.

[12] Y. Ogawa, J. H. Lee, S. Mori, A. Takagi, C. Kasuga and H. Hashimoto, "The positioning system using the digital mark pattern-the method of measurement of a horizontal distance," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp.731–741, 1999.

[13] S. J. Ahn, W. Rauh, and M. Recknagel, "Circular coded landmark for optical 3D-measurement and robot vision," *Proceedings of International Conference on Intelligent Robots and Systems*, pp.1128–1133, 1999.

[14] S. Kim and S.Y. Oh, "SLAM in indoor environments using omni-directional vertical and horizontal line features," *Journal of Intelligent and Robotic Systems*, vol. 51, no. 1, pp. 31–43, 2008.

[15] J. Kannala and S. Brandt, "A generic camera calibration method for fish-eye lenses," *Proceedings of the 17th International Conference on Pattern Recognition*; Cambridge, U.K, vol. 1, pp. 10–13, 2004.

[16] S. Shah and J. K. Aggarwal, "Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and. accuracy estimation," *Pattern Recognition*, vol. 29, no. 11, pp. 1775–1788, 1996.

[17] Y. C. Liu, K. Y. Lin, and Y. S. Chen, "Bird's-eye view vision system for vehicle surrounding monitoring," *Proceedings of Conference on Robot Vision*, Berlin, Germany, pp. 207–218, 2008.

[18] S. W. Jeng. "A study on camera calibration and image transformation techniques and their applications," Ph. D. Dissertation, Institute of Information Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2007.

[19] L. Tian, L. C. Wu, Y. Wang, G. S. Yang, "Binocular vision system design and its active object tracking," *Proc. IEEE International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, pp. 278–281, 2011.

[20] Y. Xie, Y. N. Wang, B. T. Guo, H. H. Wang, "Study on human-computer interaction system based on binocular vision technology," *Proc. IEEE International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, pp. 1541–1546, 2012.

[21] H. Koyasu, J. Miura and Y. Shirai, "Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments," *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 31–36, Maui, Hawaii, U. S. A., 2001.

[22] S. Laakso and M. Laakso, "Design of a body-driven multiplayer game system," *Computers in Entertainment (CIE)*, vol. 4, no. 4, 2006.

[23] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human–computer interface using symmetry between eyes to detect gaze direction," *IEEE Trans. on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 38, no. 6, pp. 1248–1261, Nov. 2008.

[24] X. Zabulis, T. Sarmis, D. Grammenos and A. A. Argyros, "A multicamera vision system supporting the development of wide-area exertainment applications," *IAPR Conf.on Machine Vision Applications (MVA 2009)*, Yokohama, Japan, pp. 269–272, 2009.

[25] J. Starck, A. Maki, S. Nobuhara, A. Hilton, and T. Matsuyama, "The multiple-camera 3-D production studio," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 6, 2009.

[26] S. Sefvic and S. Ribaric, "Determining the absolute orientation in a corridor using projective geometry and active vision," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 3, pp. 696–710, June 2001.

[27] R. Carelli, R. Kelly, O. H. Nasisi, C. Soria, and V. Mut, "Control based on perspective lines of a non-holonomic mobile robot with camera-on-board," *Int'l Journal of Control*, vol. 79, no. 4, pp. 362–371, 2006.

[28] X. Ying and H. Zha, "Simultaneously calibrating catadioptric camera and detecting line features using Hough transform," *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 412–417, Aug. 2005.

[29] X. Ying, "Catadioptric Camera Calibration Using Geometric Invariants," *Proc. IEEE Int'l Conf. on Computer Vision*, vol. 2, pp. 1351–1358, Oct. 2003.

[30] F. Duan, F. Wu, M. Zhou, X. Deng, and Y. Tian, "Calibrating effective focal length for central catadioptric cameras using one space line," *Pattern Recognition Letters*, vol. 33, pp. 646-653, 2012.

