國立交通大學

資訊學院

資訊科學與工程研究所

博士論文

利用環場視覺作自動車應用之定位與影像分析新技術之研究

New Localization and Image Adjustment Techniques Using Omni-Cameras for Autonomous Vehicle Applications

研究生: 吳至仁

指導教授: 蔡文祥 博士

中華民國 九十八 年 六 月

利用環場視覺作自動車應用之定位與影像分析新技術之研究

New Localization and Image Adjustment Techniques Using Omni-Cameras for Autonomous Vehicle Applications

研究生：吳至仁　　Student: Chih-Jen Wu

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 and Information Science

June 2009
Hsinchu, Taiwan, 30010
Republic of China

中華民國 九十八 年 六 月

利用環場視覺作自動車應用之定位與影像分析新技術之研究

研究生：吳 至 仁　　　　指導教授：蔡文祥博士

國立交通大學資訊學院

資訊科學與工程研究所

摘 要

許多自動車的應用中，定位是欲操控自動車所必備的功能。一個廣泛採用的方法是利用電腦視覺來作自動車定位，其方法是藉由電腦分析攝影機所攝得的影像來推算自動車本身的位置。近來新型態的環場攝影機漸漸被廣泛地應用在自動車定位上。相較於傳統攝影機，環場攝影機寬廣的視角可讓更大的景物範圍出現在視野中，更有助於利用視覺作定位的計算。然而，實際上環場攝影機擷取到的影像是扭曲的，造成影像處理上的困難，使定位工作變得難以進行，因此需要進行影像修正。

在本論文中，我們提出了一系列基於環場視覺的新的自動車定位與環場影像修正技術，並將其應用在各種自動車的用途上。我們使用了雙曲面反射鏡式與魚眼透鏡式的環場攝影機，而且，我們採用了路標式定位法，該法使用環境中明顯路標來作自動車的定位。在另一方面，我們在環場影像上，直接分析基本的路標幾何特徵（如直線與圓形等）的投影與投影之間的相互關係（如平行與垂直等）。並將前述所提出的方法應用在多種自動車的導航上，包含室內自動車導航、直升

機降落、汽車輔助駕駛等。更進一步地，我們提出了解決影像定位技術應用在自動車上常見的兩個問題的方法，一個是所謂扭曲環場影像的轉正方法，可用以解決自動車震動所導致的影像扭曲，另一個是一所謂空間與影像間的對映方法，此法可解決應用場合中攝影機重新安裝所導致的定位失效問題。這兩個方法讓前述所提自動車定位方法在實際應用中變得更為有效。茲將前述所提各種方法分為兩類—新自動車定位技術與新影像修正技術—詳細說明如下。

(A)針對自動車新定位技術 ---

(a)我們提出了一個描述環場影像中的圓形路標投影的新方法。在此法中，我們證明了此一投影可用橢圓來加以逼近，其中我們用了泰勒展開的技巧。以此，我們得以提出一個新法則來抽取環場影像中的橢圓狀投影。

(b)我們提出了一個利用圓錐曲線來描述環場影像中的直線投影的新方法。在此方法中我們推導出簡單而且有公式解的直線投影方程式，並接著提出可用以從環場影像中抽取圓錐曲線的簡便方法，其原理乃基於赫夫轉換。

(c)我們提出了一個以環場影像中的天花板圓形圖案，作為路標的室內自動車定位與導航之方法。此利用此路標具有容易偵測與辨識，且不易被遮蔽等優點。

(d)我們提出了一個利用環場影像中的 Y 形屋角影像，作為路標的室內自動車定位與導航之方法。因為燈光或拍攝角度的關係，Y 形直角並非每次都能完整的出現在環場影像中，所以我們分析了 Y 型屋角在環場影像中，所有可能出現的樣式，如點、垂直線、水平線及其可能組合樣式，並分別提出相對應的定位方法。

(e)我們提出了一個利用環場影像中的標準停機坪影像，作為路標的直升機降落之定位方法。在充分且有效的分析環場影像中停機坪圖案的幾何特徵，包含

圓形、水平線，加以參考直升機距離停機坪的遠近，我們提出了包含接近、對正與觸地的三階段定位方法。其中，求出的定位資訊包含高度、方位、與距離。

(f)我們提出了一個利用環場影像中的小客車車輪影像，作車側車輛定位方法。只要利用本身車輛上的一台環場攝影機所拍到的鄰車影像，再經分析中車輪影像後，此方法即可推算出鄰車相對於本身車輛的位置及方向，並具有公式解。

(B)針對新影像修正技術 ---

(a)我們提出了一個針對非置中環場攝影機所拍攝的扭曲影像，將其轉正的方法。造成非置中環場攝影機的現象，乃因在置中環場攝影結構中，原先的透鏡/反射鏡相對位置改變。造成此一改變的原因，常常是因為自動車的震動或攝影機的重新安裝。我們的方法可以當場解決此問題，而不必送回當初的攝影機製造廠作校正。

(b)我們提出了一個可應用於物體定位或自動車定位的空間對映方法，並能在實際應用時，適應攝影機高度與角度的改變。此方法基於空間對應表，可求出於多種型攝影機的影像座標與空間座標的對應關係，使得面臨實際環境時，該方法更加具有應用價值。

實驗結果顯示本論文提出的所有方法，皆具有優越性及有效性。最後，討論與未來可能研究方向也附於本論文中。

New Localization and Image Adjustment Techniques Using Omni-Cameras for Autonomous Vehicle Applications

Student: Chih-Jen Wu **Advisor: Dr. Wen-Hsiang Tsai**

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

Abstract

Vehicle localization is essential for autonomous vehicle guidance in many applications. A widely adopted approach is the vision-based technique by which the locations of vehicles can be computed by analyzing the images captured by cameras. Omni-cameras have become more and more popular recently for their wider field of views (FOVs). Wider FOVs make the job of localization easier because a larger scene range can be taken in a single shot. However, due to the distortion in the images taken by omni-cameras, it is difficult to use omni-cameras in vehicle localization applications unless taken omni-images are properly adjusted.

In this study, investigation of new omni-vision based vehicle localization techniques and omni-image adjustment, as well as their applications is conducted. Two types of omni-camera are used, including hyperboloidal omni-camera and fish-eye camera. Also, the landmark based approach to localization is adopted, in which obvious landmarks in vehicle navigation environments are utilized. On the

other hand, the projections of basic landmark features (lines, circles, etc.) in omni-images, and their relations (parallelism, perpendicularity, etc.) are analyzed mathematically. Accordingly, methods for varoius vehicle localization applications using the basic landmark features are proposed, including indoor vehicle guidance, helicopter landing, car driving assistance, etc. Furthermore, solutions to two vehicle localization problems frequently encountered with in real applications are also proposed, one being an omni-image unwarping method for dealing with image distortions caused by a misaligned omni-camera, and the other being a space-to-image mapping method which is adaptive to camera setup changes found in in-field environments. These two solutions make the proposed vehicle localization methods more effective in real environments. The above-mentioned proposed methods are summarized in the following, classified into two categories: new vehicle localization techniques and image adjustment techniques.

A. New vehicle localization techniques

(a) A new method for describing the projection of a circular-shaped landmark in omni-images is proposed. It is shown that such a projection may be approximated by an ellipse based on the application of Taylor expansion. In accordance, a new algorithm is designed for extracting the elliptical-shaped projections from omni-images of circular-shaped landmarks.

(b) A new method for describing the projection of a line in an omni-image as a conic section is proposed. Equations of such a projection are derived to be simple and analytic, and consequently uncomplicated effective image analysis algorithms are designed for extracting such conic sections out of omni-images by the Hough transform.

(c) A new method for vehicle localization by omni-vision for autonomous vehicle navigation in indoor environments using circular landmarks on ceilings is proposed. Such landmarks have several advantages can be identified, including ease to detect and recognize, and freedom from oclussion.

(d) Systematic vision-based vehicle localization techniques by hyperboloidal omni-cameras using Y-shaped house corners in indoor environments as landmarks are proposed. All possible partial structures of a house corner consisting of a corner point, a horizontal line, and a vertical one are considered, facilitating flexible vehicle localization under various lighting, occlusion, and imaging posture conditions.

(e) An omni-vision-based self-localization method for automatic helicopter landing on a helipad with a circled H-shape is proposed. The landing process includes three stages: approaching, alignment, and docking. Three types of image features, circle, line, and point, are used to derive skillfully analytic equations for computing the helicopter height, distance, and orientation with respect to the landing site.

(f) A lateral vehicle localization method by omni-image analysis is proposed for car driving assistance. The method estimates analytically the position and orientation of a lateral vehicle by utilizing the geometric properties of a circular-shaped wheel image of the lateral car taken by a single omni-camera.

B. Image adjustment techniques

(a) A new method for solving the problem of unwarping a distorted omni-image taken by a lateral-directionally misaligned omni-camera is proposed. Such camera misalignment is a frequently encountered problem of vision-based localization in real applications due to vehicle vibrations or camera redeployments. The method can solve this problem without camera calibration which is usually done in advance in the factory.

(b) A new space-mapping method for object location estimation or vehicle localization, which is adaptive to camera setup changes in application environments is proposed. The method, which is general for various types of cameras, estimates the location of an object appearing in an image by mapping the image coordinates of an object point to the real-world coordinates of the point using a space-mapping table. Such a method makes the space-mapping based approach to object localization more useful to real applications.

Good experimental results are shown to prove the feasibility and effectiveness of all the proposed methods. Discussions on possible future research directions are also included.

Acknowledgements

I would like to express my sincere appreciation to my advisor, Professor Wen-Hsiang Tsai, for his patience and kind guidance throughout the course of this dissertation study and the invaluable training. Thanks are also extended to the colleagues in the Computer Vision Laboratory at National Chiao Tung University for their valuable help during this study.

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

Table of Contents

Chapter 3 A Systematic Approach to Indoor Vision-Based Robot Localization Using Corner Features in Omni Images ..43

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation of Study

In the era of automation today, autonomous vehicles have been adopted in many application domains. Types of autonomous vehicles include mobile robot, land vehicle, computer-assisted car, helicopter, airplane, unmanned airborne vehicle, etc. In the study of autonomous vehicles, *vehicle localization* is indispensable in many applications, like navigation guidance, collision avoidance, automatic landing, driving assistance, etc. This problem is mentioned in the literature alternatively as *robot location*, *vehicle location estimation*, *object posture estimation*, etc. A widely adopted approach to vehicle localization is the vision-based technique, in which visual sensors like still and video cameras are used for environment sensing.

Most existing vision-based techniques deal only with frontal scenes acquired by traditional cameras and are easily interfered by unexpected objects around the vehicle. A feasible solution to this problem is to use an omni-directional camera (abbreviated as *omni-camera* in the sequel) [1][2][3][4][5][6][7] which looks at certain target shapes, usually called *landmarks*, attached on some objects in the environment. Uses of omni-cameras for solving this problem have the advantage of obtaining larger fields of view (FOVs) in the acquired omni-directional images (*omni-images*). Several factors should be considered in a solution of this kind, as discussed in the following.

(1) Types of omni-cameras ---

With the advance of camera technology, many types of omni-camera have been proposed. A survey of them will be given later in this chapter. The larger variety of them in general provides more convenience and effectiveness for vehicle location estimation. A systematic investigation of them is advantageous for improving the location estimation techniques.

(2) Types of landmarks ---

Many types of landmarks have been proposed in the past [1][6][10]. Although some specially-designed landmarks with good properties, like ease to extract from images of them, leading to analytic solutions, etc., have been proposed, they are not commonly seen in daily environments and need special arrangements which sometimes cause inconvenience. It is more desirable to use naturally-existing features in the environment, like straight lines, circles, rectangle, etc. The straight line might be the most commonly used landmark in the environment. It appears in indoor building structures such as baseboards, house corners, etc. Parameters from straight lines are widely studied for use in camera calibration. However, how to precisely and constantly acquire the parameters of lines is an important issue to solve. Circle is also a popular landmark shape. A circular-shaped landmark can be observed from any direction. Such rotational symmetry of the circle is a good landmark property, resulting in more precise and robust location.

(3) Locations of landmarks ---

Most landmarks proposed so far are located on the ground or attached on building walls, and so are apt to be occluded by people or objects around. An improper choice of the landmark location could lead to unreliable vehicle location estimation. On the contrary, a good landmark location will result in less landmark occlusion and image noise. Therefore, it is more convenient to use landmarks attached on the ceiling or affixed to other higher positions like house corners [8][9].

(4) Postures of omni-cameras ---

A conventional solution to the vehicle location estimation problem is to conduct a work of *omni-cameras calibration* to obtain a set of camera parameters, followed by the use the parameters to compute the object location work [1][23][55][56][57][58]. However, the calibration process is *sensitive to camera posture changes*. To solve such a *camera-posture sensitive* problem, a commonly way is to abandon the original calibration results and reconstruct a new one in the new camera-environment configuration.

(5) Misalignment of omni-cameras ---

In real applications like vision-based autonomous vehicle navigation or security surveillance [13][46][65][66][67], a camera equipped on a vehicle might be shaken due to vehicle vibrations or one installed on a wall might be removed due to re-employment, causing possibly destruction of the camera structure and resulting in *camera misalignment*, which causes displacements or/and re-orientations of the CCD camera with respect to the reflective mirror of the omni-camera. A feasible solution is to unwarp a distorted omni-image taken by a misaligned omni-camera.

In this dissertation study, we investigate vehicle localization and related image analysis techniques for autonomous vehicle applications using new types of omni-cameras. We try to design methods to solve the related problems mentioned above, aiming at providing more effective techniques or algorithms for practical application uses.

1.2 Survey of Related Works

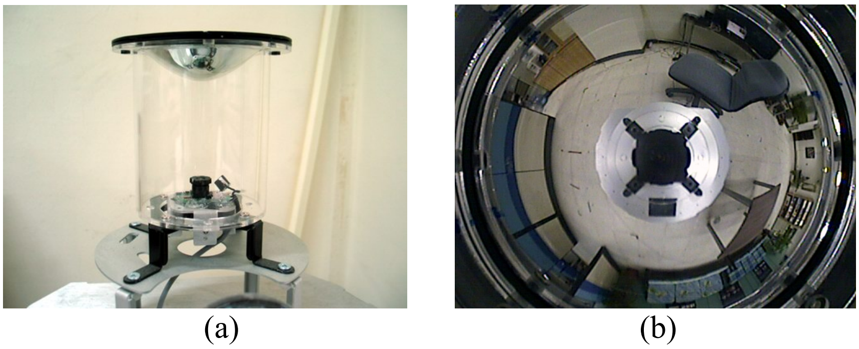
In this section, we survey works related to our study, including the categories of: (1) structures of various omni-cameras; (2) approaches to location estimation for different types of vehicle applications; and (3) image adaptation methods for adjusting images taken by misaligned or posture-slanted cameras for the purpose of precise vehicle localization. These three categories of related works will be reviewed in Sections 1.2.1, 1.2.2, and 1.2.3, respectively.

1.2.1. Survey of Types of Omni-cameras

New types of omni-cameras can be categorized into the following categories: (1) catadioptric camera; (2) dioptric camera; (3) two-mirror catadioptric camera; and (4) omni-camera pair. More details about each category are described 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 the mirror, a camera of this type can obtain larger FOVs 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 ellipsoidal, as illustrated in Fig. 1.2. With distinctive mirrors or lens, the images and calibration methods of the cameras are different in this category. Some works of vehicle location and image unwarping for this type of camera can be found in [1][2][3][4][5][6][7][8][9][10].



(a) (b)

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

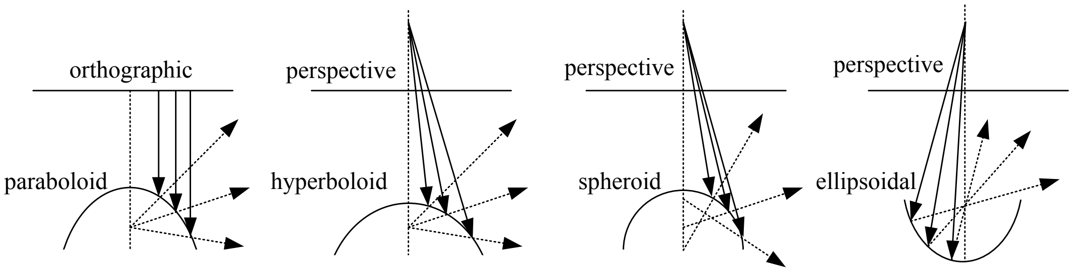


Fig. 1.2 Illustration of camera and reflective mirror type.

(2) Dioptric camera ---

A dioptric omni-camera, looking like a traditional camera, has no reflective mirror, but with a "wider-angle" lens. It can capture incoming light rays from a wider FOV to form an omni-image. An illustration of such imaging difference from 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 an fish-eye camera is shown in Fig. 1.4. Some works of vehicle location and image unwarping for fish-eye cameras can be found in [11][12][13].

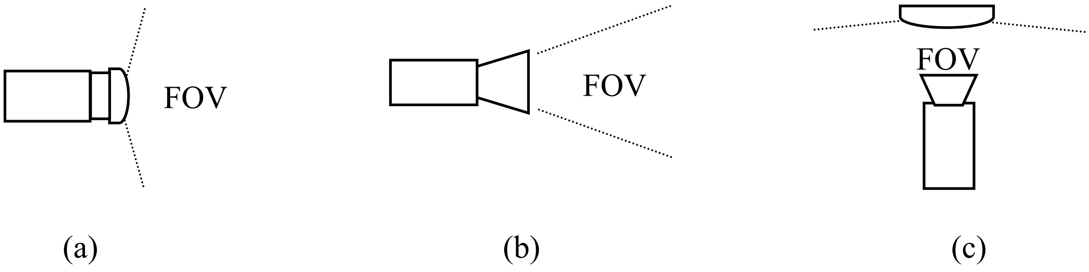


Fig. 1.3 FOVs of different camera types [14]. (a) Dioptric camera. (b) Traditional (perspective) camera. (c) Catadioptric camera.

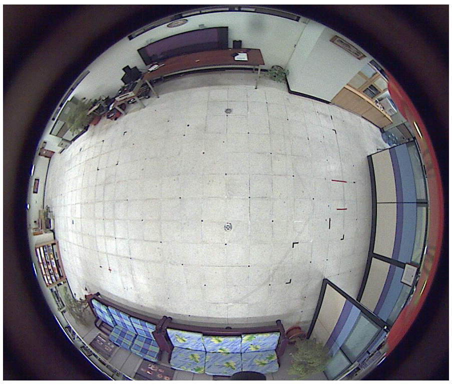
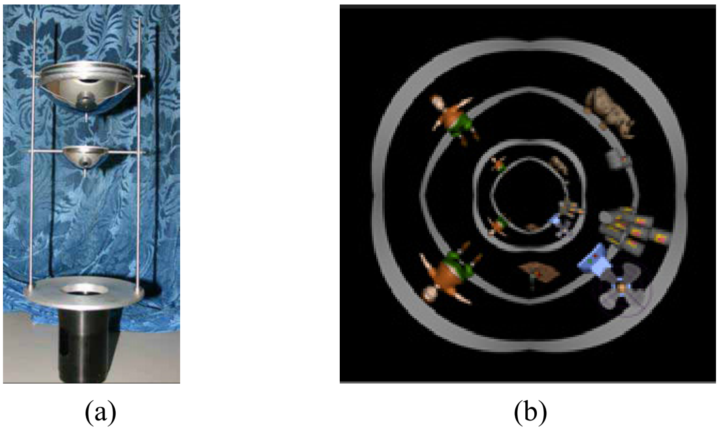


Fig. 1.4 An image acquired by a fish-eye camera.

(3) Two-mirror catadioptric camera ---

Recently, the design trend of the catadioptric camera is to combine two reflective mirrors in a single camera. We call the resulting camera a *two-mirror omni-camera* [15]. An example is shown in Fig. 1.5. The lights coming from a scene point in the real world space are reflected by each mirror into the CCD camera, causing two corresponding image points in the resulting single image. Such image pairs form two "image belts" of the same scene with different scales.

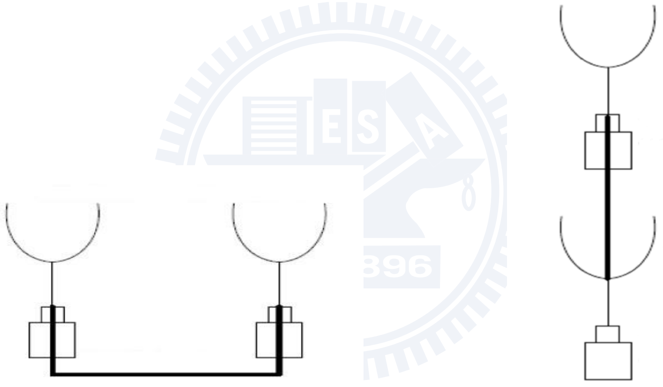


(a) (b)

Fig. 1.5 Two-mirror omni-camera [15]. (a) Structure of camera. (b) Acquired image.

(4) Omni-camera pair ---

An omni-camera pair consists of two omni-cameras with different locations. An illustration is shown in Fig. 1.6, 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 stereo information. However, most research works in the past focus on omni-camera pairs with hyperbolic-shaped reflective mirrors [16]; works for pairs with other mirror shapes are yet to be completed.



(a) (b)

Fig. 1.6 Omni-camera pairs. (a) Laterally paralell combinaiton. (b) Longitudinally coaxial combinaiton.

1.2.2. Survey of Localization Methods for Different Vehicle Applications

There are many appilications of autonomous vehicle localization techniques, such as: (1) indoor vehicle or robot guidance; (2) automatic helicopter landing; and (3) car driving assistance, etc. We review studies about these applications and related techniques in the following.

(1) Indoor vehicle or robot guidance ---

Indoor vehicle or robot guidance has been studied intensively in the past two decades. A frequently-adopted solution to this problem is to localize the vehicle or robot continuously in its navigation sessions, and guide its navigation path accordingly. The most commonly adopted vision-based approach aims to locate robots by analyzing images acquired with visual cameras [20][21][22]. Many kinds of techniques of this approach have been proposed, among which a popular one is the use of landmark images [1][23][24]. This kind of technique aims to compute the relative location of the vehicle or robot with respect to a landmark by analyzing the geometric hint exhibited by the landmark appearing in the acquired image. In such techniques, the location of the vehicle or robot is represented by both its position with respect to a reference point and its orientation with respect to a reference line included in the landmark, respectively, as we do in this study.

For *indoor* vision-based vehicle or robot localization, environmental structures abundant in buildings are often taken as landmarks. Among them, ceiling corners are the most commonly seen. They are good for vehicle localization because they are high and not easily occluded. In the past, there are few studies on house corners for robot localization [23][25], though there are related studies on using point and line features which compose the shapes of house corners [24][26]. For robot localization, Chou and Tsai [23] used a house corner which consists of three lines forming a shape of “Y”; Parlaktuna, et al. [25] used a range finder to detect the wall edges forming a house corner; Chen and Tsai [24] used lines on common object surfaces; and Park et al. [26] used scale-invariant feature points as visual landmarks. On the other hand, some studies on camera calibration, which is conceptually equivalent to the work of robot localization, focused on the investigation of using a sufficient number of corresponding lines for solving camera parameters [28].

The previously-mentioned studies were mostly based on the use of traditional projective cameras. There are only a few studies on the use of omni-cameras for vehicle localization [1][29][31]. Most of them are based on using global environmental features and matching them against environmental maps to locate the robot. There are also related camera calibration works [32][33], which were conducted from the viewpoint of enhancing calibration precision, and, when applied to vehicle localization, are too complicated for real-time navigation applications.

Most vehicle localization using omni-images proposed up to now can be grouped into three types: *triangulation*, *full-scene matching*, and *mirror-lens projection*, in accordance with the way of image information. In triangulation techniques [1][2], a standard location method is to identify surrounding landmarks in the environment and find their corresponding locations in the environment map built in advance. By the measured bearings of the landmarks [1], the location of an autonomous vehicle can be obtained. Yagi [2] correlated the angle of the vertical edge in an omni-image to the environment map to acquire the location of the vehicle. However, sometimes it is difficult to find natural landmarks which can be identified stably from omni-images.

In full-scene matching techniques [3][4], a vehicle locates itself by comparing images taken at its current location with reference images stored in its memory. Full-scene matching provides better feasibility by the use of omni-cameras. However, it takes a lot of reference memory space to raise location precision. Gasper [3] showed that the position of the vehicle in the environment can be determined by comparing the vehicle's current view with previously learned images, using a low-dimensional subspace of the input images obtained from a principal component analysis process.

Menegatti [4] simplified the location problem by using the Fourier components of omni-images as the signatures of the acquired views.