[31] R. G. von Gioi, J. Jakubowicz, J.-M. Morel and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, April 2010.

[32] C. J. Wu and W. H. Tsai, "An omni-vision based localization method for automatic helicopter landing assistance on standard helipads," *Proc. Int'l Conf. on Computer and Automation Engineering*, Singapore, pp. 327–332, 2010.

[33] S. J. Maybank, S. Ieng, and R. Benosman, "A Fisher-Rao metric for paracatadioptric images of lines," *Int'l Journal of Computer Vision*, vol. 99, no. 2, pp. 147–165, 2012.

[34] K. Yamazawa, Y. Yagi and M. Yachida, "3D line segment reconstruction by using hyperomni vision and omnidirectional Hough transforming," *Proc. Int'l Conf. on Pattern Recognition*, vol. 3, IEEE Computer Society, Washington, DC, USA, pp.3487–3490, 2000.

[35] S. T. Barnard, "Interpreting perspective images," *Artifical Intelligence*, vol. 21, pp. 435–462, 1983.

[36] B. Li, K. Peng, X. Ying, and H. Zha, "Vanishing point detection using cascaded 1D Hough Transform from single images," *Pattern Recognition Letters*, vol. 33, pp. 1-8, 2012.

[37] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann, "Active Visual Object Reconstruction using D-, E-, and T-Optimal Next Best Views," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-7, 2007.

[38] H. Zhang, "Two-dimensional optimal sensor placement," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 781–792, 1995.

[39] B. Alsadik, M. Gerke, and G. Vosselman, "Automated Camera Network Design for 3D Modeling of Cultural Heritage Objects," *Journal of Cultural Heritage*, 2013.

[40] C. Hoppe, A. Wendel, S. Zollmann, and S. Kluckner, "Photogrammetric Camera Network Design for Micro Aerial Vehicles," *Proc. 17th Computer Vision Winter Workshop*, Feb. 2012.

[41] G. Olague, and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern Recognition*, vol. 35, no. 4, pp. 927–944, 2002.

[42] A. H. Rivera, F. L. Shih, and M. Marefat, "Stereo camera pose determination with error reduction and tolerance satisfaction for dimensional measurements," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 423–428, April 2005.

[43] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[44] C. Geyer and K. Daniilidis, "A Unifying Theory for Central Panoramic Systems and Practical Implications," *Proc. Sixth European Conf. Computer Vision*, pp. 445-462, 2000.

[45] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *Int. J. of Computer Vision*, vol. 45, no. 3, pp. 223–243, 2001.

[46] X. Ying and Z. Hu, "Catadioptric Camera Calibration Using Geometric Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1260-1271, 2004.

[47] X. Ying and Z. Hu, "Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model?" *Proc. European Conference on Computer Vision*, pp. 442-455, 2004.

[48] X. M. Deng, F. C. Wu, and Y. H. Wu, "An Easy Calibration Method for Central Catadioptric Cameras," *Acta Automatica Sinica*, vol. 33, no. 8, pp. 801-808, 2007.

[49] Y. Bastanlar, L. Puig, P. Sturm, J. J. Guerrero, and J. Barreto, "DLT-Like Calibration of Central Catadioptric Cameras," *Proc. 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, Oct. 2008.

[50] S. Gasparini, P. Sturm, and J. P. Barreto, "Plane-Based Calibration of Central Catadioptric Cameras," *Proc. IEEE 12th International Conference on Computer Vision*, pp. 1195-1202, 2009.

[51] D. Loannou, W. Huda, and A. F. Laine, "Circle recognition through a 2D Hough transform and radius histogramming," *Image and Vision Computing*, vol. 17, no. 1, pp. 15–26, 1999.

[52] Y. C. Cheng and S. C. Lee, "A New method for quadratic curve detection using K-RANSAC with acceleration technique," *Pattern Recognition*, vol. 28, no. 5, pp. 663–682, 1995.

[53] H. Ukida, N. Yamato, Y. Tanimoto, T. Sano and H. Yamamoto, "Omni-directional 3D measurement by hyperbolic mirror cameras and pattern projection," *Proc. 2008 IEEE Conf. on Instrumentation and Measurement Technology*, Victoria, BC, Canada, pp. 365–370, 2008.

[54] R. I. Hartley and P. Sturm, "Triangulation," *Proc. ARPA Image Understanding Workshop*, pp. 957–966, 1994.

[55] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs, and mathematical Tables*. pp. 72.

[56] S. Baker and S. Nayar, "A theory of single-viewpoint catadioptric image formation," *Int. J. of Computer Vision*, vol. 35, no. 2, pp. 175–196, 1999.

[57] D. Pedoe, *Circles: A Mathematical View (Spectrum)*, 2nd ed. The Mathematical Association of America, 1997.

[58] R. S. Irving, *Integers, Polynomials, and Rings.* New York: Springer, 2004.

[59] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications.* New York: Springer, 1997.

[60] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1996.

[61] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, 1983.

[62] J. P. Barreto and H. Araujo, "Geometric Properties of Central Catadioptric Line Images," *Proc. Seventh European Conference on Computer Vision*, pp. 237-251, 2002.

[63] T. Apostol, *Calculus, Vol 1: One-Variable Calculus with an Introduction to Linear Algebra*, Wiley, 2nd edition, June, 1967.

List of Publications

Journal Papers and Book Chapters

(1) **S. E. Shih** and W. H. Tsai, "A two-omni-camera stereo vision system with an automatic adaptation capability to any system setup for 3D vision applications," *IEEE Transactions on Circuits and Systems for Video Technology*, accepted and to appear.

(2) **S. E. Shih** and W. H. Tsai, "Optimal design and placement of omni-cameras in binocular vision systems for accurate 3D data measurement," *IEEE Transactions on Circuits and Systems for Video Technology*, accepted and to appear.

(3) **S. E. Shih** and W. H. Tsai, "A new two-omni-camera system with a console table for versatile 3D vision applications and its automatic adaptation to imprecise camera setups," *Advances in Multimedia Modeling (MMM 2011) — Lecture Notes in Computer Science (LNCS)*, vol. 6523, K. T. Lee, W. H. Tsai, H. Y. M. Liao, T. Chen, J. W. Hsieh and T. T. Tseng. (eds.), Springer, Berlin/Heidelberg, Germany, pp. 193–205, 2011.

(4) **S. E. Shih** and W. H. Tsai, "A convenient vision-based system for automatic detection of parking spaces in indoor parking lots using wide-angle cameras," *IEEE Transactions on Vehicular Technology*, submitted

(5) **S. E. Shih** and W. H. Tsai, "A dynamic thresholding technique for Hough Transform with applications to equal-width curve detection," *IEEE Transactions on Image Processing*, submitted.

Conference Papers

(1) **S. E. Shih** and W. H. Tsai, "A new two-omni-camera system with a console table for versatile 3D vision applications and its automatic adaptation to imprecise camera setups," *Proceedings of 17th International Multimedia Modeling Conference* (*MMM 2011*), pp. 193–205, Taipei, Taiwan, 2011.

(2) **S. E. Shih** and W. H. Tsai, "Hough transform with dynamic thresholding for robust and real-time detection of complex curves in images," *Proceedings of 38th IEEE International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP 2013*), Vancouver, Canada, 2013.

Vitae

Shen-En Shih was born in Taipei, Taiwan, R.O.C. on December 31, 1986. He received the B.S. degree in computer science from National Chiao Tung University, Taiwan, in 2009, and works toward his Ph.D. degree at the College of Computer Science, National Chiao Tung University. He has been a research assistant at the Computer Vision Laboratory in the Department of Computer Science at National Chiao Tung University from August 2009. His current research interests include computer vision, image processing, human-machine interfacing, and stereo vision.