The unique mirror-lens projection relation of the omni-camera also can be used to derive the location of an autonomous vehicle by 3-D computer vision techniques [5][6]. In Koyasu [5], the range information, which is obtained by an omni-directional stereo vision system composed of a pair of vertically-aligned omni-cameras, helps not only creating a 3-D map but also locating the vehicle itself. In [6], Cauchois compared synthetic landmark images with real landmark images taken from a calibrated omni-camera to accomplish absolute vehicle location works. However, these methods might be unstable or inapplicable for locating a vehicle in a space crowded with people or full of objects, like in an exhibition room or in a library.

(2) Automatic helicopter landing ----

Many studies have been conducted for unmanned helicopter flying in the past [38][39][40][41][42][43][44][45]. Tsai and Yang [38] used parallel line information on a standard helipad as a hint for helicopter self-localization with respect to the circled H-shape on the helipad. See Fig. 1.7 for an illustration. The AVATAR project described in Saripalli, Montgomery, and Sukhatme [39] designed methods for landing a UAV in unstructured 3D environments using visual cues of the helipad described by moments. The work done by Shakernia, et al. [40] used point correspondence techniques to estimate the height and position of a UAV using multiple images. Garcia-Pardo et al. [41] presented techniques for locating circular landing areas using image processing operations like the contrast descriptor and correlation function. Mejias et al. [42] presented a vision-based feature tracking system for an autonomous helicopter, which estimates the position and velocity of window features in images and combines the result with GPS-positioning references to navigate the helicopter.

Tsai, Gibbens, and Stone [43] used the geometric properties of a standard landing mark of the T-shape to decide the vehicle position using single images. Yakimenko et al. [44] designed a method for shipboard landing of a UAV using infrared vision. Hespanha et al. [45] presented an integrated system to estimate the relative position and velocity of a UAV with respective to a ship using IR vision, inertial, and air data sensors.

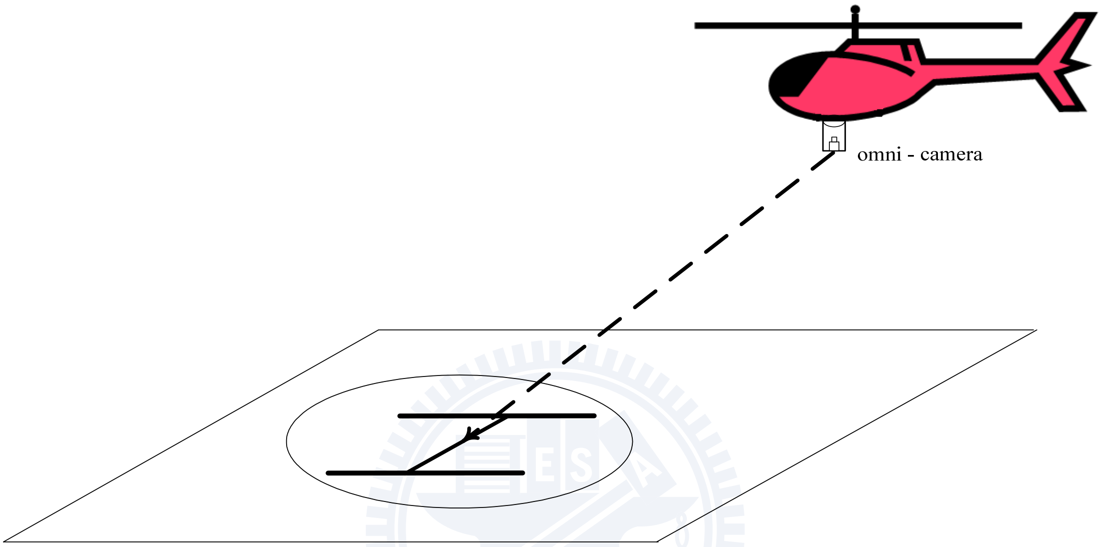


Fig. 1.7 Illustration of automatic helicopter landing on a helipad with a circled H-shape.

All the above methods used the traditional projective camera for visual sensing, which has a fixed FOV, compared with that of the recently commonly-used *omni-camera*. It is advantageous to use the omni-camera in automatic helicopter landing to enlarge the viewing scope and consequently speed up the automatic landing process. In this direction, Hrabar and Sukhatme [46] designed an omni-vision system which tries to find the centroid of the H-shape on the helipad to generate appropriate commands for guiding the helicopter. Demonceaux, Vasserur, and Pegard [47] proposed a helicopter posture computation method using catadioptric omni-images of the horizon line to estimate the pitch and roll angles of the helicopter. Bazin et al. [48] extended the method of [47] to estimate further the yaw angle of the helicopter using the information of vanishing points for automatic flight in urban areas.

(3) Car driving assistance ---

Car driving assistance using traditional cameras has been studied intensively [50][51][52][53]. Recently, omni-cameras with wider views become popular. They are more suitable for car driving assistance because fewer cameras need be equipped. For example, Lai and Tsai [52] affixed a traditional camera on the right-frontal side of a *host car* to take the image of a *lateral car*. To acquire a full frontal view, two more traditional cameras should be used. Instead, one frontal omni-camera is sufficient. Additionally, car wheels are circular-shaped, providing geometric hints for lateral car localization [52]. However, when a circle appears in an omni-image, it becomes irregular in shape and cannot be described mathematically [54], leading to difficulty of extending the existing vehicle localization methods for omni-images. This problem has been solved in this study.

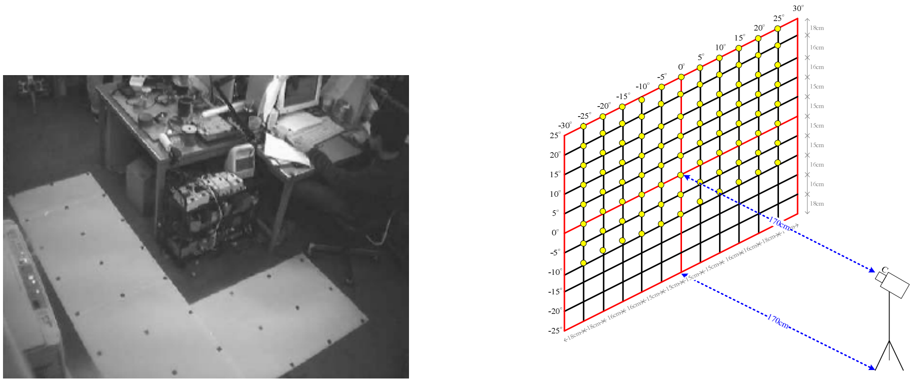
1.2.3. Survey of Image Adaptation Methods for Adjusting Images Taken by Misaligned or Posture-Slanted Cameras

Variou image adaptation methods for adjusting images taken by non-standardly-postured cameras, including: (1) adaptation to camera posture changes and (2) adjustment for misaligned omni-cameras, are surveyed as follows.

(1) Image adaptation to camera posture changes ---

A conventional solution to the object location estimation problem, which a *reverse* of the vehicle localization problem in concept, is to conduct a work of camera calibration to obtain a set of camera parameters, followed by the use of the parameters to compute the object location [1][23][55][56][57][58]. Camera calibration methods often use specially landmarks or environment features to derive formulas or algorithms to compute camera parameters. The computation process is in general complicated and time-consuming. The camera used in such methods is usually mounted on a robot or vehicle and so is mobile, while the landmark or feature used by the methods is usually fixed in the environment.

An alternative solution to the object location estimation problem is to use a *space-mapping* approach [36][59][60][61] which transforms the image space into the real-world space according to a *space-mapping table*. That is, the coordinates of an object point in an acquired image is mapped to the corresponding real-world coordinates of the point by a *table lookup* scheme. Thus, the above-mentioned camera calibration process is avoided, and the approach may be said to conduct *direct* object location estimation. The space-mapping table is constructed in advance, usually with the aid of a certain *calibration pattern*, before the camera is deployed in an application environment. Two examples of calibration patterns used in [59][60] are shown in Fig. 1.8, where a point pattern laid on the floor with a camera affixed on a ceiling was used in Takeshita, Tomizawa, and Ohya [59], and a grad pattern attached on a wall with the camera mounted on an autonomous vehicle was used in Wang and Tsai [60].

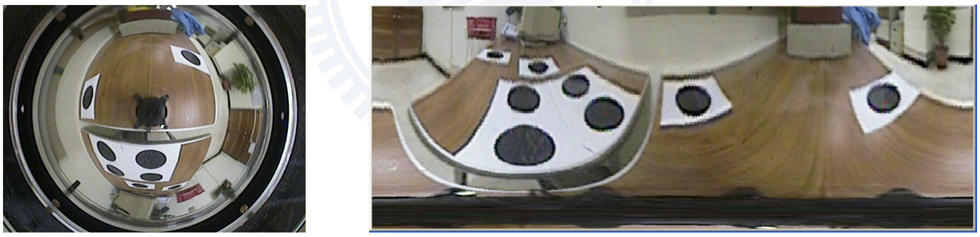


(a) (b)

Fig. 1.8 Calibration objects used for mapping table construction. (a) A point pattern used in Takeshita, et al. [59], laid on a floor. (b) A grid pattern used in Wang and Tsai [60], attached on a wall.

(2) Image adjustment for misaligned omni-cameras ---

Omni-camera misalignment occurs, for example, when the optical axis of the perspective camera of an omni-camera system are not coincident with the mirror axis which is perpendicular to the mirror base and through the mirror base center. Camera misalignment causes conventional image unwarping methods for image rectification inapplicable because of the resulting changes of the camera parameters. For an example of convenctional image unwarping, see Fig. 1.9. To solve this problem, Jeng and Tsai [71] proposed an omni-image unwarping method for dealing with the axial-directional camera misalignment problem. On the other hand, there are very few studies on image unwarping for lateral-directional camera misalignment so far except Mashita, Iwai, and Yachida [34] in which camera calibration is conducted first before image unwarping is carried out. In this study, we propose another method which is more convenient to apply and yields more accurate results.



(a) (b)

Fig. 1.9 Example of unwarping an omni-image into a panoramic perspective image. (a) Original omni-image. (b) Unwarping result of (a) which is a panoramic image.

1.3 Contributions of This Study

The major contributions of this study include proposing of the following techniques:

(1) A new method for proving and approximating circular shape images acquired by hyperboloidal omni-cameras as ellipses is proposed.

(2) A novel method for describing the projection of a space line on the omni-image plane as a conic-section and extracting it by a low-dimensional Hough transform is derived.

(3) A new localization method for indoor autonomous vehicle guidance using omni-images of circular landmarks on ceilings is proposed.

(4) A systematic investigation of possible indoor corner structures for applications of vehicle localization is conducted and relevant image analysis techniques are proposed.

(5) Analytic formulas for vehicle localization using single-view images of all possible house corner structures are derived for fast computation and real-time indoor vehicle guidance.

(6) A novel omni-vision based self-localization method for automatic helicopter landing on standard helipads is proposed.

(7) A new method for localization of lateral vehicles for car driving assistance using omni-images is proposed.

(8) A new approach to adaptation of existing space-mapping methods for object localization to camera setup changes is proposed.

(9) An unwarping method for images taken by misaligned omni-cameras without camera calibration by curved quadrilateral morphing using quadratic pattern classifiers is proposed.

An illustration of the relations of the above-mentioned techniques is shown in Fig 1.10 where each technique is categorized into three groups: image adjustment, new feature analysis, and new localization techniques and applications, colored by three different blocks. Moreover, the mutual relations among the techniques are annotated by arrows. That is, between two neighboring techniques, the direction of the arrow means that the previous technique can be applied to the following one.

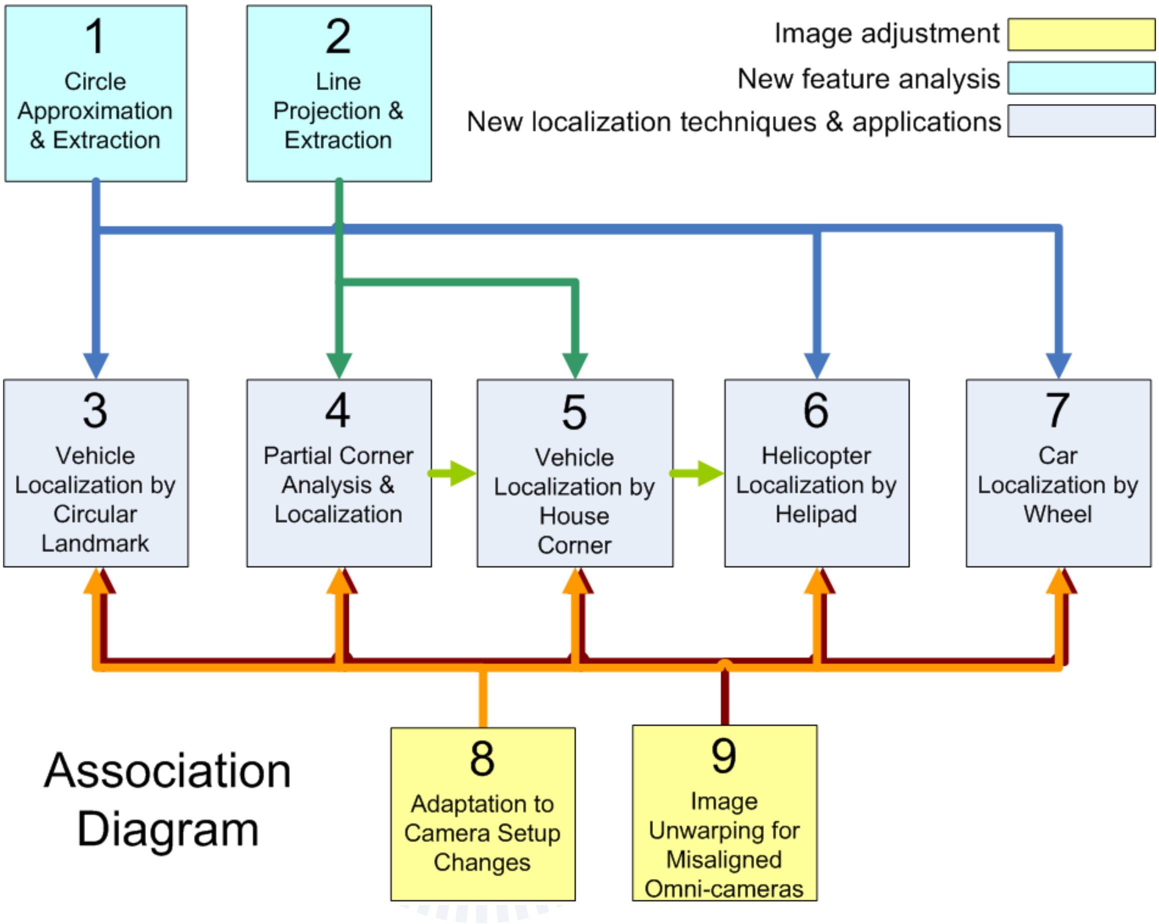


Image adjustment
New feature analysis
New localization techniques & applications
1 Circle Approximation & Extraction
2 Line Projection & Extraction
3 Vehicle Localization by Circular Landmark
4 Partial Corner Analysis & Localization
5 Vehicle Localization by House Corner
6 Helicopter Localization by Helipad
7 Car Localization by Wheel
8 Adaptation to Camera Setup Changes
9 Image Unwarping for Misaligned Omni-cameras
Association Diagram

Fig. 1.10 Relations among the proposed techniques in this study.

1.4 Dissertation Organization

In the remainder of this thesis, we describe the proposed methods for various applications in the chapters, respectively. In Chapter 2, we propose a method for location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings. In Chapter 3, we propose a systematic approach to indoor vision-based robot localization using corner features in omni-images. In Chapter 4, we propose an omni-vision based self-localization method for automatic helicopter landing on standard helipads. In Chapter 5, we propose a method for omni-vision based localization of lateral vehicles for car driving assistance. In Chapter 6, we propose a technique for adaptation of space-mapping methods for object location estimation to camera setup changes. In Chapter 7, we propose a method for unwarping of images taken by misaligned omni-cameras without camera calibration by curved quadrilateral morphing using quadratic pattern classifiers. In each chapter, we describe relevant techniques and applications of the proposed methods, and include experimental results to show the feasibility of the methods. Discussions and suggestions for future studies are also included.

Chapter 2
Location Estimation for Indoor Autonomous Vehicle Guidance by Omni-vision Using Circular Landmarks on Ceilings

2.1 Idea of Proposed Method

Vehicle localization is essential for guidance of autonomous vehicles in many indoor navigation applications. Most existing vision-based techniques deal only with frontal scenes acquired by traditional cameras and are easily interfered by unexpected objects around the vehicle. A feasible solution to this problem is to use an omni-camera which looks upward at certain landmarks attached on the ceiling [7]. This solution has the unique advantage of providing wide-angle views with fewer objects appearing in the FOV, thus reducing the guidance error coming from landmark occlusion, noise inference, etc. This is important for applications of intelligence robots such as cleaning robots, pet robots, tour guide robots, etc., which must work among humans or objects at close distances. On the other hand, even though obtaining the distance and orientation of the circular landmarks on the ceiling can be easily realized with a traditional perspective camera [8][9], a well-designed single omni-camera system may be used to replace several standard cameras so far as the image taking range is concerned.

In this study, a location estimation method for indoor autonomous vehicle guidance using omni-images of circular landmarks on ceilings is proposed. Analysis of circular shapes in omni-images is not well studied so far. It is found in this study that a circular shape, which becomes an irregular shape in an omni-image with no known shape descriptor, can be well approximated analytically by an elliptical shape. Consequently, it is appropriate to guide a vehicle equipped with an upward-looking omni-camera using a circular shape attached on a ceiling as a landmark, as is done in this study. Several merits can be identified in this approach, including: (1) the circular-shaped landmark attached on the ceiling is identically observable from every direction; (2) the circular shape, being elliptical when imaged, is easier to detect in low-resolution omni-images; (3) the elliptical shape provides more precise parameters for location estimation; (4) the elliptical shape does not get mixed up easily with other shapes found in the environment. Owing to these merits, stable and precise relative vehicle location estimation can be achieved for navigation. An illustration of the experimental navigation environment for this study, including a vehicle, a ceiling, and a landmark, is shown in Fig. 2.1.

In the proposed method, at first an upward-looking omni-camera on a vehicle is used to take an image of a circular-shaped landmark attached on the ceiling of an indoor space. An ellipse detection algorithm [17] is applied next to detect the projected shape of the landmark in the image. The irregular shape formed from the circular shape in the omni-image is approximated by an elliptical shape. The location of the landmark, including its distance and orientation, with respect to the camera on the vehicle are then derived analytically in terms of the major axis length and the center coordinates of the approximating ellipse. Finally, the move distance and the orientation change of the vehicle between two consecutive observations of the landmark are derived, which are useful for a number of autonomous vehicle applications.

The remainder of this chapter is organized as follows. In Section 2.2, we derive the analytic equations for approximating as an elliptical shape the irregular shape in an omni-image taken of a circular-shaped landmark. In Section 2.3, we describe how we estimate the vehicle location from the acquired image using the derived analytic equation, and show an application of the results to autonomous vehicle guidance. In Section 2.4, some experimental results are described to show the precision and feasibility of the proposed method. Finally, some conclusions are given in Section 2.5.

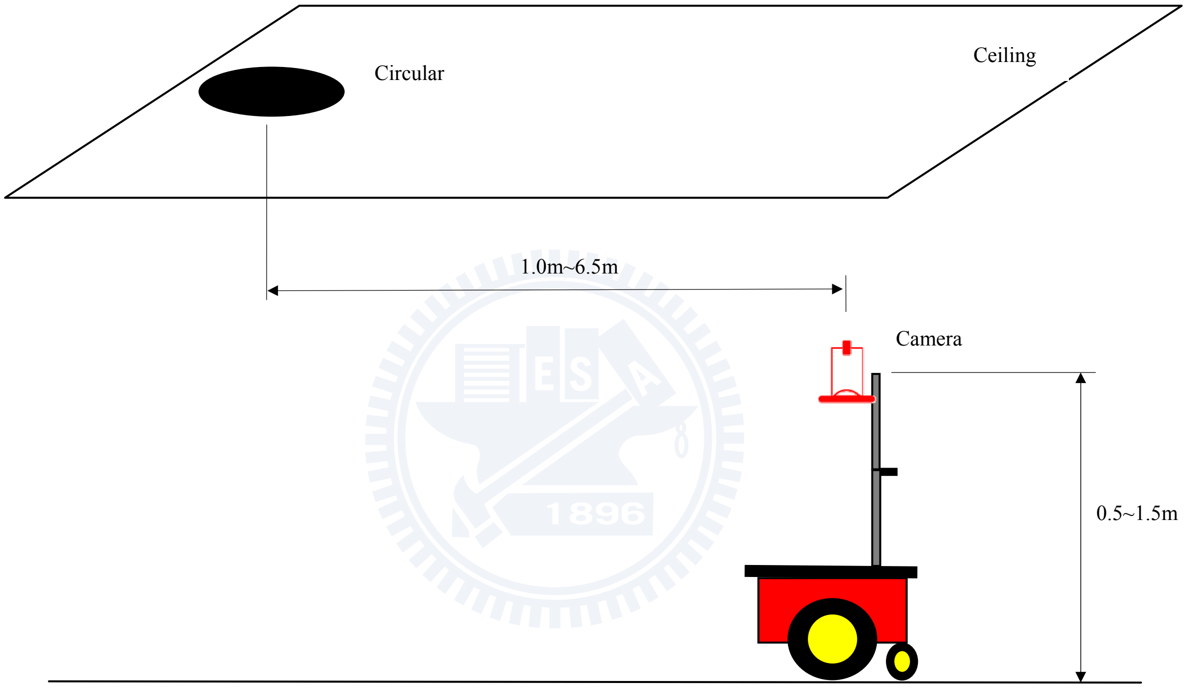


Fig. 2.1 Relative positions of camera, ceiling, and circular landmark for providing sufficient field of view and avoiding unexpected objects and humans appearing in acquired images.

2.2 Approximation of Irregular Shape in Omni-image Taken of Circular-shaped Landmark by Ellipse

The circular shape attached on the ceiling of the vehicle navigation environment for use as a landmark becomes irregular with no mathematical shape descriptor in an omni-image taken with a hyperboloidal omni-camera. We can approximate the irregular shape well by an ellipse, as mentioned previously, and this fact will be proved here. Specifically, an equation of the approximating elliptical shape in the image will be derived. The validity of this ellipse approximation will become clear in the derivation. The precision of the approximation will also be proved by some experimental results.

In Section 2.2.1, the projection transformation between the camera coordinate system and the image coordinate system will be described first. Then, the coordinate systems will be rotated horizontally to derive the equation of an ellipse in the image. In Section 2.2.2 a simulated shape of the circular landmark computed with the derived equation will be compared with the shape obtained by an imaging projection based on [5] to show the effectiveness of the proposed elliptical shape approximation.

2.2.1. Approximation of Distorted Circular Shapes in Omni-images by Ellipses

The camera and image coordinate systems involved in this study using a hyperboloidal omni-camera are depicted in Fig. 2.2, with their coordinates specified by (X, Y, Z), and (u, v), respectively. The hyperbolic shape of the omni-directional mirror in the camera coordinate system may be described as:

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

The focal point O_M of the mirror is located at (0, 0, $-c$) and the camera center O_C at (0, 0, $+c$), in the camera coordinate system, where $c = \sqrt{a^2 + b^2}$. The projection relationship between the image coordinates (u, v) and the camera coordinates (X, Y, Z) can be described as follows [5][34][35][36]:

$$u = \frac{Xf(b^2 - c^2)}{(b^2 + c^2)(\mathrm{Z} - c) - 2bc\sqrt{(\mathrm{Z} - c)^2 + X^2 + Y^2}},$$

$$v = \frac{Yf(b^2 - c^2)}{(b^2 + c^2)(\mathrm{Z} - c) - 2bc\sqrt{(\mathrm{Z} - c)^2 + X^2 + Y^2}}, \tag{2.2}$$

where f is the focal length of the camera.

In Fig. 2.2, let the circular-shaped landmark and its center and radius be denoted by W, P_w and R_w, respectively. And let the image of W in the hyperboloidal image be denoted by Q. Also, let (X_w, Y_w, Z_w) denote the camera coordinates of P_w. In this study, the normal vector of the landmark is assumed to be parallel to the optical axis of the camera. To simplify the derivation described later, we rotate, as shown in Fig. 2.3, horizontally the camera coordinate system and the image coordinate system through an angle of θ_w defined by:

$$\theta_w = tan^{-1}\frac{Y_w}{X_w}. \tag{2.3}$$

Then, the relation between the original camera coordinates (X, Y, Z) and the resulting ones (X', Y', Z') may be described by:

$$X' = X\cos\theta_w + Y\sin\theta_w,\quad Y' = Y\cos\theta_w\ - X\sin\theta_w,\ Z' = Z \tag{2.4}$$

and the relation between the original image coordinates (u, v) and the resulting ones (u', v') may be described by:

$$u' = u\cos\theta_w + v\sin\theta_w,\quad v' = u\sin\theta_w - v\cos\theta_w. \tag{2.5}$$

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

$$(X' - X_w')^2 + (Y' - Y_w')^2 = R_w{}^2,\ Z' = Z_w',$$

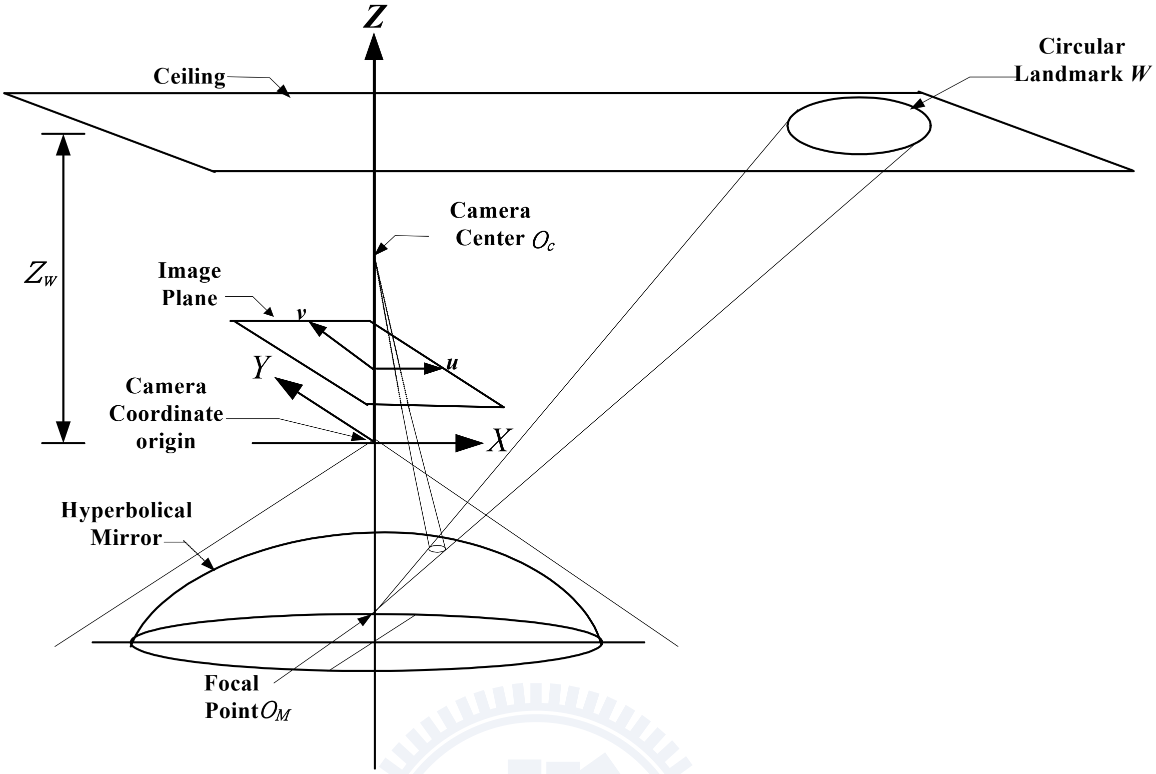


Fig. 2:2 Coordinate systems involved in this study.

$$(X' - X_w')^2 + (Y' - Y_w')^2 = R_w{}^2, \quad Z' = Z_w',$$

given that the center point P_w of W is located at (X_w', Y_w', Z_w') with $X_w' = X_w\cos\theta_w + Y_w\sin\theta_w$, $Y_w' = X_w\sin\theta_w - Y_w\cos\theta_w$, $Z_w' = Z_w$ according to (2.4). Notice that Y_w' is now zero after the rotation according to Fig. 2.3, so that the above equation becomes

$$(X' - X_w')^2 + Y'^2 = R_w{}^2,\ Z' = Z_w'. \qquad (2.6)$$

Also, by the optical geometry of the camera described by (2.2), we have

$$\frac{v'}{u'} = \frac{Y'}{Z'}, \qquad (2.7)$$

or equivalently,

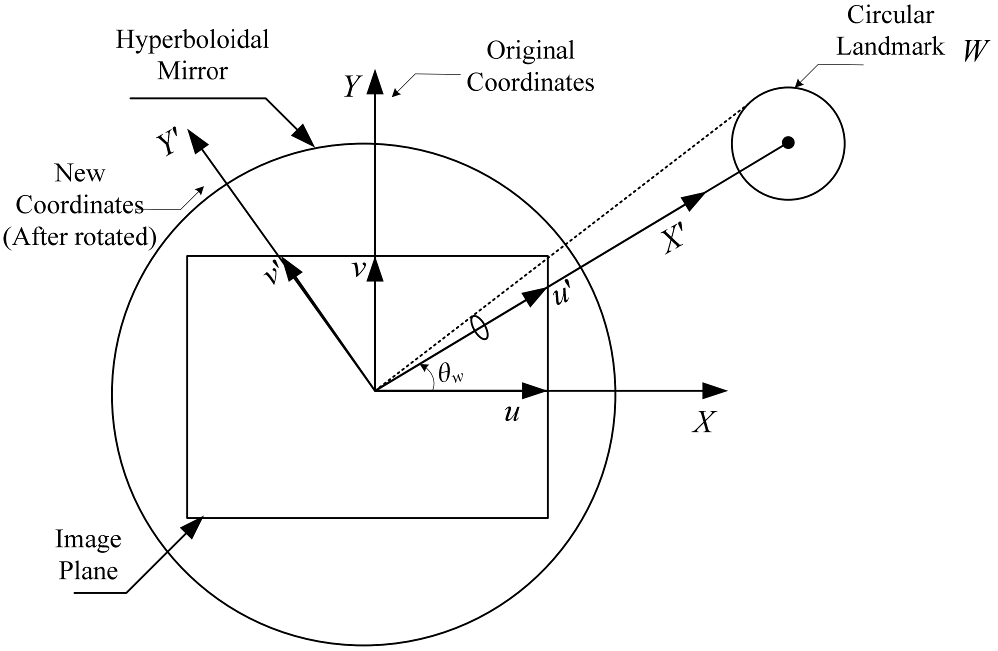


Fig 2.3 Top view from the Z direction showing the relationship between new and original coordinate system with the new image coordinate system (u', v') obtained by rotating the u-axis through an angle of θ_w = $\tan^{-1}(Y_w/X_w)$ with respect to the center of the circular-shaped landmark W.

$$v' = \frac{u'}{X'}Y' . \tag{2.8}$$

Eq. (2.8) will be used in Section 2.3.1 later.

We are now ready to prove the previously-mentioned fact that the irregular shape of the circular landmark W appearing in the omni-image may be well approximated by an ellipse. After horizontally rotating the camera and the image coordinate systems for the angle of $\theta_w = \tan^{-1}(Y_w/X_w)$ described by Eq. (2.3), Eq. (2.2) becomes

$$u' = \frac{X' f(b^2 - c^2)}{(b^2 + c^2)(Z' - c) - 2bc\sqrt{(Z' - c)^2 + X'^2 + Y'^2}}$$

$$v' = \frac{Y' f(b^2 - c^2)}{(b^2 + c^2)(Z' - c) - 2bc\sqrt{(Z' - c)^2 + X'^2 + Y'^2}} . \tag{2.2A}$$

In addition, the new Y-coordinates Y_w' of the landmark circle center is 0. Then, by assuming that the horizontal distance from the origin of the camera coordinate system to the landmark is much larger than the radius of the landmark, we have $Y' \ll X'$ and the circle $(X' - X_w')^2 + (Y' - Y_w')^2 = R_w^2$ with $Y_w' = 0$ may be regarded relatively as a point which is its center located at (X_w', Y_w') so that $X'^2 + Y'^2 \approx X_w'^2 + Y_w'^2 = X_w'^2$. Also, Z' is a constant (denoted as h_w now) because the ceiling on which the landmark is attached is assume to be parallel to the camera coordinate system. As a consequence, the second Eq. in (2.2A) above for v' can be simplified to

$$v' = \frac{Y' f(b^2 - c^2)}{(b^2 + c^2)(\mathrm{Z} - c) - 2bc\sqrt{(\mathrm{Z} - c)^2 + x_W'^2}}$$

or equivalently,

$$Y' = v'M \tag{2.9}$$

where M is

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

The other coordinate u' of each shape pixel of the landmark W may also be derived by approximation, but in a different way. Under the same assumption mentioned above that the radius of the landmark is relatively very small with respect to the horizontal distance from the origin of the camera coordinate system to the landmark, the magnitude of X' of each shape pixel of the circular landmark W in the camera coordinate system is much larger than that of Y'. Therefore, we may neglect the influence of the magnitude of Y' in the computation of u' described by the first equation in (2.2A) so that

$$u' = \frac{X' f(b^2 - c^2)}{(b^2 + c^2)(\mathrm{Z} - c) - 2bc\sqrt{(\mathrm{Z} - c)^2 + X'^2}}$$

and compute u' just in terms of X'. Regarding the above equation in the form $u' = F(X')$, we may use the Taylor series to expand the function around X_w' as

$$u' = F(X') = F(X_w') + [(X' - X_w')/1!]F'(X_w') + [(X' - X_w')^2/2!]F''(X_w') + \ldots.$$

Ignoring the terms after the second, we have

$$u' \approx F(X_w') + [(X' - X_w')/1!]F'(X_w') = u_w' + (X' - X_w')F'(X_w'). \quad (2.10)$$

Eq. (2.10) may be transformed easily into

$$X' \approx X_w' + (u' - u_w')/F'(X_w') \quad (2.11)$$

with the first derivative F' calculated to be:

$$F'(X_w') = A_E\left(\frac{1}{B_E} - \frac{C_E X_w'}{B_E{}^2\sqrt{D_E + X_w'}}\right), \quad (2.12)$$

where

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

$$B_E = (b^2 + c^2)(z - c) - C\sqrt{D + X_w'^2}\ ,$$

$$C_E = 2bc\ ,$$

$$D_E = (z - c)^2$$

$$Z = h_w.$$

Now with X' and Y' available, we come to the final stage of the derivation of the equation of the ellipse for approximating the distorted circular shape of the landmark in the omni-image. By substituting Eqs. (2.9) and (2.11) into Eq. (2.6) and rearranging the result, we can get

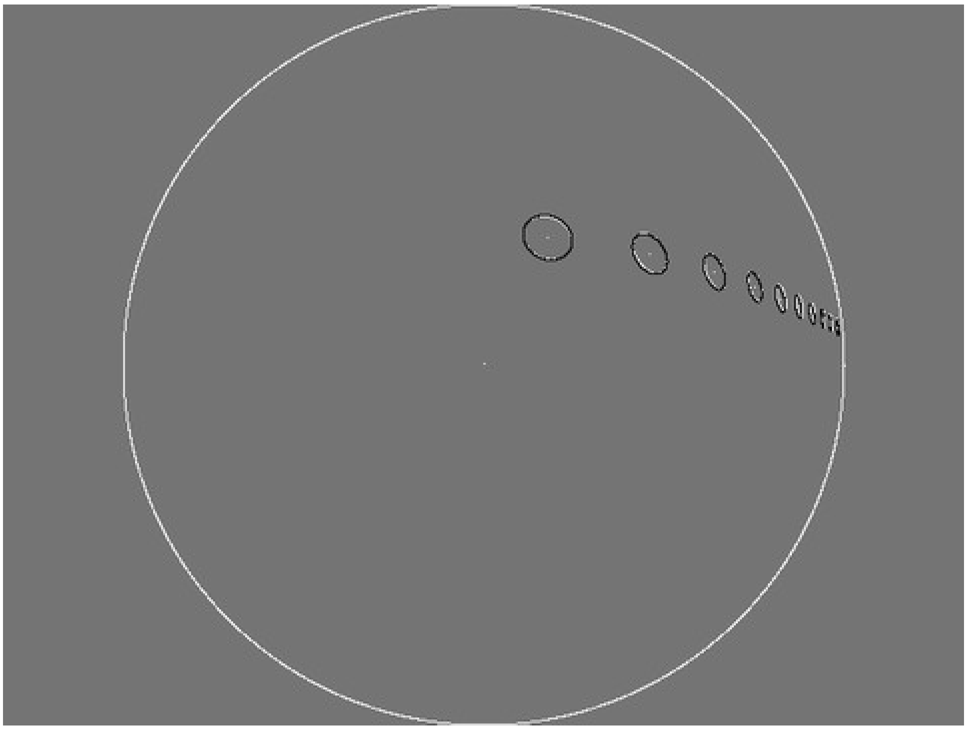
$$\frac{(u' - u_w')^2}{R_w^{\ 2} F'\left(X_w'\right)^2} + \frac{v'^2 M^{\ 2}}{R_w^{\ 2}} = 1 \qquad (2.13)$$

which obviously specifies *exactly* an elliptical shape centered at (u_w', 0) with the lengths of the major and minor axes being $R_wF'(X_w')$ and R_w/M, respectively. This completes the proof.

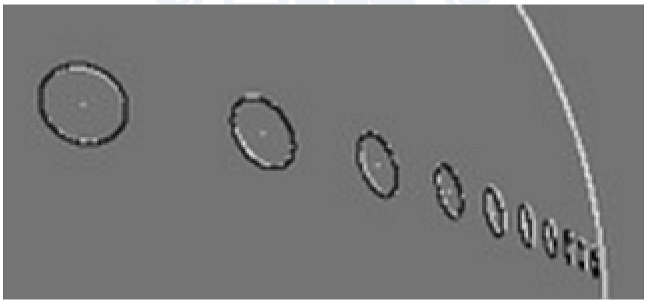
2.2.2. Effectiveness of Shape Approximation

To check the effectiveness of the approximation of the circular shape of the landmark by the elliptical shape using Eq. (2.13), we show in Fig. 2.4 an example of the simulation results obtained in this study, in which both the original circular shape and the approximating elliptical one are drawn and superimposed on each other for comparison: the former shape being drawn by Eq. (2.6) and then projected into the image plane by Eq. (2.2), and the latter being drawn directly by Eq. (2.13). The outer big circle in the figure marks the field of view of the camera. Inside the big circle, the original distorted circular shapes of W at different positions are drawn with white pixels, and the approximate elliptical shapes are computed using the coordinates of the white pixels and drawn by black pixels.

From the figure, we can see that each black ellipse overlaps the corresponding white distorted circle quite well. This shows that the distorted circular shape of the landmark in the omni-image indeed may be approximated by the ellipse described by Eq. (2.13). This discovery offers great helps, as found in this study, in utilizing this kind of circular landmark to provide the location information for vehicle guidance, as described in the following section.



(a)



(b)

Fig. 2.4 Simulation of a series of circular shapes of the landmark at different places, showing that the distorted landmark shape may be approximated well by ellipses. (a) illustration of the simulation results. (b) partially enlarged view of (a).

2.3 Vehicle Location Estimation

In this section we describe the proposed method for vehicle location estimation using the detected landmark as a known reference point. From each acquired image of the ceiling, we extract the circular landmark shape by image processing techniques, including thresholding, edge detection, and ellipse detection. For thresholding, since the artificially-made circular-shaped landmark contrasts well with the background in the image, a threshold value is selected to segment the distorted circular shape of the landmark in the image. And for edge detection in the image, the Sobel edge detector is applied. Since the landmark shape in the image can be regarded as an ellipse according to Section 2.2, an ellipse detection algorithm [17] is employed to extract the landmark, yielding an approximating ellipse with its axis lengths computed.

2.3.1. Vehicle Location Estimation by Axis Lengths of Ellipse

We will derive here the location of the landmark, including its distance and orientation, with respect to the camera coordinate system on the vehicle for use in vehicle guidance. We derive first the horizontal distance of the landmark in terms of the lengths of the two axes of the approximating ellipse. As depicted in Fig 2.3, the X' axis is directed to the landmark center P_w'. For a better view of the involved situation, Fig.2.3 is redrawn as Fig. 2.5 in which d is the horizontal distance between the camera and the center of the landmark W to be derived; $\overline{P_\alpha' P_\beta'}$ is a diameter of W perpendicular to the X' axis; point I_α' is the image of point P_α'; and point I_β' is the image of point P_β'. It follows from Eq. (2.7) that

$$\frac{v_\alpha'}{u_\alpha'} = \frac{Y_\alpha'}{X_\alpha'} \tag{2.14}$$

where (u_α', v_α') are the coordinates of I_α' in the (u', v') coordinate system and (X_α', Y_α') are the coordinates of P_α' in the (X', Y') coordinate system. Similarly, we have

$$\frac{v_\beta'}{u_\beta'} = \frac{Y_\beta'}{X_\beta'} \tag{2.15}$$

where (u_β', v_β') are the coordinates of I_β' in the (u', v') coordinate system and (X_β', Y_β')

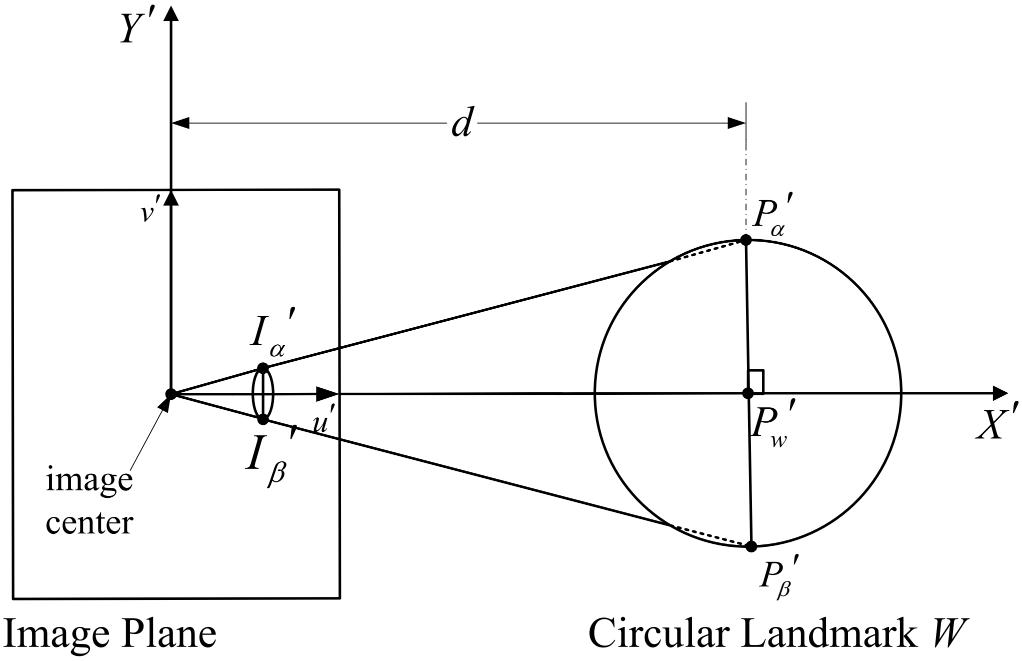


Fig. 2.5 Top view from the Z direction illustrating the relation between the axes of the approximating ellipse and the horizontal distance of the circular-shaped landmark.

are the coordinates of P_β' in the (X', Y') coordinate system. Since $\overline{P_\alpha' P_\beta'}$ is the diameter of W perpendicular to the X' axis, we have $X_\alpha' = X_\beta'$ and so $u_\alpha' = u_\beta'$. Therefore, Eqs. (2.14) and (2.15) can be merged and rearranged to result in

$$\frac{v_\alpha' - v_\beta'}{u_\alpha'} = \frac{Y_\alpha' - Y_\beta'}{X_\alpha'} \tag{2.16}$$

where $Y_\alpha' -\ Y_\beta'$ is the length of $\overline{P_\alpha' P_\beta'}$ (denoted as $\left\|\overline{P_\alpha' P_\beta'}\right\|$) which is the known in advance; $v_\alpha' - v_\beta'$ is the length of $\overline{I_\alpha' I_\beta'}$ (denoted as $\left\|\overline{I_\alpha' I_\beta'}\right\|$) which is the major axis of the approximating ellipse in the image plane; and u_α' is the u' coordinate of the center of the ellipse. The latter two parameters $\left\|\overline{I_\alpha' I_\beta'}\right\|$ and u_α' can be computed right after the approximating ellipse is obtained. Also, as seen from the figure, X_α' is just the desired relative horizontal distance *d* between the camera and the center of the landmark *W*, or equivalently, the desired horizontal distance *d* of the landmark with respect to the vehicle. Therefore, we can derive from Eq. (2.16) the desired value of *d* as

$$d = \frac{\left\| \overline{P_\alpha' P_\beta'} \right\|}{\left\| \overline{I_\alpha' I_\beta'} \right\|} \times u_\alpha' . \tag{2.17}$$

Furthermore, using the ellipse makes it easier to solve the orientation θ_w of the landmark with respect to the camera coordinate system, or equivalently, with respective to the vehicle. Given that the center of the found ellipse in the image is located at (u_w, v_w), from Eq. (2.3) and the following equation derived from Equation (2.7),

$$\frac{Y_w}{X_w} = \frac{v_w}{u_w},$$

we can calculate θ_w by

$$\theta_w = \tan^{-1}\left(\frac{Y_w}{X_w}\right) = \tan^{-1}\left(\frac{v_w}{u_w}\right). \tag{2.18}$$

A merit of the above procedure is that there is no need of the vertical height value Z_w of the landmark with respect to the camera coordinate system in computing the distance *d* and the orientation θ_w of the landmark. This provides an advantage of allowing dynamic changes of the camera height for the purpose of avoiding camera view occlusion by surrounding people or objects. Taking this advantage, we have included a shaft in the vehicle system for adjusting the height of the camera dynamically: if the landmark shape cannot be well extracted from the image taken at a certain camera height to yield an approximating ellipse, then the camera is lifted up automatically and gradually until a good result is obtained.

Another merit of vehicle location estimation by Eqs. (2.17) and (2.18) is that the exact hyperbolic mirror shape of the omni-camera, as described in Eq. (2.1), need not be considered. This reduces the estimation error caused by using the possibly imprecise shape parameters of the mirror obtained from calibration, and simplifies the computation process involved in the estimation. Also, it provides a nature of generality of the proposed location estimation process using a given type of camera, so that it is unnecessary to re-design the location estimation process when a camera with a different hyperbolic mirror shape is used.

2.3.2. Estimation of Vehicle Moving Distances and Orientation Changes

In the previous section, we show how to estimate the distance and orientation of the landmark with respect to the vehicle. In this section we show how to find out, as an application of the previous results, the relative distance and orientation of the vehicle in a navigation cycle with respect to its location in the previous cycle, which we call the *move distance* and *orientation change* of the vehicle, respectively. Derivations of these parameters are useful for a number of autonomous vehicle applications, including recording of navigation paths, path planning for vehicle guidance, measurement of guidance errors, etc.

As shown in Figs. 2.6(a) and 2.6(b), we denote the move distance of the vehicle between two consecutive observation times T_1 and T_2 as D. Using the approximating ellipse, we can, according to Section 2.3.1, calculate the horizontal distances d_1, d_2 and the orientations θ_1, θ_2 of the landmark at T_1 and T_2, respectively, with respect to the vehicle. Based on the cosine theorem, we have

$$d_1^2 + D^2 - 2d_1 \times D \times \cos\theta_1 = d_2^2,$$

which may be solved to get the desired move distance D as

$$D = d_1 \cos\theta_1 - \sqrt{d_1^2 \cos^2\theta_1 - d_1^2 + d_2^2} \qquad \text{if } \theta_2 \leq 90^{\text{o}}\text{; or} \tag{2.19}$$

$$D = d_1 \cos\theta_1 + \sqrt{d_1^2 \cos^2\theta_1 - d_1^2 + d_2^2} \qquad \text{if} \quad 2 > 90^{\text{o}}. \tag{2.20}$$

Next, to find the desired orientation change of the vehicle, which we denote as γ, caused by a vehicle turning between the two consecutive observations, we acquire first, using Eq. (2.18), the orientations of the landmark with respect to the vehicle before and after the vehicle turning, denoted as θ_2 and ϕ, respectively, as depicted in Figs. 2.6 (c) and 2.6(d). Then, γ can be computed easily by

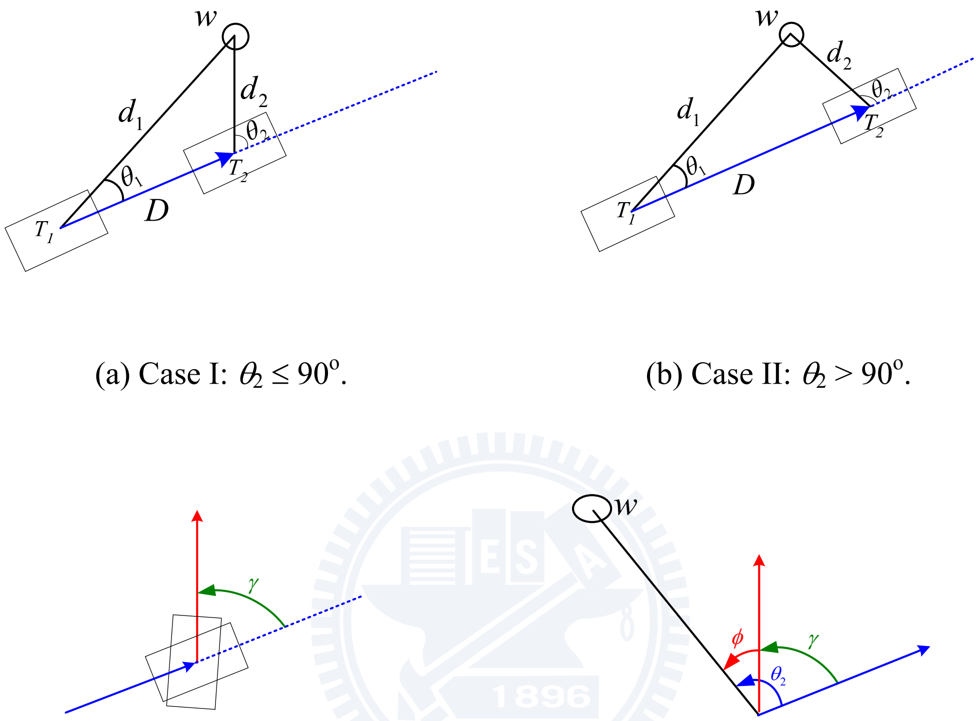
$$\gamma = \theta_2 - \phi. \tag{2.21}$$

2.4 Experimental Results

The effectiveness of the proposed location estimation method has been tested by some experiments conducted in this study, which include two parts: (1) using computer simulations to test if the circular shape of the landmark in the acquired images taken with omni-cameras with different shapes of hyperboloidal mirrors can be detected by the proposed ellipse approximation method; (2) using real images to determine the precision of the estimated vehicle locations relative to the landmark.

(A) Simulations

In the first experiment of landmark shape approximation by ellipses, the first step was to create landmark shapes at different locations. A series of virtual circular-shaped landmarks with a radius of 20cm were created and projected onto the image plane, in which the centers of these landmarks are located in a range of 300cm in intervals of 50cm with the orientation angles θ ranging from 0° to 360° in intervals of 5°. Two kinds of common distortion, the barrel and the pincushion distortion as described in [18][19], were added to make the simulation more realistic. In our simulations, the original image coordinates (u, v) are inherently normal and distortion-free, as described by Eq. (2.2).



(a) Case I: $\theta_2 \leq 90^{\circ}$.

(b) Case II: $\theta_2 > 90^{\circ}$.

(c)

(d)

Fig. 2.6 Illustration of the relative ALV location estimation. (a) and (b): the displacement D of the vehicle. (c) and (d): the orientation of the vehicle.

Then, we add barrel and pincushion distortion to them by the following equations:

$$\breve{u} = u + u[k_1(u^2 + v^2) + k_2(u^2 + v^2)^2];$$

$$\breve{v} = v + v[k_1(u^2 + v^2) + k_2(u^2 + v^2)^2]$$

where ($\breve{u}$, $\breve{v}$) are the distorted image coordinates, and k_1 and k_2 are two parameters to control the shape of the geometric distortion. The coordinates ($\breve{u}$, $\breve{v}$) then are taken as input to our method.

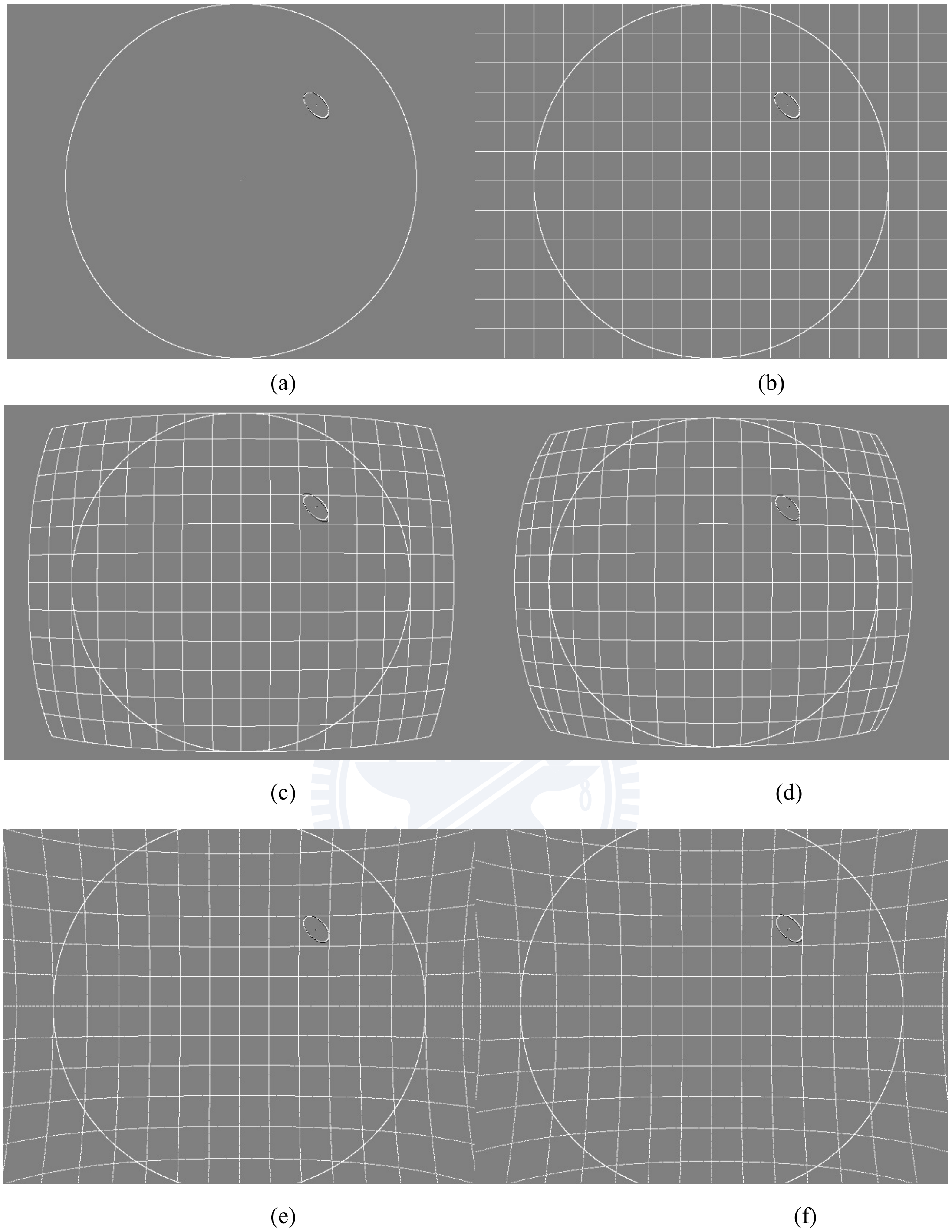
Second, an ellipse detection algorithm [17] was used to detect these landmarks. Third, a rate of successful ellipse detections was computed. And at last, the three steps were repeated to compute the rates of successful ellipse detections for different kinds of omni-cameras which were created virtually only by changing the parameters of their hyperboloidal mirrors. More specifically, we changed only the mirror parameter *b* in (1) while maintaining the value of *c* invariant (= 20 in our simulations) since the position and the focus of the cameras need not be changed. The

smallest value of *b* was taken to be 146mm since otherwise the landmark image will exceed the range of the image plane. And the largest value of *b* was taken to be 158mm because otherwise the rate of successful ellipse detection will drop dramatically owing to the shrinking of the landmark shape to a point in the image. The results are shown in Table 2.1. Examples of successful detections of ellipses in both original and distorted images are shown in Fig. 2.7.

The simulation shows that the lens distortion has only a minor influence on the landmark detection. As indicated by Table 2.1, given shapes with proper sizes and reasonable image distortions like the examples shown in Fig. 2.7, almost all landmarks can be successfully detected from the omni-images, even when the images are taken with cameras with different hyperboloidal mirrors.

(B) Experimental Results of Real Images for Vehicle Location Estimation

The proposed method was also applied to two sets of real images. As shown in Fig. 2.8, both sets of images were taken with an upward-looking omni-camera mounted on an expandable vertical shaft on an autonomous vehicle. The relative positions of the camera, ceiling, and landmark are illustrated in Fig. 2.1. When



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

Fig. 2.7 Examples of successful detection of elliptical shapes. In (a)-(f), the elliptical shapes of simulated landmark images are marked by white pixels. Black pixels are approximate ellipse points computed by the ellipse detection algorithm [17]. In (b)-(f), both horizontal and vertical grid lines are added in order to indicate the level of geometric distortion in each simulated image.

performing location estimation, the shaft was lifted to raise the camera to the height of 1.4m. Then, a landmark with a radius of 40cm attached on the ceiling with a height of

2.9m from the ground was imaged and its shape in the image detected. All the images in the first set were taken from an identical orientation but with different distances from the camera to the landmark for the purpose of investigating the relation between the landmark distance and the estimation error. The results of this experiment are shown in Table 2.2. All the images in the second set like those shown in Fig. 2.9 were taken from an identical orientation and at an identical location but with different camera heights for the purpose of investigating the relation between the depth variation with respect to the estimation. For measurement of the estimation precision, we define a distance error ratio and an orientation error as follows:

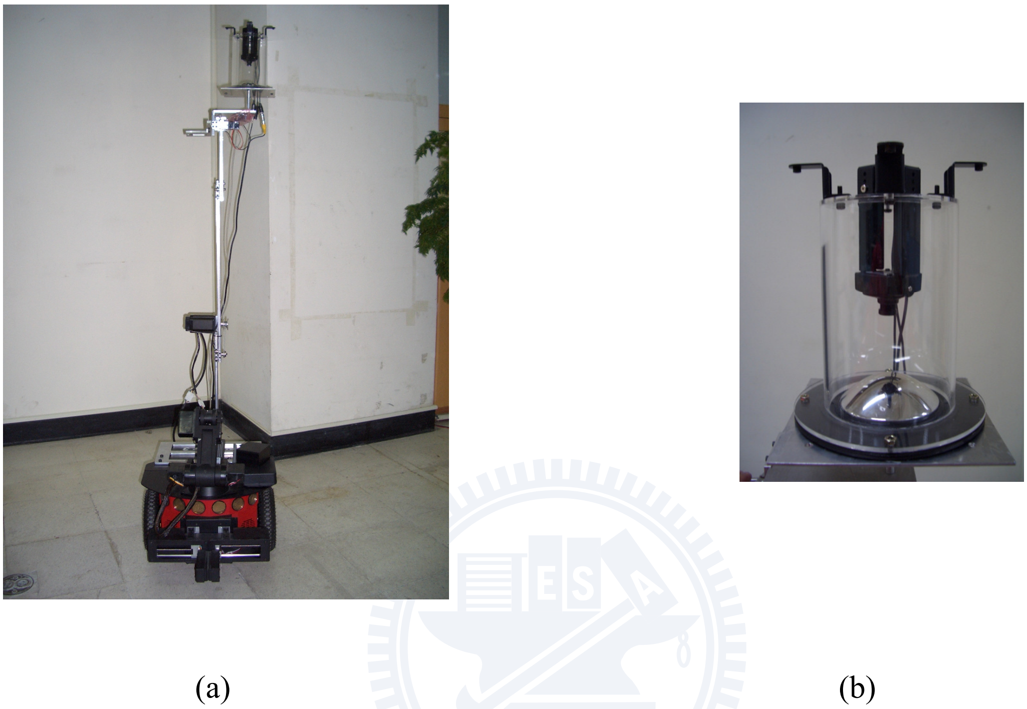


Fig 2.8 All the experimental images were taken by an autonomous land vehicle equipped with an upward-looking omni-camera. (a): the autonomous land vehicle. (b): a close look of the camera on the vehicle.

distance error ratio = (real distance − estimated distance)/(real distance);

orientation error = real orientation − estimated orientation.

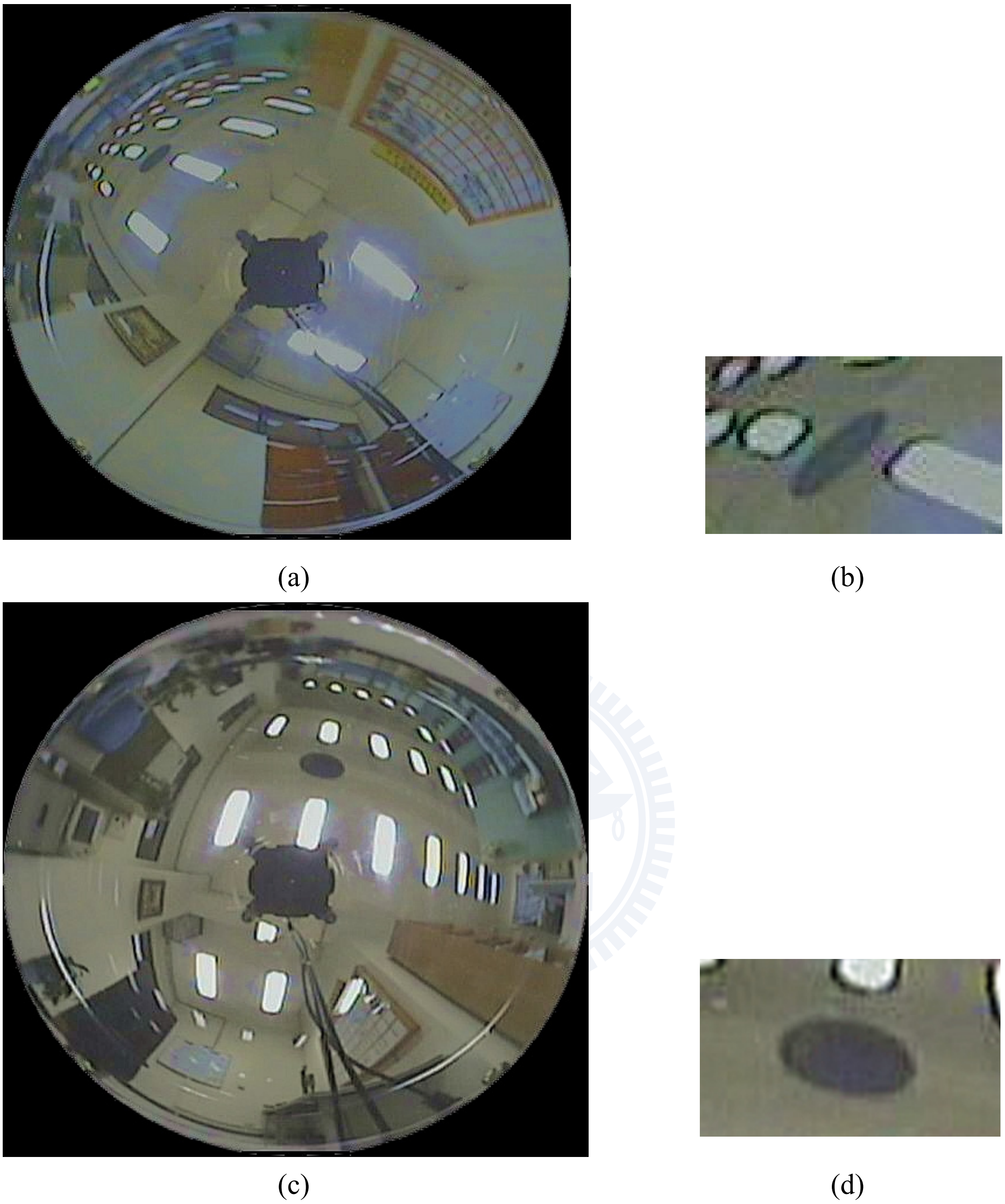


Fig. 2.9 Two example images acquired by the camera equipped on the vehicle. (a) and (c): images acquired at different positions. (b) and (d): the enlarged images of the landmarks in (a) and (c), respectively.

The distance error ratios and orientation errors computed from the two sets of images are shown in Tables 2.2 and 2.3, respectively. The resolution of image is 640 by 480.

The number of times of experiments is five, that is, each distance or orientation value in Table 2.2 and Table 2.3 is the average of the data measured from the same orientation for five times. From the tables, we see that all the distance error ratios are smaller than 5% and that all the orientation errors are smaller than 2^{o}. Such estimation precisions are sufficient for the purpose of autonomous vehicle guidance and navigation purposes.

2.5 Identification and arrangement of the proposed landmarks in real applications

With a consideration of the integrity of a landmark-based navigation system, solving the identification and arrangement problems of the proposed landmarks is inevitable for our system. For example, how does the robot identify some landmark at which it is looking at a certain moment if it was taught the arrangement of landmarks in advance? In this section, we will discuss the identification and arrangement of the proposed landmarks in real applications.

We solve the identification problem of individual landmarks by creating a color coding system. Each landmark is colored by two colors forming two concentric circles with each color being with one of nine pre-selected colors, as illustrated in Fig. 2.10. Furthermore, the nine colors are equally spaced in the hue attribute of the HSI color system, in order to make them more separable and reduce the influence of the lighting condition. Consequently, there are 81 different color combinations for use as the identification numbers to differentiate the landmarks in the map.

Moreover, since the location accuracy using our method becomes worse than

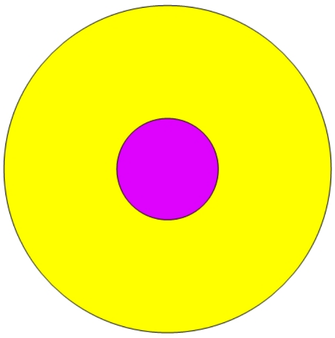


Fig. 2.10 Coloring scheme for identification of multiple circular shapes.

4% at distances of about 5m, we suggest repeating the circular landmark on the ceiling at intervals of 10m. Also, the repetitions should be done in two dimensions, forming a pattern of square grids with 10m sides. The diameter of the circular landmark is 80cm for a ceiling at a height of 2.9m. If the ceiling height is changed, then the size of the circle may be changed proportionally.

2.6 Concluding Remarks

In this study, a new approach to location estimation of an autonomous vehicle for navigation guidance in an indoor environment using a circular-shaped landmark on the ceiling by omni-directional vision techniques has been proposed. It is proved both by theoretical derivations and experimental results that the distorted landmark shape appearing in the image may be well approximated, based on the application of Taylor series expansion, by an ellipse. After the elliptical shape is extracted by an ellipse extraction algorithm, a location estimation approach is proposed by using the ellipse parameters for autonomous vehicle applications.

The proposed location estimation approach with a unique ability of allowing dynamic camera height changes increases the flexibility of creating an occlusion-free camera view in unknown complex environments with obstacles. Additionally, the exact hyperbolic mirror shape of the omni-camera need not be considered which

Table 2.1 Rates of successful detections of landmark shapes taken by cameras with different shapes of hyperboloidal reflection mirror (***c*** = 20mm).

***b* (unit mm)**	**Detection rate**
146	100%
147	100%
148	100%
149	100%
150	100%
151	100%
152	100%
153	100%
154	100%
155	100%
156	100%
157	100%
158	95%
159	<66%

Table 2.2 Error ratios in location estimations with landmarks located in the same directions but at different distances.

estimated distance*: *d	***real distance*: *d***	***estimated orientation***	***real orientation***	***distance error ratio* (%)**	***orientation error* (degree)**
0.985	1.000	-80.8	-81.5	1.5	0.7
1.474	1.500	-81.4	-81.5	1.7	0.2
1.959	2.000	-83.0	-81.5	2.0	1.5
2.440	2.500	-83.0	-81.5	2.4	1.5
2.925	3.000	-82.8	-81.5	2.5	1.3
3.384	3.500	-81.7	-81.5	3.3	0.2
3.862	4.000	-80.9	-81.5	3.5	0.6
4.359	4.500	-81.6	-81.5	3.1	0.1
4.859	5.000	-81.1	-81.5	2.8	0.4
5.281	5.500	-81.6	-81.5	4.0	0.1
5.801	6.000	-81.8	-81.5	3.3	0.3
6.212	6.500	-81.1	-81.5	4.4	0.4

Table 2.3 Errors in location estimations with the landmarks located in the same directions and at the same distances but with different camera heights.

height of landmark from camera **(m)**	***estimated distance*: *d***	***real distance*: *d***	***estimated orientation***	***real orientation***	***distance error ratio*** **(%)**	***orientation error*** **(degree)**
2.5	3.076	3.000	-43.5	-42	2.5%	-1.5
2.25	3.032	3.000	-44.2	-42	1.1%	-2.2
2	3.074	3.000	-43.802	-42	2.5%	-1.802
1.75	3.039	3.000	-41.845	-42	1.3%	0.155
1.5	3.051	3.000	-40.569	-42	1.7%	1.431
1.25	3.097	3.000	-42.466	-42	3.2%	-0.466
1	2.942	3.000	-41.388	-42	-1.9%	0.612

reduces the estimation error caused by the use of the imprecise shape parameters of the mirror obtained from calibration. The computation of the proposed approach is analytic, thus speeding up the estimation process and so reducing the navigation cycle time. Both simulated and real images were tested and good experimental results prove the effectiveness of the proposed approach.

Chapter 3
A Systematic Approach to Indoor Vision-Based Robot Localization Using Corner Features in Omni Images

3.1 Idea of Proposed Method

Robot localization, also termed *robot location*, using landmarks has been studied intensively in the past two decades. Many kinds of vision-based techniques have been proposed, in which the location of a robot is represented by both its position with respect to a reference point and its orientation with respect to a reference shape included in the landmark, respectively, as we do in this study.

In this study, a systematic approach to indoor omni-vision based robot localization using house corners as landmarks is proposed. We derive simple analytic solutions to speed up omni-image analysis and robot location computation. An omni-image of the ceiling of a house, which includes many corners, is shown in Fig. 3.1(a). Specifically, we use an omni-camera with a hyperboloidal mirror for image acquisition, and derive systematically the solutions for various structures of house corners appearing images. A complete structure of a commonly-seen house corner is usually formed by a ceiling and two walls which are all planes, as shown in Fig. 3.1(b). An edge line appears at the intersection of every two planes, resulting in three lines forming a shape of "Y" and intersecting at the *corner point*, as shown in Fig. 3.1(c). Two of the lines are horizontal, and one vertical. Here, by *vertical* and *horizontal*, we mean respectively "perpendicular" and "parallel" to the floor surface on which the robot navigates.

Due to ill lighting conditions, spatial occlusions, or inappropriate image-taking directions or positions, a corner structure in an acquired image may appear to be *incomplete*, e.g., with only one edge line and the corner point, or even with just a partial edge line in the extreme case. Robot guidance and navigation will be much more flexible if incomplete corner structures are still used for robot localization. Therefore, in this study we conduct a systematic analysis of the adequacy of every possible partial corner structure for robot localization (including computations of the robot position and orientation), and propose accordingly an appropriate robot location computation technique for each case. Incomplete house corner structures as appearing in images include the following cases, where it is assumed that the omni-camera is equipped to look upward and that the height of the ceiling is known in advance.

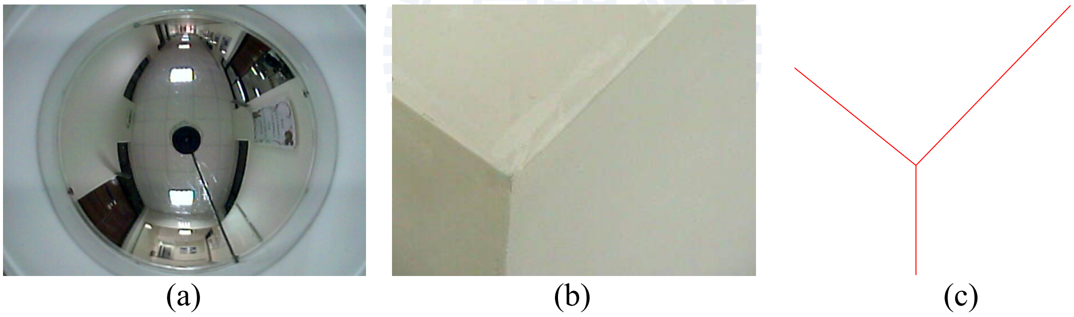


Fig. 3.1 A house corner with a Y-shaped structure. (a) An omni-image of a corridor ceiling with corners. (b) An image part of a corner. (c) Illustration of Y-shape of the corner.

(1) *A single horizontal line with no endpoint* --- this case happens when the robot takes images in a long corridor or in a large room space, with only part of an edge line between two house corners appearing in the image. Neither of the two corner points is within the circular omni-image scope; or they may be within the scope but very close to the circular scope border where the image resolution is low so that they are difficult to be extracted, or, even when extracted, are no good for

precise robot location computation. See Fig. 3.2(a) for an illustration.

(2) *A single vertical line with no endpoint* --- this case happens when the ceiling is too high so that the projection of a corner point, which is one endpoint of a vertical line, falls into the scope of the “black hole” in the central part of the omni-image. The corner point so does not appear in the image and cannot be extracted. Note that the other endpoint, the foot of the vertical line, is also invisible because the camera is “looking upward” as mentioned previously. See Fig. 3.2(b) for an illustration.

(3) *Two horizontal lines intersecting at a corner point* --- this case includes a “V”-shaped structure with the vertical line of the original Y-structure missing or difficult to extract from the image, possibly due to bad lighting on the lower part of the corner. See Fig. 3.2(c) for an illustration.

(4) *A horizontal line and a vertical one intersecting at a corner point* --- this case comes from missing of either of the two horizontal lines, possibly due to bad lighting, directional occlusion, or improper image-taking locations. See Fig. 3.2(d) for an illustration.

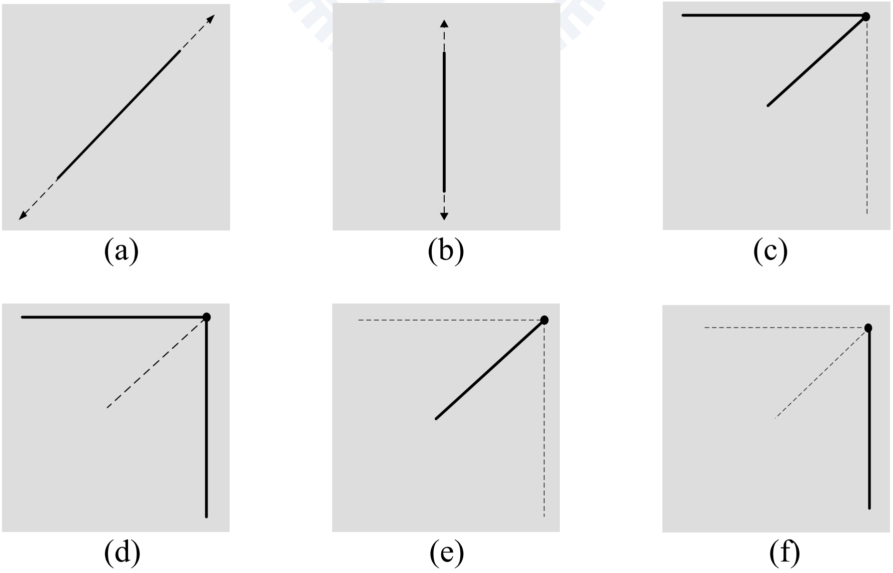
(5) *A horizontal line with a corner point as an endpoint* --- only one of the two horizontal lines with the corner point is left in this case, with the other horizontal line and the vertical one both missing, possibly due to lighting or view occlusion. See Fig. 3.2(e) for an illustration.

(6) *A vertical line with a corner point as an endpoint* --- the two horizontal lines in a corner are missing in this case, possibly due to ill lighting, leaving the vertical line with the corner point. See Fig. 3.2(f) for an illustration.

As found in this study, all the above cases of house corner structures can be used to derive both the position and orientation of a robot, except Cases (2) and (6). In

Case (2), only the relative orientation of the robot can be derived; the robot position is undecidable. And in Case (6), though the absolute robot position can be derived, again only the relative robot orientation is derivable. Here, by *relative orientation*, we mean the direction angle of a reference point with respect to a coordinate system on the robot, instead of that of the robot with respect to a reference line mentioned previously.

The six cases, as identified, are all *significant* in the sense that a *distinct* solution is required to solve the problem of each of them, as done in this study. It is noted that a *complete* house corner is not included as a case in the above. The reason is that it contains more information than needed for computing the robot location, and whenever it is available in an image, we can take just part of its structure to locate the robot according to one of the four cases of (1), and (3) through (5), or more precisely, compute the solutions of all the four cases and average them as the final result of the robot localization process.


(a) (b) (c)

(d) (e) (f)

Fig. 3.2 Six types of partial house corner structures. (a) A horizontal line with no endpoint. (b) A vertical line with no endpoint. (c) Two horizontal lines intersecting at a point. (d) A horizontal line and a vertical one intersecting at a point. (e) A horizontal line with an endpoint. (f) A vertical line with an endpoint.

More details and the merits of the proposed approach are described in the following.

(1) A *simple* quadratic equation for describing the conic-section projection of a space line on the image plane is derived, leading to the possibility of designing a simple algorithm for extracting conic sections out of images using simple low-dimensional Hough transform techniques, in contrast with some complicated conic-section fitting methods adopted in [32][33].

(2) A *systematic* investigation of all possible indoor corner structures, i.e., the above-mentioned six cases, for use in robot localization is conducted, and corresponding robot location computation and image analysis techniques are proposed for flexible applications of robot localization in different lighting, occlusion, and imaging posture conditions.

(3) Formulas for computing robot positions and orientations for the six cases are skillfully derived by appropriate uses of direction vectors of space lines and normals of space planes, all resulting in *analytic* forms appropriate for fast computation and real-time robot guidance and navigation.

(4) The proposed techniques are all based on the use of *single-view* images taken by hyperboloidal omni-cameras, enabling faster robot localization in contrast with some approaches using multiple views [1][20].

In the remainder of this chapter, in Section 3.2 we derive equations for computing the positions and orientations of points and lines. Then, in Section 3.3 we propose techniques for using each of the above-mentioned six cases of partial house corner structures to compute the locations of a robot. Some experimental results are shown in Section 3.4, followed by conclusions in Section 3.5.

3.2 Properties of Projections of Space Points and Lines on Omni-Images

In the proposed approach to robot localization using house corners, first an omni-image of a house corner is acquired, and the projection of the corner in the image is extracted. Then, the projection is fitted into points, lines, and/or curves, and the equations describing them are extracted. Finally, the coefficients of the equations are used to compute the robot location according to some formulas.

In this section, based on some optics and geometry of the omni-camera, first some equations for describing the projections of space points and lines on omni-images will be derived, in a novel way such that the resulting equations become *simple* for designing uncomplicated algorithms to extract line and curve projections easily, as well as *analytic* for fast computation of the robot location. Then, algorithms for extracting two types of projections of space lines, namely, *conic section* and *radial line*, using the Hough transform technique are proposed. Proposed environments for computation of the robot location using the coefficients of the derived analytic equations will be described in the next section.

3.2.1. Derivation of Equation of Space-Point Projection on Omni-image

Two coordinate systems, namely, the *camera coordinate system* and the *image coordinate system*, are involved in the omni-camera used in this study, which includes a traditional perspective camera and a hyperboloidal-shaped mirror, as depicted in Fig. 3.3. The *camera coordinates*, denoted as (*X*, *Y*. *Z*), are used to specify the location of each space point in the real world and the *image coordinates*, denoted as (*u*, *v*), to specify that of the corresponding image point. The perspective camera and the mirror are assumed to be properly aligned such that the omni-camera becomes *single-viewpointed* and the optical axis of the perspective camera coincides with the *mirror axis*, defined to be the line going through the mirror surface center and perpendicular to the mirror base plane. The middle point between the perspective camera's lens center O_l and the mirror's focus point O_m is taken to be the origin O_a of the camera coordinate system. Then, the hyperboloidal mirror shape may be described by (2.1) where a and b are two parameters; and O_m is located at $(0, 0, -c)$ and O_l at $(0, 0, +c)$ in the camera coordinate system where $c = \sqrt{a^2 + b^2}$.

According to [34][35], the relation between the camera coordinates (X, Y. Z) of a space point P and the image coordinates (u, v) of its corresponding projection point p in the image may be described by

$$\tan\alpha = \frac{(b^2 + c^2)\sin\beta - 2bc}{(b^2 - c^2)\cos\beta}; \tag{3.1}$$

$$\cos\beta = \frac{r}{\sqrt{r^2 + f^2}}; \tag{3.2}$$

$$\sin\beta = \frac{f}{\sqrt{r^2 + f^2}}; \tag{3.3}$$

$$\tan\alpha = \frac{Z - c}{\sqrt{X^2 + Y^2}}, \tag{3.4}$$

where $r = \sqrt{u^2 + v^2}$ and f is the camera's focal length. We assume that a, b, c, and f are known in advance. Also, according to the *rotational invariance* property of the omni-camera [36], we have

$$\cos\theta = \frac{X}{\sqrt{X^2 + Y^2}}; \tag{3.5}$$

$$\sin\theta = \frac{Y}{\sqrt{X^2 + Y^2}}, \tag{3.6}$$

$$\cos\theta = \frac{u}{\sqrt{u^2+v^2}}; \tag{3.7}$$

$$\sin\theta = \frac{v}{\sqrt{u^2+v^2}}, \tag{3.8}$$

where θ is both the angle of space point P with respect to the X-axis, and that of image point p in the image coordinate system with respect to the u-axis. The above equations may be used to derive the relation between (u, v) and (X, Y, Z) as by Eqs. (2.2).

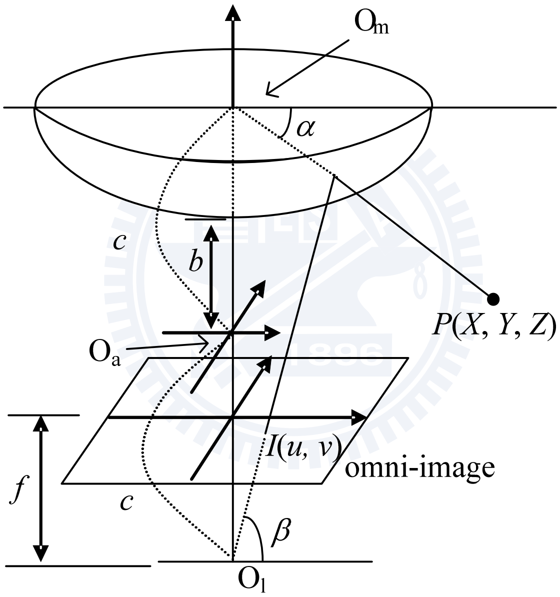


Fig. 3.3 Camera and image coordinate systems.

3.2.2. Derivation of General Equation of A Space Line Projection on Omni-image

As shown in Fig. 3.4, given a space line L with an end point P_0 with camera coordinates (X_0, Y_0, Z_0), any point P on L with camera coordinates (X, Y, Z) and point P_0 together form a vector $V_0 = (X - X_0, Y - Y_0, Z - Z_0)$. On the other hand, let the direction vector of L be denoted as $V_L = (d_X, d_Y, d_Z)$. Then, since V_0 and V_L are parallel, we get the equality $V_0 = \lambda V_L$, or equivalently,

$$(X, Y, Z) = (X_0 + \lambda d_X, Y_0 + \lambda d_Y, Z_0 + \lambda d_Z) \tag{3.9}$$

where λ is a parameter. Also, let S be the space plane going through line L and the mirror base center $\mathrm{O_m}$ at camera coordinates $(0, 0, c)$, and let $N_S = (l, m, n)$ be the normal of S. Then, any point P' at camera coordinates (X, Y, Z) on S and point $\mathrm{O_m}$ together form a vector $V_\mathrm{m} = (X - 0, Y - 0, Z - c) = (X, Y, Z - c)$ which is perpendicular to N_S so that the inner product of V_m and N becomes zero, leading to the following equality:

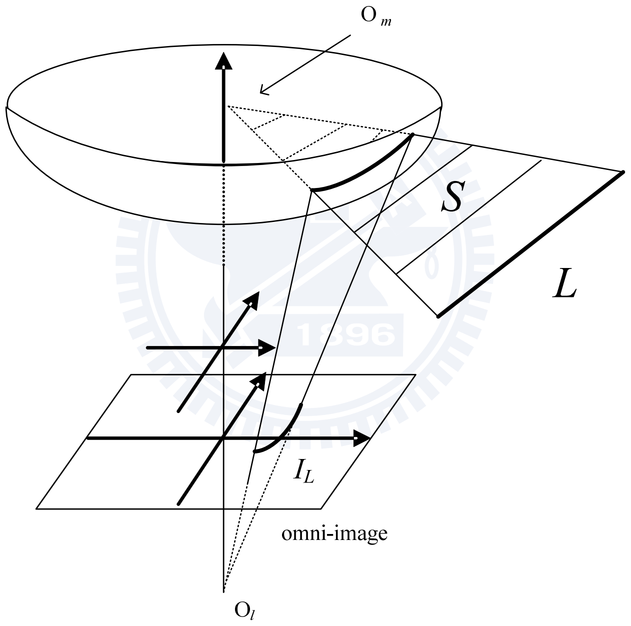


Fig. 3.4 Illustration of a space line projected on to the image plane.

$$lX + mY + n(Z - c) = 0, \tag{3.10}$$

or equivalently,

$$Z - c = -\frac{lX + mY}{n}. \tag{3.11}$$

Now we want to derive the equation of the projection of space line *L* on the image, which expresses the relation between the camera coordinates (*X*, *Y*, *Z*) of a space point *P*′ on *L* and the image coordinates (*u*, *v*) of the image point *p*′ corresponding to *P*′. Note that *P*′ is also on plane *S*. Combining (3.4) through (3.8) and (3.11), we get

$$\tan\alpha = \frac{Z-c}{\sqrt{X^2+Y^2}} = -\frac{l\dfrac{X}{\sqrt{X^2+Y^2}}+m\dfrac{Y}{\sqrt{X^2+Y^2}}}{n}$$

$$= -\frac{l\dfrac{u}{\sqrt{u^2+v^2}}+m\dfrac{v}{\sqrt{u^2+v^2}}}{n} = -\frac{(\dfrac{lu}{n}+\dfrac{lv}{n})}{\sqrt{u^2+v^2}}. \qquad (3.12)$$

On the other hand, substituting (3.2) and (3.3) into (3.1), we get

$$\tan\alpha = \frac{(b^2+c^2)\dfrac{f}{\sqrt{f^2+r^2}}-2bc}{(b^2-c^2)\dfrac{r}{\sqrt{f^2+r^2}}} = \frac{(b^2+c^2)f-2bc\sqrt{f^2+u^2+v^2}}{(b^2-c^2)\sqrt{u^2+v^2}}. \qquad (3.13)$$

Equating (3.12) and (3.13) leads to

$$\frac{lu}{n}+\frac{mv}{n}+\frac{(b^2+c^2)f}{(b^2-c^2)}=\frac{2bc\sqrt{f^2+u^2+v^2}}{(b^2-c^2)}$$

which may be squared and reduced to get the following desired result:

$$(A^2-D^2)u^2+2ABuv+(B^2-D^2)v^2+2ACu+2BCv+E=0 \qquad (3.14)$$

where

$$A=\frac{l}{n}, \quad B=\frac{m}{n}, \quad C=\frac{(b^2+c^2)f}{(b^2-c^2)}, \quad D=\frac{2bc}{(b^2-c^2)}, \quad E=C^2-D^2f^2. \ (3.15)$$

Multiplying (3.14) by n^2, we can get an alternative form of (3.14) without *A* and *B* as follows:

$$(l^2 - n^2D^2)u^2 + 2lmuv + (m^2 - n^2D^2)v^2 + 2lnCu + 2mnCv + n^2E = 0. \qquad (3.14a)$$

Eq. (3.14) or (3.14a) shows that the projection of a space line on the image is a *conic section curve*. And (3.15) shows that the coefficients of the equation may be described *indirectly* in terms of the parameters of the normal $N_S = (l, m, n)$ of plane S. These coefficients actually are related to the elements of the direction vector $V_L \cdot = (d_X, d_Y, d_Z)$ of L by the equality $N_S \cdot V_L = (l, m, n) \cdot (d_X, d_Y, d_Z) = 0$ because N_S and V_L are perpendicular, or equivalently,

$$ld_X + md_Y + nd_Z = 0. \qquad (3.16)$$

Furthermore, Eq. (3.14), as derived above, has a good property that the unknown parameters l, m, and n are confined to appear in just two variables A and B, as shown by (3.15). This facilitates the extraction of the conic section curve by a simple technique using a 2D Hough transform proposed in this study, which will be presented later in this section.

3.2.3. Derivation of Specific Equation of A Space Vertical Line Projection on Omni-image

When the space line L is vertical, the projection equation described by (3.14a) may be simplified further. Specifically, the direction vector of the vertical line L is $V_L = (d_X, d_Y, d_Z) = (0, 0, 1)$. Therefore, (16) leads to $0 \times l + 0 \times m + 1 \times n = 0$, or equivalently, $n = 0$. Accordingly, (3.14a) becomes $l^2u^2 + 2lmuv + m^2v^2 = 0$, or equivalently, $(lu + mv)^2 = 0$ which describes a line going through the image center at (0, 0) of the form $v = -(l/m)u$. That is, every vertical line in the real-world space becomes a *radial line* in the image space going through the image center, which can be described by

$$v = -Ku, \qquad (3.17)$$

where

$$K = \frac{l}{m}. \tag{3.18}$$

Eq. (3.17) above has only one parameter *K*, the slope of the radial line, and this fact facilitates the extraction of the radial line from the image, as described later in Section 3.2.5.

3.2.4. Detection of Conic-section Projection of A Horizontal Space Line by Hough Transform

The algorithm which we propose to extract the conic-section projection of a horizontal line described by Eq, (3.14) is presented in the following, where the goal is to estimate the two parameters *A* and *B*, the only two unknowns in (3.14).

Algorithm 3.1 Extraction of conic-section projection onf an image by Hough transform.

Input: the conic section projection *L*′ in an image *I* of a horizontal space line *L*.

Output: the values of the two parameters *A* and *B* in Eq, (3.14) which describes *L*′.

Step 1. Extract the points of *L*′ out of *I*, by thresholding or edge detection, to form a new image *I*′.

Step 2. Set up a 2D Hough space with two parameters *A* and *B* and set all cell values to be zero.

Step 3. For each point in *I*′ at coordinates (*u*, *v*) and for each cell at parameters (*A*, *B*), if *u*, *v*, *A*, and *B* satisfy Eq. (3.14), then increment the cell value by one.

Step 4. Detect the peak cell value in the Hough space and take the parameters (*A*, *B*) of the cell with the peak value as output to draw a conic-section curve described by (3.14).

Note that the above conic-section detection process using the Hough transform is simple, much less complicated than those used in the related studies of [32][33], as mentioned previously. This merit comes from the simplicity of the form of (3.14) derived in this study, in which there are only two variables *A* and *B*. In the sequel, whenever a conic section described by (3.14) is mentioned, we assume that the two parameters *A* and *B* in (3.14) have been obtained by Algorithm 3.1 above. This means in turn that the ratios *l*/*n* and *m*/*n* are known because *l*/*n* = *A* and *m*/*n* = *B* according to (3.15).

3.2.5. Detection of Radial-line Projection of A Vertical Space Line

To extract a radial line described by (3.17), which is the projection of a vertical space line, again we can use the Hough transform technique [37] in a very simple way, as described by the following algorithm.

Algorithm 3.2 Extraction of radial line projection on an image by Hough transform.

Input: the radial line projection *L*′ in an image *I* of a vertical space line *L*.

Output: the value of the parameter *K* in Eq. (3.17) which describes *L*′.

Step 1. Extract the points of *L*′ out of *I*, by thresholding or edge detection, to form a new image *I*′.

Step 2. Set up a 1D Hough space with a parameter *K* and set all cell values to be zero.

Step 3. For each point in *I*′ at coordinates (*u*, *v*) and for each cell at parameter *K*, if *u*, *v* and *K* satisfy Eq. (3.17), then increment the cell value by one.

Step 4. Detect the peak cell value in the Hough space and take the parameter *K* of the cell with peak cell as output to draw a radial line described by (3.17).

In the sequel, whenever a radial line described by Eq. (3.17) is mentioned, we assume that the parameter K in (3.17) have been obtained according to Algorithm 3.2 above. This means also that the ratio l/m is known according to (3.18).

3.3 Robot Localization by Partial House Corner Structures

We now describe one by one the robot localization technique we propose for each of the six cases of partial house corner structures mentioned previously. In each case, we have to compute the position and orientation of the robot with respect to a reference point and a reference line, respectively, included in a house corner structure. For convenience, we compute reversely the position of the reference point and the orientation of the reference line with respect to the camera coordinate system on the robot. The formulas we derive in these techniques for such robot location computations are all analytic.

3.3.1. Case (1) Robot Localization Using A Single Horizontal Line with No Endpoint

In this case as shown in Fig. 3.5, we are given a single horizontal line L at a known height h with no endpoint. To compute the robot position, we have to find a reference point, which we propose to be the *minimum-distance point* $P_{\min}$ on L to the origin O_a of the camera coordinate system, as illustrated by Fig. 3.6. And we compute the direction angle θ of L with respect to the X-axis of the camera coordinate system as the desired robot orientation.

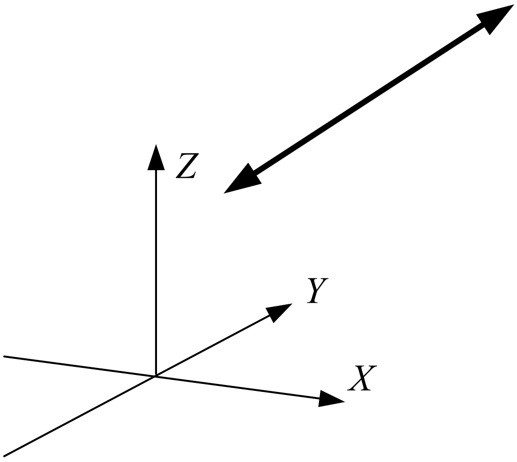


Fig. 3.5 Case (1) – a single horizontal line with no endpoint used for robot localization.

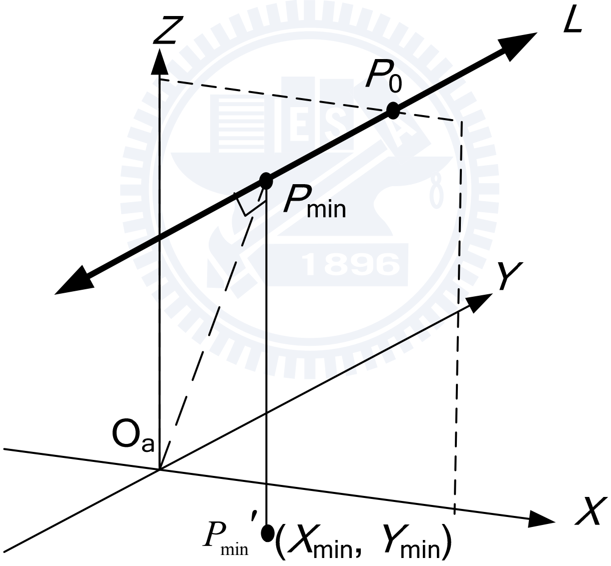


Fig. 3.6 Finding minimum-distance point $P_{\min}$ on horizontal line L for Case (1).

To find $P_{\min}$ on L, since L is parallel to the floor, the value of d_Z of the direction vector $V_L = (d_X, d_Y, d_Z)$ is zero. So, (3.16) may be transformed into

$$\frac{d_X}{m/n} = -\frac{d_Y}{l/n} = s \tag{3.19}$$

where s is a parameter. Combining (3.9) and (3.19), we get a parametric equation for

L as

$$(X, Y, Z) = (X_0 + \frac{m}{n}s\lambda,\ Y_0 - \frac{l}{n}s\lambda,\ Z_0 + d_Z\lambda) \qquad (3.20)$$

where (X_0, Y_0, Z_0) specify the coordinates of another point P_0 on *L* yet to be determined. Since *L* is at the height *h*, Z =0 for all points on *L*, and so (3.20) above may be rewritten as

$$(X, Y, Z) = (X_0 + \frac{m}{n}t,\ Y_0 - \frac{l}{n}t,\ h) \qquad (3.21)$$

where the new parameter $t = s\lambda$. The distance from the origin O_a at (0, 0, 0) to an arbitrary point on *L* at (X, Y, Z) therefore is

$$d(t) = ||(X, Y, Z) - (0, 0, 0)|| = \sqrt{(X_0 + \frac{m}{n}t)^2 + (Y_0 - \frac{l}{n}t)^2 + h^2}\ .$$

And the minimum distance $d_{\min}$ from O_a to *L* may be obtained by taking the derivative of the square of $d(t)$ above, setting it to be zero, and solving the resulting equation, which is of the following form, for the corresponding parameter $t_{\min}$:

$$2(X_0 + \frac{m}{n}t)(\frac{m}{n}) + 2(Y_0 - \frac{l}{n}t)(-\frac{l}{n}) = 0\,. \qquad (3.21)$$

The solution $t_{\min}$ of (3.21) is

$$t_{\min} = \frac{-\frac{m}{n}X_0 + \frac{l}{n}Y_0}{(\frac{m}{n})^2 + (\frac{l}{n})^2} \qquad (3.22)$$

in which the two parameters X_0 and Y_0 are yet to be solved. To determine X_0 and Y_0, we may regard *L* as an infinite-length line going through the space plane *Q* described geometrically by the equation $X_0 = 0$, and take the intersection point of *L* and *Q* as the point P_0 with coordinates (X_0, Y_0, Z_0) where $X_0 = 0$ and $Z_0 = h$. Accordingly, (3.11) may be reduced to be

$$Y_0 = \frac{-n(h-c)}{m}, \tag{3.23}$$

so that (3.22) becomes

$$t_{\min} = \frac{\dfrac{-l(h-c)}{m}}{(\dfrac{m}{n})^2 + (\dfrac{l}{n})^2}. \tag{3.24}$$

Therefore, the coordinates ($X_{\min}$, $Y_{\min}$, $Z_{\min}$) of the minimum-distance point $P_{\min}$ on L to $\mathrm{O_a}$ may now be calculated from (3.21) and (3.24) to be

$$\begin{aligned}(X_{\min}, Y_{\min}, Z_{\min}) &= (\frac{m}{n} \times \frac{\dfrac{-l(h-c)}{m}}{(\dfrac{m}{n})^2 + (\dfrac{l}{n})^2},\ \frac{-n(h-c)}{m} - \frac{l}{n} \times \frac{\dfrac{-l(h-c)}{m}}{(\dfrac{m}{n})^2 + (\dfrac{l}{n})^2}, h) \\ &= (\frac{-\dfrac{l}{n}(h-c)}{(\dfrac{m}{n})^2 + (\dfrac{l}{n})^2},\ \frac{-\dfrac{m}{n}(h-c)}{(\dfrac{m}{n})^2 + (\dfrac{l}{n})^2}, h) = (\frac{-A(h-c)}{A^2+B^2},\ \frac{-B(h-c)}{A^2+B^2}, h)\end{aligned} \tag{3.25}$$

where $A = l/n$, $B = m/n$ are known values obtained from Algorithm 3.1, as mentioned previously. Therefore, the desired robot position with respect to projection $P_{\min}'$ of the reference point $P_{\min}$ on the floor is specified by

$$(X_{\min}, Y_{\min}) = (\frac{-A(h-c)}{A^2+B^2},\ \frac{-B(h-c)}{A^2+B^2}). \tag{3.26}$$

As to the robot orientation which is the angle θ of line L with respect to the X-axis of the camera coordinate system, it can be computed from

$$\cos\theta = \frac{V_L \cdot V_X}{\| V_L \| \times \| V_X \|} \tag{3.27}$$

where V_L and V_X are the direction vectors of L and the X-axis, respectively. We know

$V_X = (1, 0, 0)$. And $V_L = (d_X, d_Y, d_Z)$ may be computed to be the unit vector from point P_0 to point $P_{\min}$ in the following way:

$$V_L = \frac{(X_{\min} - X_0,\ Y_{\min} - Y_0,\ Z_{\min} - Z_0)}{\|(X_{\min} - X_0,\ Y_{\min} - Y_0,\ Z_{\min} - Z_0)\|}$$

where

$$(X_0, Y_0, Z_0) = (0,\ \frac{-n(h-c)}{m}, h) = (0,\ \frac{-(h-c)}{B}, h);$$

$$(X_{\min}, Y_{\min}, Z_{\min}) = (\frac{-A(h-c)}{A^2+B^2},\ \frac{-B(h-c)}{A^2+B^2}, h).$$

Consequently, because $Z_{\min} - Z_0 = h - h = 0$, we get finally the desired θ as

$$\theta = \cos^{-1}\left(\frac{V_L \cdot V_X}{\|V_L\| \times \|V_X\|}\right) = \cos^{-1}\left(\frac{X_{\min} - X_0}{\sqrt{(X_{\min} - X_0)^2 + (Y_{\min} - Y_0)^2}}\right)$$

which may be reduced to be

$$\theta = \cos^{-1}\left(\frac{-B}{\sqrt{A^2+B^2}}\right). \tag{3.28}$$

3.3.2. Case (2) Robot Localization Using A Single Vertical Line with No Endpoint

In this case as shown in Fig. 3.7, we are given a vertical line L with no endpoint. As proven previously, the projection of L on the image, according to (3.17) and (3.18), is a radial line L' described by $v = -Ku$, where the slope K may be extracted by Algorithm 3.2. Because the plane S going through L and O_a (the origin of the camera coordinate system) is vertical to the floor, all lines lying in S are projected on the image plane identically to form the radial line L'. Therefore, L' cannot be used to compute unambiguously the position of the robot. Furthermore, because the projection of L on the floor is just a point P and not a line, L' is also insufficient for determining the orientation of the robot.

However, the slope information K is not totally useless; instead, it can be utilized to find the *relative* orientation θ of the vertical line L with respect to the robot (actually with respect to the X-axis of the camera coordinate system). Specifically, let the projection point P' have camera coordinates (X_0, Y_0) which, we know, are identical to the camera coordinates (X, Y) of any point P on L. Then, according to the rotational invariance property described by (3.5), (3.6), (3.7), and (3.8), we have

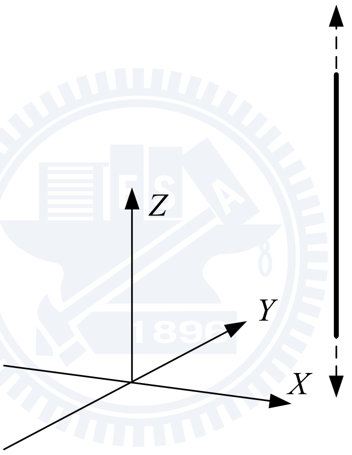


Fig. 3.7 A vertical line with no endpoint used in robot localization.

$$\cos\theta = \frac{X}{\sqrt{X^2+Y^2}} = \frac{X_0}{\sqrt{X_0^{\ 2}+Y_0^{\ 2}}} = \frac{u}{\sqrt{u^2+v^2}}\ ;$$

$$\sin\theta = \frac{Y}{\sqrt{X^2+Y^2}} = \frac{Y_0}{\sqrt{X_0^{\ 2}+Y_0^{\ 2}}} = \frac{v}{\sqrt{u^2+v^2}}\ ,$$

where (u, v) are the coordinates of the image point p corresponding to P. Though (X_0, Y_0) cannot be solved from the above equations, their ratio is related to the value of θ by $\tan\theta = Y_0/X_0 = v/u = -K$ according to (3.17) and (3.18). Therefore, the desired robot orientation is

$$\theta = \tan^{-1}(-K) \tag{3.29}$$

where K is obtained by Algorithm 3.2.

3.3.3. Case (3) Two Horizontal Lines Intersecting at A Corner Point

In this case as shown in Fig. 3.8, we have two horizontal lines L_1 and L_2 both at a known height h and intersecting at a corner point P_0. Assume that the camera coordinates of P_0 are (X_0, Y, Z_0) where $Z_0 = h$. Let the plane formed by L_1 and O_m be denoted as S_1, and that by L_2 and O_m as S_2. If the normals of S_1 and S_2 are (l_1, m_1, n_1) and (l_2, m_2, n_2), respectively, then by (10) we have $l_iX + m_iY + n_\mathrm{i}(Z - c) = 0$, $i = 1, 2$, or equivalently,

$$\frac{l_i}{n_i}X + \frac{m_i}{n_i}Y + Z = c\,,\ i = 1,\ 2, \tag{3.30}$$

By replacing l_i/n_i and m_i/n_i respectively with A_i and B_i which are obtained by Algorithm 3.1, and knowing that L_1 and L_2 intersect at P_0 with coordinates (X_0, Y_0, Z_0) where $Z_0 = h$, we may transform Eqs. (3.30) into the following simultaneous equations:

$$A_iX_0 + B_iY_0 + h = c,\ i = 1,\ 2 \tag{3.31}$$

which may then be solved to get X_0 and Y_0 as the robot position as follows:

$$X_0 = \frac{(h-c)(B_1 - B_2)}{(A_1B_2 - A_2B_1)}; \tag{3.32}$$

$$Y_0 = \frac{(h-c)(A_1 - A_2)}{(A_1B_2 - A_2B_1)}. \tag{3.33}$$

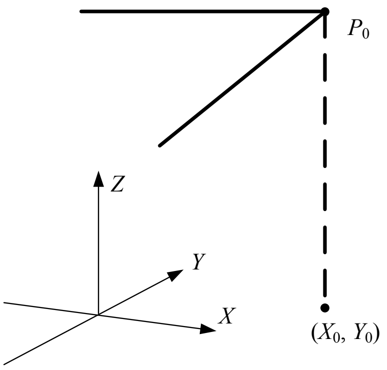


Fig. 3.8. Two horizontal lines intersecting at a corner point used in robot localization.

As to the robot orientation, we take it to be the angle θ_1 of L_1 with respect to the X-axis of the camera coordinate system, which may be computed according to the process we proposed for Case (1). According to (3.28), the result is as follows:

$$\theta_1 = \cos^{-1}\left(\frac{-B_1}{\sqrt{A_1^{\ 2} + B_1^{\ 2}}}\right). \tag{3.34}$$

Notice the simplicity and analyticity of the above solutions for robot position and orientation, which we have mentioned before.

In practical applications of this case, we first have to extract the conic-section projections of the two horizontal lines from an image of them by Algorithm 3.1. But we do not have to compute the intersection point using the two extracted conic-section curves. Instead, by the above-proposed process, we can compute *directly* the position and orientation of the robot according to Eqs. (3.32), (3.33), and (3.34) using the parameters A_1, B_1, A_2, and B_2 extracted by Algorithm 3.1 as well. This is a merit of the above-proposed technique for this case.

3.3.4. Case (4) A Horizontal Line And A Vertical One Intersecting at A Corner Point

In this case as shown in Fig. 3.9, we have a horizontal line L_1 and a vertical one L_2 at a known height h and intersecting at a corner point P_0. Assume that the camera coordinates of P_0 are (X_0, Y, Z_0) where $Z_0 = h$. Let the plane formed by L_1 and $\mathrm{O_m}$ be denoted as S_1, and that by L_2 and $\mathrm{O_m}$ as S_2. If the normals of S_1 and S_2 respectively are (l_1, m_1, n_1) and (l_2, m_2, n_2), then since P_0 is on L_1, similarly to the derivation of (3.31) we can get

$$A_1X_0 + B_1Y_0 + h = c \tag{3.35}$$

where $A_1 = l_1/n_1$ and $B_1 = m_1/n_1$ are obtained by Algorithm 3.1 from an image I of L_1 and L_2.

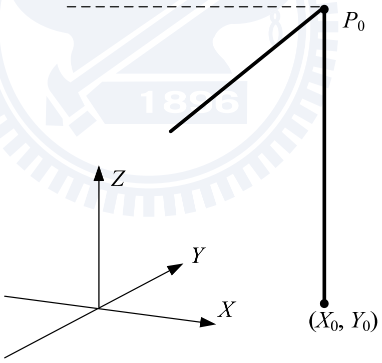


Fig. 3.8 A horizontal line and a vertical one intersecting at a corner point used in robot localization.

On the other hand, since L_2 is a vertical line, according to the analysis of Section 2.3, we have $n_2 = 0$ so that (3.10) becomes $l_2X + m_2Y = 0$, or equivalently, $Y = (-l_2/m_2)X$. Using the known value $K_1 = l_2/m_2$ obtained by Algorithm 3.2, we may rewrite the equality $Y = (-l_2/m_2)X$ just derived to be $Y = -KX$. Accordingly, since P_0 with coordinates (X_0, Y_0, Z_0) is also on the vertical line, we get

$$Y_0 = -KX_0. \tag{3.36}$$

Now, (3.35) and (3.36) may be solved to get the solution (X_0, Y_0) to specify the robot position as follows:

$$X_0 = -\frac{(h-c)}{(A_1 - KB_1)}; \tag{3.37}$$

$$Y_0 = \frac{(h-c)K}{(A_1 - KB_1)}. \tag{3.38}$$

As to the robot orientation, we take it to be the angle θ_1 of L_1 with respect to the X-axis of the camera coordinate system, which may be computed according to the process we proposed for Case (1). According to (3.28), the result is as follows:

$$\theta_1 = \cos^{-1}\left(\frac{-B_1}{\sqrt{A_1^{\,2} + B_1^{\,2}}}\right). \tag{3.39}$$

Notice again the simplicity and analyticity of the above solutions for robot localization.

In practical applications of this case, we have to extract the conic-section and radial-line projections of the horizontal and vertical lines from an image of the two lines by Algorithms 3.1 and 3.2, respectively. Similar to the last case, we do not have to compute the intersection point using the two extracted projections in the image. Instead, we compute *directly* the position and orientation of the robot according to Eqs. (3.37), (3.38), and (3.39) using the parameters A_1, B_1, and K extracted from the image by the algorithms as well. This is again a merit of the above-proposed technique for this case.

3.3.5. Case (5): A Horizontal Line with A Corner Point as An Endpoint

As shown in Fig. 3.9, in this case we are given a horizontal line L with a corner point P_0 as an endpoint at a known height h. Assume that the camera coordinates of P_0 are (X_0, Y_0, Z_0) where $Z_0 = h$. Using Algorithm 3.1, the conic-section projection L' of L on the input image I can be extracted to get the two parameters A and B as in Case (1).

However, a difference here is that the image point p_0 corresponding to P_0 is *observable*, which may be reached by tracing L' to an end of it. Let the image coordinates of p_0 so obtained be (u_0, v_0). Then, we may derive the robot position (X_0, Y_0) in terms of the values of u_0, v_0, h, A, and B, as done in the following.

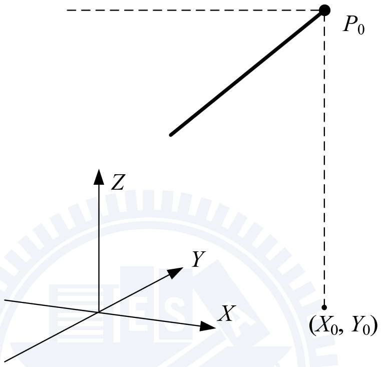


Fig. 3.9 A horizontal line with a corner point as an endpoint used in robot localization.

First, being derived for a horizontal line in Case (4), Eq. (3.35) is valid as well here, but in the form of

$$AX_0 + BY_0 + h = c. \tag{3.40}$$

Next, by substituting (X_0, Y_0, Z_0) and (u_0, v_0) into (3.5) ~ (3.8) which describe the rotational invariance property of the omni-camera, the result may be combined to get

$$u_0/v_0 = X_0/Y_0. \tag{3.41}$$

Finally, the two Eqs. (3.40) and (3.41) above may be solved to obtain the following desired solution for (X_0, Y_0):

$$X_0 = -\frac{u_0(h-c)}{(Au_0 + Bv_0)}; \tag{3.42}$$

$$Y_0 = -\frac{v_0(h-c)}{(Au_0 + Bv_0)}. \tag{3.43}$$

As to the robot orientation θ, it can be obtained similarly by the process proposed for Case (1) using L as a reference line. The result is described by (3.28) which is repeated here:

$$\theta = \cos^{-1}\left(\frac{-B}{\sqrt{A^2 + B^2}}\right). \tag{3.44}$$

3.3.6. Case (6): A Vertical Line with A Corner Point as An Endpoint

As shown in Fig. 3.10, in this case we are given a vertical line L with a corner point P_0 as an endpoint at a known height h. Let the camera coordinates of P_0 be (X_0, Y_0, Z_0) where $Z_0 = h$. We want to compute (X_0, Y_0) as the robot position using an image I of L and P_0. Using Algorithm 3.2, the radial-line projection L' of L on I can be extracted to obtain its slope K, as in Case (2). However, different from Case (2) here is that the image point p_0 corresponding to P_0 is *observable*, which may be reached by tracing L' outward from the image center. Let the image coordinates of p_0 be (u_0, v_0). The hint that p_0 is the projection of P_0 at the height of h gives us the possibility to derive the values of X_0 and Y_0 as the robot position, as done in the following.

First, substituting (X_0, Y_0, Z_0) and (u_0, v_0) into (3.4) and (3.13) with $Z_0 = h$, we get

$$\frac{h-c}{\sqrt{{X_0}^2 + {Y_0}^2}} = \frac{(b^2+c^2)f - 2bc\sqrt{f^2 + {u_0}^2 + {v_0}^2}}{(b^2-c^2)\sqrt{{u_0}^2 + {v_0}^2}} = \frac{C - D\sqrt{f^2 + {u_0}^2 + {v_0}^2}}{\sqrt{{u_0}^2 + {v_0}^2}} \tag{3.44}$$

where

$$C = \frac{(b^2+c^2)f}{(b^2-c^2)},\ \ D = \frac{2bc}{(b^2-c^2)}.$$

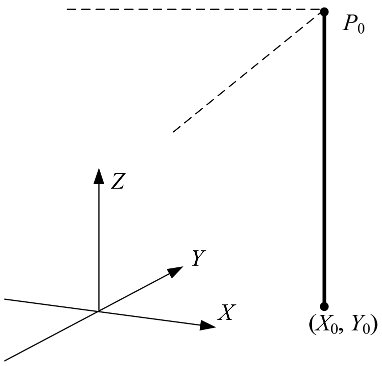


Fig. 3.10 A vertical line with a corner point as an endpoint used in robot localization.

Next, by (3.41) which is also true here, we get

$$Y_0 = (v_0/u_0)X_0. \tag{3.45}$$

Eqs. (3.44) and (3.45) above may be solved to get the following solution for the robot position:

$$X_0 = \frac{u_0(h-c)}{C - D\sqrt{f^2 + u_0^{\ 2} + v_0^{\ 2}}}; \tag{3.46}$$

$$Y_0 = \frac{v_0(h-c)}{C - D\sqrt{f^2 + u_0^{\ 2} + v_0^{\ 2}}}. \tag{3.47}$$

Similar to Case (2), only the relative orientation θ of the robot can be obtained because no line can be used as a reference line. The result is described by (3.29) which is repeated below, where K is the slope of the radial line projection L' of L:

$$\theta = \tan^{-1}(-K). \tag{3.48}$$

3.4 Experimental Results

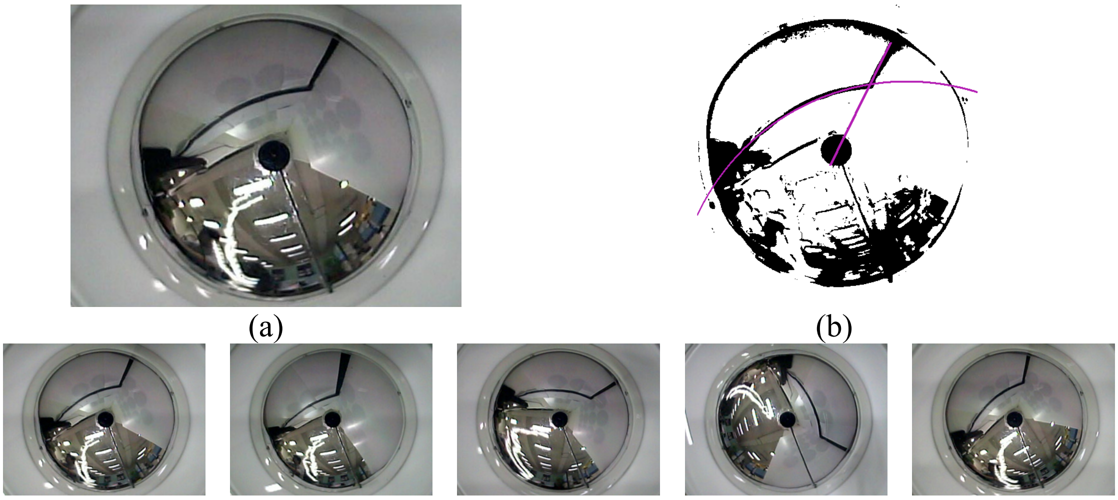
A series of experiments have been conducted, and some of the results are reported here. The omni-camera with a hyperboloidal mirror we used is MapCam MRC530N produced by EeRise Co. The images were taken from the laboratory where this study was conducted. Each image was processed by Algorithms 3.1 and

3.2. An experimental result of Case (4) is shown in Fig. 3.11, where Fig. 3.11(a) is the original omni-image of an indoor scene with a large L-shape corner formed by a horizontal line and a vertical one on a wall, and Fig. 3.11(b) is the extraction result of the conic-section and radial-line projections respectively of the horizontal and vertical lines using Algorithms 3.1 and 3.2. The extracted radial line goes through the image center as expected, and the extracted conic-section curve is part of an ellipse, with their equations specified by (3.14) and (3.17), respectively, being computed to be:

$$v = -1.3537u;$$
$$-16.314u^2 + 0.0318uv - 13.786v^2 - 41.360u - 13152.511v + 998036 = 0,$$

where the values of the extracted A, B, and K, and the camera parameters are shown in Table 3.2. These results show that the two Hough transforms proposed in Algorithms 3.1 and 3.2 are effective and yield good projection extraction results.

To test the accuracy of the robot localization results computed by the formulas derived in this study, we took five more images of the scene as shown in Fig. 3.10(c), processed them to extract the projections, and used the results to compute the robot position and orientation for each of the six images according to Eqs. (3.37), (3.38), and (3.39). Comparisons of these results against the real position and orientation data measured manually are shown in Table 3.1, from which we can see that the average error ratios for the computed X and Y coordinates of the robot position are both smaller than 3%, and the average error of the computed orientations is smaller than 0.6 degree, all good enough for robot guidance and navigation. The errors might come from imprecise camera parameter calibration results, inaccurate manual measures of the real position and orientation data, low resolutions of acquired omni-images, etc.



(a)

(b)

(c)

Fig. 3.11 Extraction of horizontal and vertical lines by proposed Hough transform techniques for robot localization of Case (4). (a) An omni-image with a corner consisting of a horizontal line and a vertical one. (b) Extracted conic-section and radial-line projections shown as purple curves. (c) Five more images used in experiments.

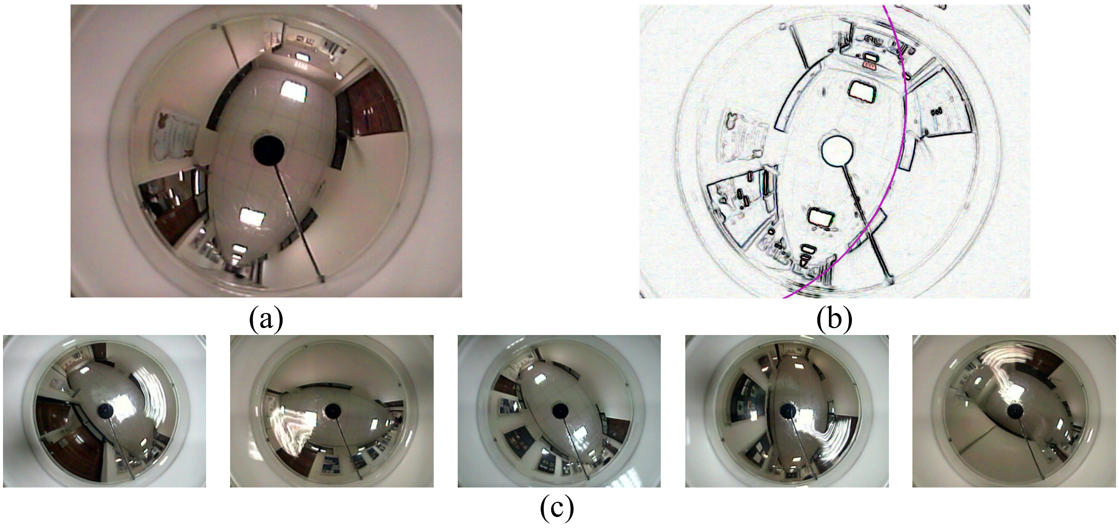
Table 3.1 Robot Location computation results for Case (4).

Real X	Computed X	Error ratio of X	Real Y	Computed Y	Error ratio of Y	Real θ (degree)	Computed θ (degree)	Angle error (degree)
15.0	15.5	4.0%	20.5	21.0	2.9%	0.0	0.4	0.4
31.0	29.6	4.3%	28.5	28.7	0.9%	0.0	0.5	0.5
35.0	34.1	2.4%	28.5	29.3	2.8%	0.0	0.5	0.5
44.1	43.2	2.0%	9.5	9.6	1.3%	27.0	27.4	0.4
29.7	29.8	0.4%	-33.9	-34.3	1.0%	88.0	89.5	1.5
26.4	26.7	1.1%	36.5	37.1	1.6%	-15.0	-15.5	0.5
average error		2.4%	*average error*		1.8%	*average error*		0.6

Table 3.2 Parameters involved or computed in extraction of projections in Fig. 3.10.

B	c	f	C	D	E	A	B	K
3.6	4.6	994	-4136.01	-4.0390	988036	0.01	1.59	1.3537

Another experimental result we show here is for Case (1) where only a single horizontal line is used. Six omni-images, as shown in Fig. 3.11, were taken from a long corridor ceiling with unconnected horizontal edge line segments, fitting the description of the partial corner structure of Case (1). The robot localization results using Eqs. (3.26) and (3.28) are shown in Table 3.3, from which we see again that the accuracy of the position and orientation computation results are sufficient for general robot navigation applications.


(a)

(b)

(c)

Fig. 3.12 Images used in experiment of robot localization of Case (1). (a) An image of a corridor with horizontal line segments as ceiling edges. (b) Extracted conic-section projection shown as purple curve. (c) Five more images used in experiments.

Table 3.3 Robot location computation results for Case (1).

Real X	Computed X	Error ratio of X	Real Y	Computed Y	Error ratio of Y	Real θ (degree)	Computed θ (degree)	Angle error (degree)
101.5	102.8	1.3%	101.5	102.8	1.3%	45.0	45.0	0.0
184.3	190.8	3.6%	-89.9	-92.3	2.7%	116.0	115.8	0.2
41.5	42.6	2.7%	-74.8	-77.2	3.2%	151.0	151.1	0.1
137.7	136.2	1.1%	-42.1	-41.7	1.0%	107.0	107.0	0.0
198.9	208.9	5.0%	49.6	50.7	2.1%	76.0	76.4	0.4
143.9	145.9	1.4%	-143.9	-140.3	2.5%	135.0	133.9	1.1
average error		2.5%	*average error*		2.1%	*average error*		0.3

3.5 Concluding Remarks

A systematic investigation of vision-based robot localization techniques by omni-cameras using indoor house corners as landmarks has been conducted. An omni-camera with a hyperboloidal mirror was used for omni-image acquisition. Equations for describing conic-section projections of space lines on omni-images were carefully derived by the use of space lines' direction vectors and related space planes' normals to be simple and analytic. The simple forms of the equations with only one or two unknown parameters to be determined makes possible the design of uncomplicated low-dimensional Hough transform-based algorithms for effective extraction of conic-section or radial-line projections out of omni-images. The analyticity of the equations makes possible fast computations of robot positions and orientations, facilitating real-time robot guidance and navigation applications.

Identification of all possible partial structures of house corners appearing images has also been conducted, resulting in six significant types: (1) a horizontal line with no endpoint, (2) a vertical line with no endpoint, (3) two horizontal lines intersecting at a point, (4) a horizontal line and a vertical one intersecting at a point, (5) a horizontal line with an endpoint, and (6) a vertical line with an endpoint. Such variations of corner structures in images may come from ill lighting conditions, spatial occlusions, or inappropriate image-taking directions or positions. A distinct robot localization technique has been proposed for each partial corner type, including a process for curve or line projection extraction as well as one for analytic robot localization computation. Experimental results show that the robot position computation results have average position error ratios smaller than 2.5%, and orientation errors smaller than 0.6 degree, which are accurate enough for robot guidance and navigation applications. Further researches may be directed to uses of more complicated house structures for robot localization, like multiple parallel lines, combinations of multiple structures, etc.

Chapter 4
An Omni-vision Based Self-localization Method for Automatic Helicopter Landing on Standard Helipads

4.1 Idea of Proposed Method

Automatic helicopter landing is one of the desired techniques for safe helicopter aviation or unmanned air vehicle (UAV) applications. Among the many possible ways for solving this problem, the vision-based approach is a promising one, which uses cameras to acquire environment images and applies appropriate image analysis techniques to conduct helicopter self-localization and flight guidance during the landing process. See Fig. 1.7 for an illustration.

Many studies conducted for unmanned helicopter flying in the past used parallel line information on a standard helipad as a hint for helicopter self-localization with respect to the circled H-shape on the helipad. See Fig. 4.1 for an illustration. However, these studies did not fully use the shape information on the helipad. Also, many of the methods used the traditional projective camera for visual sensing, which has a fixed FOV, compared with that of the omni-camera. An example of *omni-images* of a simulated helipad acquired with an omni-camera is shown in Fig. 4.2. It is advantageous to use the omni-camera in automatic helicopter landing to enlarge the viewing scope and consequently speed up the automatic landing process. Furthermore, most of the above methods do not investigate the full details of the automatic landing process.

In this study, we investigate omni-vision based self-localization techniques using the circled H-shape on a helipad as the landmark. The features on the landmark, including point, line, and circle, and their mutual relation properties are used *systematically* for helicopter location estimation. The helicopter landing process is divided into *three stages*, namely, *approaching* the helipad, *aligning* with the H-shape boundaries, and *docking* on the helipad center. Analytic equations are derived for faster computation of the helicopter location, including its the position, orientation, and height, for automatic flight guidance in each stage. More details and merits of the proposed method are described in the following.

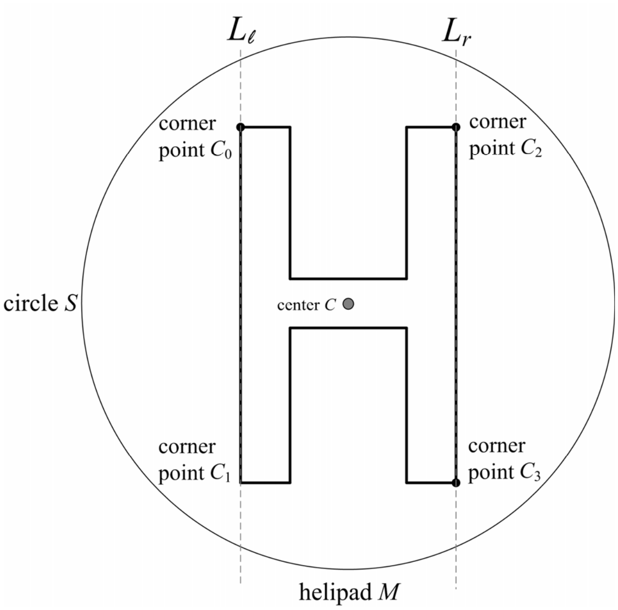


Fig. 4.1 Detail of a circled H-shape on a standard helipad.

1. A *simple* conic-section equation for describing a line in an omni-image is derived, leading to the possibility of extracting the boundary lines of the H-shape using a simple low-dimensional Hough transform technique.
2. Skillful combinations of possible features are carried out for *effective* helicopter self-localization in each stage of the proposed automatic landing process.
3. Formulas for computing the helicopter location parameters for the three landing

stages are derived, all resulting in *analytic* forms for fast computations in real applications.

4. The proposed techniques are all based on the use of *single-view* omni-images taken with omni-cameras which enables faster helicopter localization in general, in contrast with approaches using multiple views taken by projective cameras.

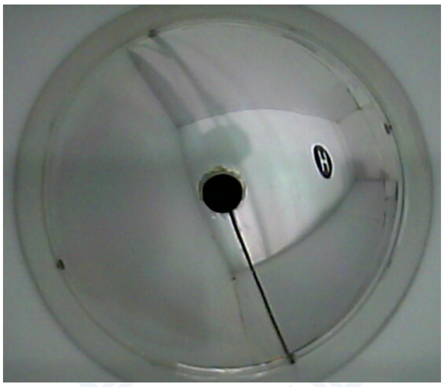


Fig. 4.2 An omni-image of a simulated helipad.

In the remainder of this chapter, in Section 4.2 we introduce the idea of the proposed helicopter self-localization method. Then, in Section 4.3 we present the techniques for estimations of the helicopter position, orientation, and height in the proposed three-stage helicopter landing process. Some experimental results are presented in Section 4.4 to show the feasibility of the proposed method, followed by conclusions in Section 4.5.

4.2 Idea of Three-stage Helicopter Self-localization Method

The proposed three-stage automatic helicopter landing process is described as follows, each stage utilizing certain geometric information contained in single omni-images of the circled H-shape to estimate the real-world location information (including the position, orientation, or/and height) of the helicopter.

Stage 1. (*Approaching*) maneuver the helicopter to approach the helipad.

Stage 2. (*Aligning*) maneuver the helicopter to fly between, and then in alignment with, the two outmost boundary lines of the H-shape.

Stage 3. (*Docking*) maneuver the helicopter to land on the center C of the H-shape.

More details in each stage are described in the following algorithm. For mentioned notations in the algorithm, see Fig. 4.1.

Algorithm 4.1 Three-stage process of automatic helicopter landing.

Stage 1. Approaching the helipad.

Step 1. Take an omni-image I_1 of the helipad M.

Step 2. Find the circular shape S in I_1 by approximating it as an ellipse S' by the Hough transform.

Step 3. Compute the orientation θ_1 and distance d_1 of the helicopter with respect to the center C of the H-shape using the information of the radius of S and the parameters of S'.

Step 4. Maneuver the helicopter to approach M using the information of θ_1 and d_1.

Step 5. Repeat Steps 1 through 4 until the helicopter is at a pre-defined distance to M.

Stage 2. Aligning with the H-shape boundaries.

Step 6. Take an image I_2 of M.

Step 7. Find the leftmost and rightmost boundary lines L_ℓ and L_r of the H-shape in I_2 by approximating them as conic sections L_ℓ' and L_r' using the Hough transform.

Step 8. Compute the helicopter height h over the helipad using the information of the known distance d_m between L_ℓ and L_r.

Step 9. Compute the distances d_ℓ and d_r of the helicopter with respect to L_ℓ and L_r, respectively, as well as the orientation θ_2 of the helicopter with respect to L_ℓ, using the information of L_ℓ', L_r', and h.

Step 10. Maneuver the helicopter to fly between L_ℓ and L_r and align its flying direction to be parallel to L_ℓ using the information of d_ℓ, d_r, andθ_2.

Step 11. Repeat Steps 6 through 10 until θ_2 approaches zero (i.e., until it is equal to a pre-selected small value).

Stage 3. Docking on the helipad center.

Step 12. Take an image I_3 of M.

Step 13. Perform Steps 6 through 8 using I_3 to detect L_ℓ' and L_r' in I_3 and obtain the helicopter height h.

Step 14. Trace L_ℓ' and L_r' found in the last step to find the four outmost corners C_1' through C_4' of the H-shape in I_3.

Step 15. Locate the four outmost corners C_1 through C_4 of the H-shape in the real-world space using the information of L_ℓ', L_r', and C_1' through C_4'.

Step 16. Locate the center C of the H-shape using the location information of C_1 through C_4.

Step 17. Compute the real-world distance d_C of the helicopter to C using the information of the location of C and the helicopter height h.

Step 18. Maneuver the helicopter toward C and lower its height gradually until both d_C and h becomes zero, i.e., until the plane is touchdown.

4.3 Proposed Self-localization Techniques for Automatic Helicopter Landing

The respective self-localization techniques proposed for the previously-described three stages of the helicopter landing process are described in this section.

4.3.1. Proposed Techniques for the Approaching Stage

The major steps in the first stage for approaching the helipad include the following tasks.

A. (*Circular shape detection*) finding the circular shape S in an omni-image by approximating it as an ellipse S' using the Hough transform (Step 2 of Algorithm 4.1).

B. (*Helicopter orientation and distance computation*) computing the orientation θ_1 and distance d_1 of the helicopter with respect to the center C of the H-shape using S' and the known radius value R of S (Step 3 of Algorithm 4.1).

The details of these tasks are described in the following.

A. Circular shape detection

Two coordinate systems, namely, the *camera coordinate system* and the *image coordinate system*, are set up on the omni-camera system used in this study, which includes a traditional perspective camera and a hyperboloidal-shaped mirror, as depicted in Fig. 3.3. The *camera coordinates*, denoted as (X, Y, Z), are used to specify the position of each space point in the real world, and the *image coordinates*, denoted as (u, v), are used to specify the position of the corresponding image point in the image. The perspective camera and the mirror are assumed to be properly aligned so that the omni-camera system becomes *single-viewpointed* and that the optical axis of the perspective camera coincides with the *mirror axis*. Here, the mirror axis is defined as the line going through the mirror surface center and perpendicular to the mirror base plane. The middle point between the perspective camera's lens center $\mathrm{O_l}$ and the mirror's focus point $\mathrm{O_m}$ is taken to define the origin $\mathrm{O_a}$ of the camera coordinate system. Accordingly, the hyperboloidal mirror shape may be described by (2.1) where *a* and *b* are two parameters, and $\mathrm{O_m}$ is located at (0, 0, +*c*) and $\mathrm{O_l}$ at (0, 0, −*c*) in the camera coordinate system where $c = \sqrt{a^2 + b^2}$.

According to [34][35], the relation between the camera coordinates (*X*, *Y*, *Z*) of a space point *P* and the image coordinates (*u*, *v*) of its corresponding image point *p* may be described by Eqs. (3.1), (3.2), (3.3), and (3.4), where α and β are two space angles as illustrated in Fig. 3.3, where $r = \sqrt{u^2 + v^2}$ and *f* is the camera's focal length. We assume that *a*, *b*, *c*, and *f* are known in advance, which may be obtained by proper camera calibration processes. Also, according to the *rotational invariance* property of the omni-camera system [36], we have (3.5), (3.6), (3.7), and (3.8) where θ is the angle of space point *P* with respect to the *X*-axis and is *also* that of image point *p* with respective to the *u*-axis. The above equations may be used to derive two equalities describing direct relation between (*u*, *v*) and (*X*, *Y*, *Z*) as Eqs. (2.1).

When the helicopter is not too close to the helipad, according to Wu and Tsai [49], the circle *S* enclosing the H-shape, though irregular in shape when appearing in a given omni-image *I*, may be approximated well as an ellipse *S*′, and the length of the major axis of the extracted ellipse gives a hint for computing the distance of the helicopter to the H-shape. Accordingly, in this study we use the Hough transform technique to detect *S*′ in *I* after *I* is processed into an edge-point image using the Sobel edge detection operator. Examples of such ellipse detection results can be seen in Figs.

4.9, 4.11, and 4.13 in Section 4.4. Specifically, the coordinates (u_C, v_C) of the center C' of the detected ellipse S' as well as the values U and V of the major and minor axes of S' may be obtained as the output of the Hough transform process for use in helicopter orientation and distance computations described next.

B. Helicopter orientation and distance computation

While referring to Fig. 4.3 which is a top view of the omni-camera system and the circular shape S, let the known radius of S be denoted by R_S and let the camera coordinates of the center C of S be denoted by (X_C, Y_C, Z_C). In the entire automatic landing process, it is assumed that the omni-camera is looking *downward*, so that the normal vector of the helipad is parallel to the optical axis of the omni-camera. We will now derive the distance and orientation of the helicopter to the helipad, or more precisely in a reverse way, the distance d_1 and orientation θ_1 of the center C of the circle with respect to the camera coordinate system on the helicopter. The two parameters d_1 and θ_1 can then be used for maneuvering the helicopter to approach S (Step 4 of Algorithm 4.1).

First, we know that the projection of the center C of S on an image I is just the center C' of the detected ellipse S' whose coordinates (u_C, v_C) can be obtained by the Hough transform process as mentioned previously. Then, according to the rotational invariance property described by (3.5), (3.6), (3.7), and (3.8), we can get the desired parameter θ_1, which is both the angle of S' with respect to the u-axis and the angle of S with respect to the X-axis, to be:

$$\theta_1 = \cos^{-1}\frac{X_C}{\sqrt{{X_C}^2 + {Y_C}^2}} = \cos^{-1}\frac{u_C}{\sqrt{{u_C}^2 + {v_C}^2}}\,; \qquad (4.1)$$

or

$$\theta_1 = \sin^{-1}\frac{Y_C}{\sqrt{{X_C}^2 + {Y_C}^2}} = \sin^{-1}\frac{v_C}{\sqrt{{u_C}^2 + {v_C}^2}}\,. \qquad (4.2)$$

As to the desired parameter d_1, to make the derivation of it easier, we rotate S' by the angle of θ_1, with the rotation result being shown in the image plane depicted in Fig. 4.3. Then, the rotated S' may be described by the following equation:

$$\frac{(u'-u_C')^2}{U^2}+\frac{v'^2}{V^2}=1 \tag{4.3}$$

where the new image coordinates are described by (u', v') and the new ellipse center is located at (u_C', 0). Furthermore, let the two endpoints of the major axis of S' be denoted as p_α and p_β, and their corresponding space points be denoted as P_α and P_β, respectively. Then, by the rotational invariance property again, it is not difficult to figure out that the following *side proportionality* property is truth as illustrated by Fig. 4.3:

$$\frac{\left\|\overline{\mathrm{O_i}C}\right\|}{\left\|\overline{\mathrm{O_i}C'}\right\|}=\frac{\left\|\overline{P_\alpha P_\beta}\right\|}{\left\|\overline{p_\alpha p_\beta}\right\|} \tag{4.4}$$

where $\mathrm{O_i}$ is the origin of the image coordinate system (also the origin of the camera coordinate system seen from the top view), $\left\|\overline{\mathrm{O_i}C}\right\|$ is the desired real-world distance d_1 on the floor from C to the camera, $\left\|\overline{\mathrm{O_i}C'}\right\|$ is the image distance from C' to $\mathrm{O_i}$ which is equal to $\sqrt{u_C{}^2+v_C{}^2}$, $\left\|\overline{P_\alpha P_\beta}\right\|$ is the diameter value $2R_s$ of S, and $\left\|\overline{p_\alpha p_\beta}\right\|$ is the length U of the major axis of S'. Accordingly, the desired value d_1 can be computed finally from (4.4) as:

$$d_1 = \frac{2R_s}{U}\sqrt{u_C{}^2+v_C{}^2}\,. \tag{4.5}$$

4.3.2. Proposed Techniques for the Aligning Stage

The major steps in the second stage for helicopter alignment with the outmost boundary lines of the H-shape include the following three tasks.

A. (*Boundary line detection*) finding the leftmost boundary line L_ℓ of the H-shape in an image by approximating it as a conic section L_ℓ' using the Hough transform, and doing the same to extract the rightmost boundary line L_r' (Step 7 of Algorithm 4.1).

B. (*Helicopter height computation*) computing the height h of the helicopter over the helipad using the information of the extracted L_ℓ', L_r' and the known distance d_m between L_ℓ and L_r in the real world (Step 8 of Algorithm 4.1).

C. (*Helicopter distance and orientation computation*) computing the distances d_ℓ and d_r to L_ℓ and L_r, respectively, as well as the orientation θ_2 of the helicopter with respect to L_ℓ, using the information of h, L_ℓ', and L_r' (Step 9 of Algorithm 4.1).

The details are described respectively in the following.

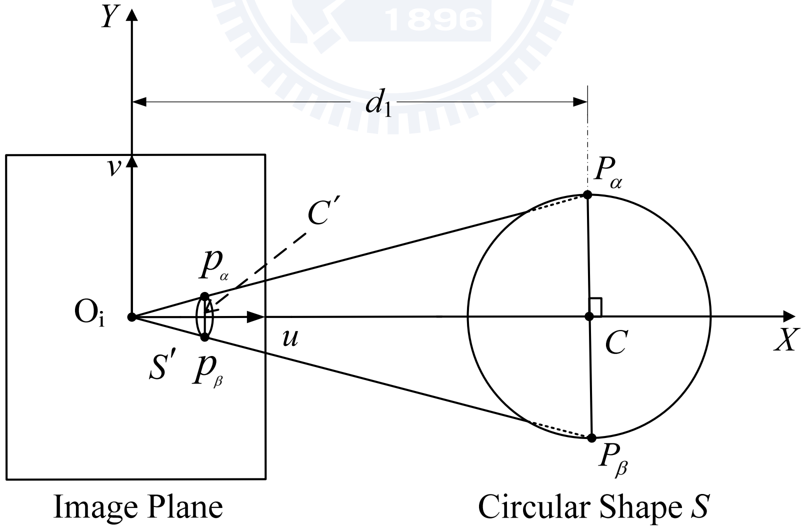


Fig. 4.3 Top view of image plane and circular shape S illustrating side proportionality relation between approximating ellipse S′ and circular shape S.

A. Boundary line detection

Based on some optics and geometry of the omni-camera, detection of an outmost boundary line L (either of L_ℓ or L_r) of the H-shape in an omni-image I is accomplished in this study first by deriving an equation to describe the projection L' of L in I, which is a conic section, as will proved later. The derivation is conducted in a novel way in this study so that the resulting equation becomes *simple* and *analytic*, facilitating design of an uncomplicated Hough transform algorithm to extract L' and computation of the helicopter location in a faster speed for practical applications.

In more detail, as shown in Fig. 4.4, suppose that the boundary line L has a specific point P_0 located at camera coordinates (X_0, Y_0, Z_0), and let P be an arbitrary point on L with camera coordinates (X, Y, Z). Then P and P_0 together form a vector V_0 = ($X - X_0$, $Y - Y_0$, $Z - Z_0$). Also, let the direction vector of L be denoted as V_L = (d_X, d_Y, d_Z). Then, by the fact that V_0 and V_L are parallel, we get the equality $V_0 = \lambda V_L$ which leads to Eq. (3.9) where λ is a parameter.

Also, let Q be the space plane going through both the line L and the mirror base center $\mathrm{O_m}$ which is located at camera coordinates (0, 0, $+c$); N_q be the normal vector of Q with components (l, m, n); and P' be an arbitrary point on Q with camera coordinates (X, Y, Z). Then, P' and $\mathrm{O_m}$ together form a vector V_m = ($X - 0$, $Y - 0$, $Z - c$) = (X, Y, $Z - c$) which is perpendicular to N_q, so that the inner product of V_m and N_q becomes zero, leading to the following equality (3.10) which is equivalent to Eq. (3.11).

Now we want to derive an equation for describing the projection of the space line L on the image, which also expresses the relation between the camera coordinates (X, Y, Z) of a space point P on L and the image coordinates (u, v) of the image point p corresponding to P. Note that P is also on plane Q. Combining (3.4) through (3.8) and (3.11), we get Eq. (3.12). On the other hand, substituting (3.2) and (3.3) into (3.1), we get Eq. (3.13)

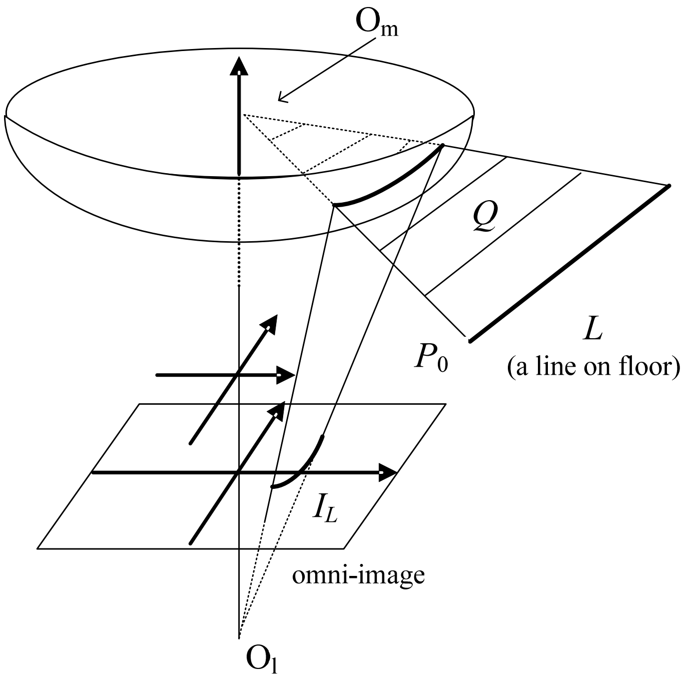


Fig. 4.4 Illustration of a space line projected onto the image plane.

Equating (3.12) and (3.13) leads to

$$\frac{lu}{n}+\frac{mv}{n}+\frac{(b^2+c^2)f}{(b^2-c^2)}=\frac{2bc\sqrt{f^2+u^2+v^2}}{(b^2-c^2)}$$

which may be squared and reduced to become Eqs. (3.14) and (3.15). Multiplying (3.14) by n^2, we get an alternative form of (3.14) without A and B as (3.14a).

Eq. (3.14) or (3.14a) shows that the projection of a space line on an image is a *conic section curve*. And (3.15) shows that the coefficients of the equation may be described *indirectly* in terms of the parameters of the normal $N_q = (l, m, n)$ of plane Q. These coefficients actually are related to the elements of the direction vector $V_L = (d_X, d_Y, d_Z)$ of L by the equality (3.16) because N_q and V_L are *perpendicular*. Furthermore, Eq. (3.14), as derived above, has a good property that the unknown parameters l, m, and n are confined to appear in just two variables A and B, as shown by (3.15). This facilitates the extraction of the conic section curve by a simple technique using a 2D

Hough transform described in Algorithm 3.1.

In the sequel, whenever a conic section described by (3.14) is mentioned, we assume that the two parameter values A and B in (3.14) have been obtained by Algorithm 3.1 above. This means that the ratios l/n and m/n are also known because $l/n = A$ and $m/n = B$ according to (3.15).

B. Helicopter height computation

Suppose that we are dealing with one of the two outmost boundary lines L_ℓ and L_r of the H-shape and let the line be denoted as L. We want to find a reference point, which we propose to be the *minimum-distance point* $P_{\min}$ on L to the origin $\mathrm{O_a}$ of the camera coordinate system, as illustrated by Fig. 4.5. This point $P_{\min}$ will be used later for the purpose of helicopter height computation here.

To find $P_{\min}$ on L, we know that L is on the floor, so L is parallel to the X-Y plane of the camera coordinate system. Consequently, the component d_Z of the direction vector $V_L = (d_X, d_Y, d_Z)$ of L is zero, and Eq. (3.16), which is $ld_X + md_Y + nd_Z = 0$, may be transformed accordingly into Eq. (3.19) where s is a new parameter. Combining (3.9) and (3.19), we get a parametric equation for L as Eq. (3.20) where (X_0, Y_0, Z_0) specify the coordinates of point P_0 on L yet to be determined. Since L is on the landing floor with the helicopter being at a certain height h which we want to find out here, we have $Z = -h$ for all points on L, and so (3.20) above may be rewritten as (3.21) where $t = s\lambda$. The distance from the camera coordinate system origin $\mathrm{O_a}$ at (0, 0, 0) to an arbitrary point on L at (X, Y, Z) therefore is

$$d(t) = \|(X, Y, Z) - (0, 0, 0)\| = \sqrt{(X_0 + \frac{m}{n}t)^2 + (Y_0 - \frac{l}{n}t)^2 + h^2}\ .$$

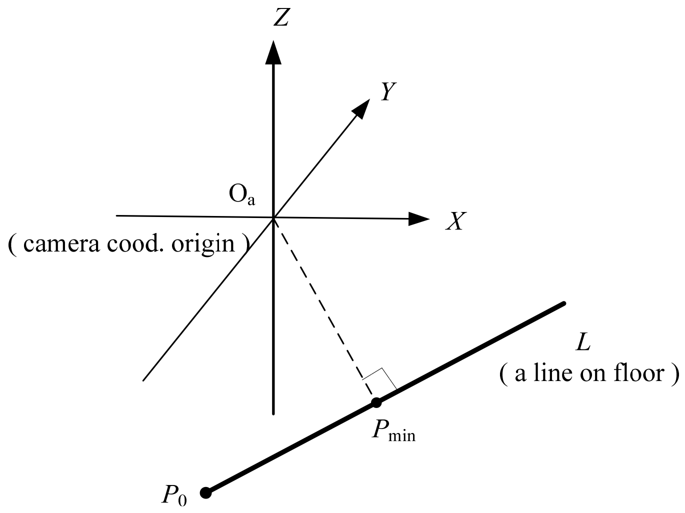


Fig. 4.5 Finding minimum-distance point $P_{\min}$ on a boundary line L.

And the minimum distance $d_{\min}$ from O$_a$ to L may be obtained by taking the derivative of the square of $d(t)$ above, setting it to be zero, and solving the resulting equation (3.21). The solution $t_{\min}$ of (3.21) above is (3.22). To decide the values of X_0 and Y_0 in the above equality, we may regard L as a line with an infinite length and going through the space plane Q_0 described geometrically by the equation $X = 0$. We then take the intersection point of L and Q_0 to be the point P_0, which has coordinates (X_0, Y_0, Z_0) with $X_0 = 0$ and $Z_0 = -h$. Accordingly, (3.11) may be reduced to be (3.23), so that (3.22) becomes (3.24). Therefore, the coordinates ($X_{\min}$, $Y_{\min}$, $Z_{\min}$) of the minimum-distance point $P_{\min}$ on L to O$_a$ may now be calculated from (3.21) and (3.24) to be

$$(X_{\min}, Y_{\min}, Z_{\min}) = \left(\frac{\frac{l}{n}(h+c)}{(\frac{m}{n})^2+(\frac{l}{n})^2}, \frac{\frac{m}{n}(h+c)}{(\frac{m}{n})^2+(\frac{l}{n})^2}, -h\right) = \left(\frac{A(h+c)}{A^2+B^2}, \frac{B(h+c)}{A^2+B^2}, -h\right) \quad (4.6)$$

where $A = l/n$, $B = m/n$ are known values as mentioned previously, of the coefficients of the projection L' of L on the image described by (3.14) and (3.15). Accordingly, we can get from (4.6) the following values for the coordinates of the minimum-distance points $P_{\min}{}^{1}$ and $P_{\min}{}^{2}$ of $L_1 = L_\ell$ and $L_2 = L_r$, respectively, with respect to $\mathrm{O_a}$:

$$(X_{\min}{}^{1}, Y_{\min}{}^{1}, Z_{\min}{}^{1}) = (\frac{A_1(h+c)}{A_1^2+B_1^2}, \frac{B_1(h+c)}{A_1^2+B_1^2}, -h); \tag{4.7}$$

$$(X_{\min}{}^{2}, Y_{\min}{}^{2}, Z_{\min}{}^{2}) = (\frac{A_2(h+c)}{A_2^2+B_2^2}, \frac{B_2(h+c)}{A_2^2+B_2^2}, -h), \tag{4.8}$$

where A_1 and B_1 are the coefficients of the projection $L_1' = L_\ell'$ of $L_1 = L_\ell$ on the image described by (3.14) and (3.15), and A_2 and B_2 are those of the projection $L_2' = L_r'$ of $L_2 = L_r$. By the fact that L_ℓ and L_r are parallel and are separated with a known distance d_m, it is not difficult to figure out that the points $P_{\min}{}^{1}$, $P_{\min}{}^{2}$, and the projection point of $\mathrm{O_a}$ on the landing floor are all on an identical line which is perpendicular both to L_ℓ and to L_r, so that the distance between $P_{\min}{}^{1}$ and $P_{\min}{}^{2}$ on the landing floor is just d_m, leading to the following equality:

$$\left(\frac{A_1(h+c)}{A_1^2+B_1^2}-\frac{A_2(h+c)}{A_2^2+B_2^2}\right)^2+\left(\frac{B_1(h+c)}{A_1^2+B_1^2}-\frac{B_2(h+c)}{A_2^2+B_2^2}\right)^2 = d_\mathrm{m}^2. \tag{4.9}$$

Furthermore, parallelism of L_ℓ and L_r means that their directional vectors are identical, which we assume to be $(d_\mathrm{X}, d_\mathrm{Y}, d_\mathrm{Z})$ with $d_\mathrm{Z} = 0$. So, if (l_1, m_1, n_1) is the normal vector of the space plane including $L_1 = L_\ell$ and $\mathrm{O_m}$ and (l_2, m_2, n_2) is that for the plane including $L_2 = L_r$ and $\mathrm{O_m}$, then according to (3.19) we get

$$\frac{d_Y}{d_X} = -\frac{l_1/n_1}{m_1/n_1} = -\frac{l_2/n_2}{m_2/n_2},$$

which yields

$$\frac{l_1/n_1}{l_2/n_2} = \frac{m_1/n_1}{m_2/n_2} \tag{4.10}$$

or equivalently,

$$\frac{A_1}{A_2} = \frac{B_1}{B_2}$$

because $A_1 = l_1/n_1$, $A_2 = l_2/n_2$, $B_1 = m_1/n_1$, $B_2 = m_2/n_2$ according to (3.15). Define λ as

$$\lambda = \frac{A_1}{A_2} = \frac{B_1}{B_2} \tag{4.11}$$

which is equivalent to

$$A_1 = \lambda A_2,\ B_1 = \lambda B_2. \tag{4.12}$$

Then, substituting (4.12) into (4.9) to eliminate the terms of A_1 and B_1, we get

$$h = \left(\frac{\lambda}{1-\lambda}\right) d_{\mathrm{m}} \sqrt{{A_2}^2 + {B_2}^2} - c. \tag{4.13}$$

And using (4.12) again, we can rewrite the above equation in terms of the known values of A_1, B_1, A_2, and B_2 in two ways to describe finally the desired helicopter height h as

$$h = \frac{A_1 d_{\mathrm{m}}}{A_2 - A_1} \sqrt{{A_2}^2 + {B_2}^2} - c = \frac{B_1 d_{\mathrm{m}}}{B_2 - B_1} \sqrt{{A_2}^2 + {B_2}^2} - c. \tag{4.14}$$

C. Helicopter orientation and height computation

From (4.7), we know that the helicopter position on the landing floor with respect to the reference point $P_{\min}$ on $L_1 = L_\ell$ is described by

$$({X_{\min}}^1,\ {Y_{\min}}^1) = \left(\frac{A_1(h+c)}{{A_1}^2 + {B_1}^2},\ \frac{B_1(h+c)}{{A_1}^2 + {B_1}^2}\right).$$

And the desired distance d_ℓ of the helicopter with respect to $L_1 = L_\ell$ on the landing floor is just

$$d_\ell = \sqrt{(X^1_{\min})^2 + (Y^1_{\min})^2} = \frac{(h+c)}{\sqrt{{A_1}^2 + {B_1}^2}}. \tag{4.15}$$

Similarly, the desired distance of the helicopter to $L_2 = L_r$ on the landing floor is

$$d_r = \sqrt{(X^2_{\min})^2 + (Y^2_{\min})^2} = \frac{(h+c)}{\sqrt{{A_2}^2 + {B_2}^2}}. \tag{4.16}$$

As to the helicopter orientation seen from the top view, it is taken to be the angle θ_2 of line $L = L_\ell$ with respect to the X-axis of the camera coordinate system, which may be computed from the following equality:

$$\cos\theta_2 = \frac{V_L \cdot V_X}{\| V_L \| \times \| V_X \|}$$

where V_L and V_X are the direction vectors of $L = L_\ell$ and the X-axis, respectively. We know that $V_X = (1, 0, 0)$. And $V_L = (d_X, d_Y, d_Z)$ may be computed to be the unit vector from the point P_0 found previously on L (see discussions in Section B above) to the point $P_{\min}$ also on L (see Fig. 4.5) in the following way:

$$V_L = \frac{(X_{\min}^1 - X_0,\ Y_{\min}^1 - Y_0,\ Z_{\min}^1 - Z_0)}{\| (X_{\min}^1 - X_0,\ Y_{\min}^1 - Y_0,\ Z_{\min}^1 - Z_0) \|}$$

where

$$(X_0, Y_0, Z_0) = (0, \frac{n(h+c)}{m}, h) = (0, \frac{(h+c)}{B_2}, -h);$$

$$(X_{\min}{}^2, Y_{\min}{}^2, Z_{\min}{}^2) = (\frac{A_2(h+c)}{A_2{}^2 + B_2{}^2}, \frac{B_2(h+c)}{A_2{}^2 + B_2{}^2}, -h).$$

Consequently, we get the desired θ as

$$\theta_2 = \cos^{-1}\left(\frac{V_L \cdot V_X}{\| V_L \| \times \| V_X \|}\right) = \cos^{-1}\left(\frac{X_{\min}^1 - X_0}{\sqrt{(X_{\min}^1 - X_0)^2 + (Y_{\min}^1 - Y_0)^2}}\right)$$

which, after some reduction, becomes

$$\theta_2 = \cos^{-1}\left(\frac{B}{\sqrt{A^2 + B^2}}\right). \qquad (4.17)$$

4.3.3. Proposed Techniques for the Docking Stage

The steps in the third stage for helicopter docking include three major tasks, as described in the following.

A. (Detection of corner points) finding the four outmost corners of the H-shape in an image to locate their corresponding space points in the real-world space. (Steps 14 and 15 of Algorithm 4.1).

B. (Helicopter distance computation) locating the center of the H-shape as the geometric center of the four corner points, and computing the distance of the helicopter to the H-shape center (Steps 16 and 17 of Algorithm 4.1).

The details are described respectively in the following.

A. Detection of corner points

Given an image *I*, suppose that we have already detected in it a boundary line L' of the H-shape using the Hough transform (Algorithm 3.1) as described previously, and have traced L' to obtain one of its endpoint, p_0, in the image, which is just one of four corners of the H-shape in the image. The real-world versions L and P_0 of L' and p_0, respectively, may be illustrated again by Fig. 4.5. Assume that the helicopter height has been computed already to be h (Step 13 of Algorithm 4.1), and that the image coordinates of p_0 are (u_0, v_0) and the camera coordinates of P_0 are (X_0, Y_0, Z_0), where $Z_0 = -h$. With L' already detected by Algorithm 3.1 as a conic section described by (3.14) and (3.15) with the two parameters A and B, we want to derive the real-world location (X_0, Y_0) of the corner point P_0 on the landing floor in terms of the values of u_0, v_0, h, A, and B in the following.

First, let Q be the plane including L and $\mathrm{O_m}$ with the normal (l, m, n). Then, by (3.11), we have

$$\frac{l}{n}X + \frac{m}{n}Y + Z = c \qquad (4.18)$$

where (X, Y, Z) specify an arbitrary space point P' on Q. Taking P' to be P_0 and replacing l/n and m/n respectively with A and B according to (3.15), we may transform Eq. (4.18) into the following equation:

$$AX_0 + BY_0 - h = c. \qquad (4.19)$$

Next, by substituting (X_0, Y_0, Z_0) and (u_0, v_0) into (4.5) and (4.6) which describe the rotational invariance property of the omni-camera, we get

$$u_0/v_0 = X_0/Y_0. \tag{4.20}$$

Finally, (4.19) and (4.20) may be solved to obtain the following solution for (X_0, Y_0) which specify the on-floor position of a corner point of the H-shape in terms of known values:

$$X_0 = \frac{u_0(h+c)}{Au_0 + Bv_0}; \tag{4.21}$$

$$Y_0 = \frac{v_0(h+c)}{Au_0 + Bv_0}. \tag{4.22}$$

In similar ways, we may derive the real-world positions (X_1, Y_1), (X_2, Y_2), and (X_3, Y_3) for the other three corner points of the H-shape, respectively.

B. Helicopter distance computation

The coordinates (X_C, Y_C) of the center point C of the H-shape on the landing floor can be derived by averaging the coordinates of the four corner points derived above to be $X_C = (X_0 + X_1 + X_2 + X_3)/4$ and $Y_C = (Y_0 + Y_1 + Y_2 + Y_3)/4$ because C is just the geometric center of the four corner points. And the distance d_C of the helicopter to C, which we need to maneuver the helicopter to land right on top of C, is just $d_C = \sqrt{X_C^2 + Y_C^2}$.

4.4 Experimental Results

Experiments in a small-scaled simulation environment as shown in Fig. 4.6 have

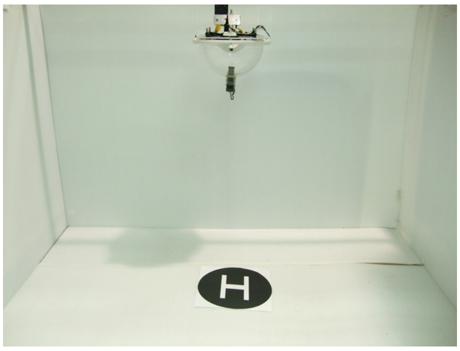


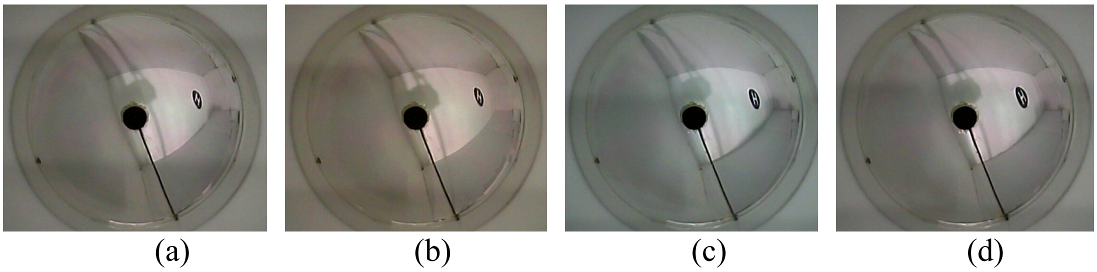
Fig. 4.6 A view of a simulated helipad used in experiments of this study.

been conducted to verify the correctness of the formulas derived previously. A simulated helipad was constructed and the proportion of its size is 1:100. The omni-camera with a hyperboloidal mirror we used for image acquisition was MapCam MRC530N produced by EeRise Co.

In the experiment we conducted for the first stage, a series of images of the simulated helipad with different postures with respect to the omni-camera was taken. Four of them are shown in Fig. 4.7. The real distance and orientation of each posture were measured manually as reference data. Then, each image was processed to extract the circular shape and the derived formulas were used to compute the distance and orientation of the center of the H-shape with respect to the simulated helicopter, which is regarded to be located at the origin of the camera coordinate system origin. An example of the processing results is shown in Fig. 4.8 where Fig. 4.8(a) is the original image, and Fig. 4.8(b) is the result of circular shape extraction using an approximating ellipse. The computation results together with the reference data are shown in Table 4.1. The *error of a computed orientation* is defined as the difference between the real and the computed ones, and the *error ratio of a computed distance* as the ratio of the absolute difference between the real and the computed distances over the real one. From the table, we can see that all the computed orientations have errors smaller than 0.5 degree and that all the computed distances have error ratios smaller than 4.5%. Such results may be considered to be within allowable tolerance for the helipad approaching stage in which the helicopter is maneuvered at a farther distance from the helipad.

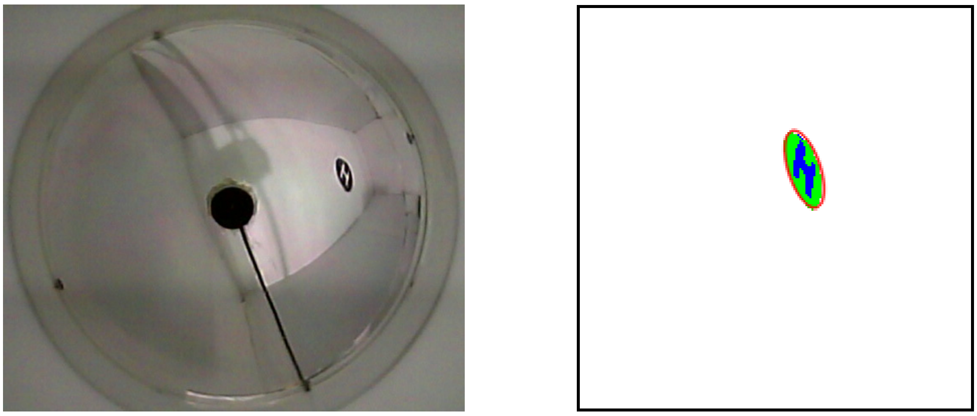
Four images taken for the second stage for helicopter alignment are shown in Fig. 4.9. And an example of image processing results is shown in Fig. 4.10 where Fig. 4.10(a) shows the circular shape extraction result and Fig. 4.10(b) shows the outmost boundary line extraction result. Table 4.2 shows the reference and computed data of the helicopter height h, the distances d_ℓ and d_r of the helicopter with respect to the two outmost boundary lines of the H-shape, and the orientation θ_2 of the helicopter with respect to the leftmost boundary line. From the table, we can see that the error ratios of h, d_ℓ, and d_r are all smaller than 5%, and the errors of θ_2 all smaller than 2.5 degrees. Such errors or error ratios are again considered to be within tolerance for helicopter flying guidance in the helicopter alignment process.

As to the third stage for helicopter docking, four images used in the simulation are shown in Fig. 4.11, and one example of image processing results is shown in Fig. 4.12 which is similar to Fig. 4.10. A comparison of the reference and the computed data of the helicopter height and its distance to the center of the H-shape is shown in Table 4.3, from which again we see the error ratios of the computed data are all smaller than 5%.



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

Fig. 4.7 Four images of approaching stage of helicopter landing process. (a)-(d) Images 1-4.

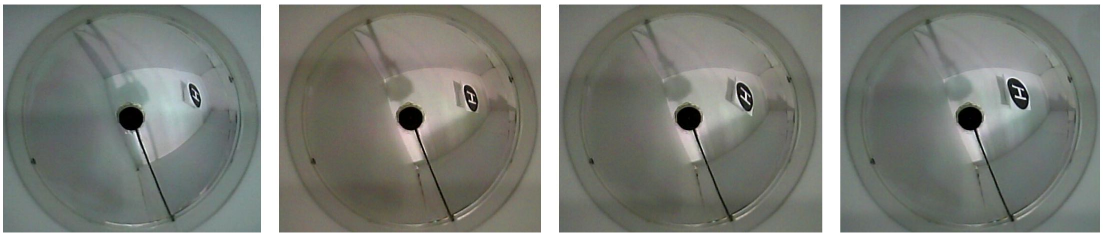


(a) Image 1. (b) Detected shape as an ellipse.

Fig. 4.8 Circular shape detection result of image 1.

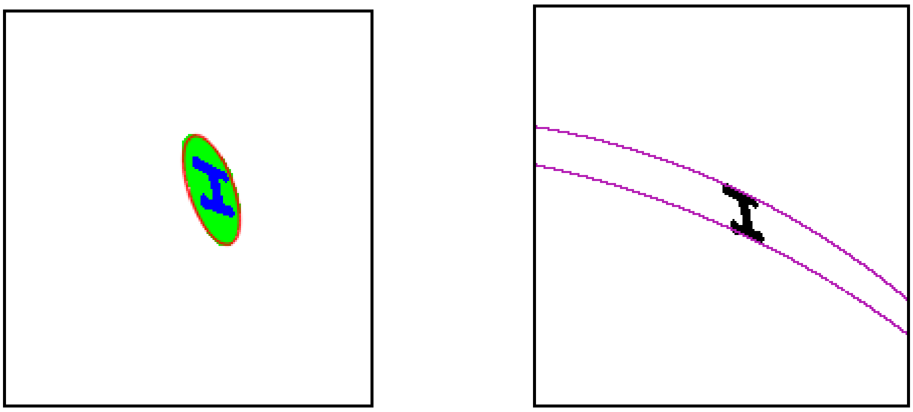
Table 4.1 Experimental result of stage-1 simulation.

Data	Image 1	Image 2	Image 3	Image 4
real orientation	-86.6	-88.2	-87.2	-88.9
computed orientation	-86.0	-87.9	-87.1	-89.3
error of computed orientation	0.5	0.3	0.2	-0.4
real distance	104.1	95.2	85.2	76.9
compute distance	99.8	96.8	87.7	80.4
error ratio of computed distance	4.1%	1.6%	3.0%	4.5%



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

Fig. 4.9 Four images of aligning stage of helicopter landing process. (a)-(d) Images 5-8.



(a) Circular shape detection result. (b) Boundary line detection result.

Fig. 4.10 Circular shape and boundary line detection results of image 7.

Table 4.2 Experimental result of stage-2 simulation.

Data	**Image 5**	**Image 6**	**Image 7**	**Image 8**
real height *h*	39.0	39.0	39.0	39.0
computed height *h*	38.1	40.3	40.2	40.0
error ratio of height *h*	-2.3%	3.3%	3.1%	2.6%
real d_r	39.7	17.7	27.8	18.7
computed d_r	38.0	17.9	27.4	19.2
error ration of d_r	4.2%	1.0%	1.5%	2.6%
real d_ℓ	49.7	27.7	37.8	28.7
computed d_ℓ	47.6	26.7	36.2	28.6
error ration of d_ℓ	-4.1%	-3.5%	-4.1%	-0.3%
real orientation θ_2	-38.0	-17.0	-35.0	-22.0
computed orientation θ_2	-40.3	-19.3	-33.6	-23.0
error of orientation θ_2	-2.3	-2.3	1.4	-1.0

4.5 Concluding Remarks

A novel self-localization method for automatic helicopter landing on a standard helipad using omni-images has been proposed. A helipad with a circled H-shape is used as the landmark for helicopter location estimation. The proposed automatic landing process is divided into three stages: helipad approaching, aligning with the outmost boundary lines of the H-shape, and docking on the center of the H-shape. In each stage, proper geometric features of point, line, and circle on the landmark are used as hints for deriving formulas for computing the location (including the height, distance, or/and orientation) of the helicopter with respect to the landing site. Specifically, a simple conic-section equation for describing a line in an omni-image is derived, which makes possible simple extraction of the H-shape boundary lines by the Hough transform. Formulas for computing helicopter locations for the three stages were skillfully derived, all resulting in analytic forms for faster computations. The proposed techniques are all based on the use of single-view omni-images taken by a hyperboloidal omni-camera, in contrast with traditional methods using multiple views taken by projective cameras. Experimental results with good location estimation precisions have been shown to prove the feasibility of the proposed method. Future researches may be directed to implementing the proposed techniques on a real helicopter test-bed, removing the requirement of maneuvering the helicopter horizontally in the landing process, self-localization using other combinations of the shape features on the helipad, etc.

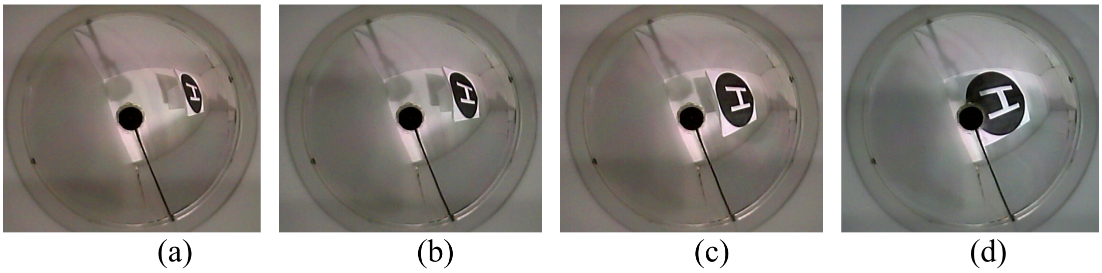
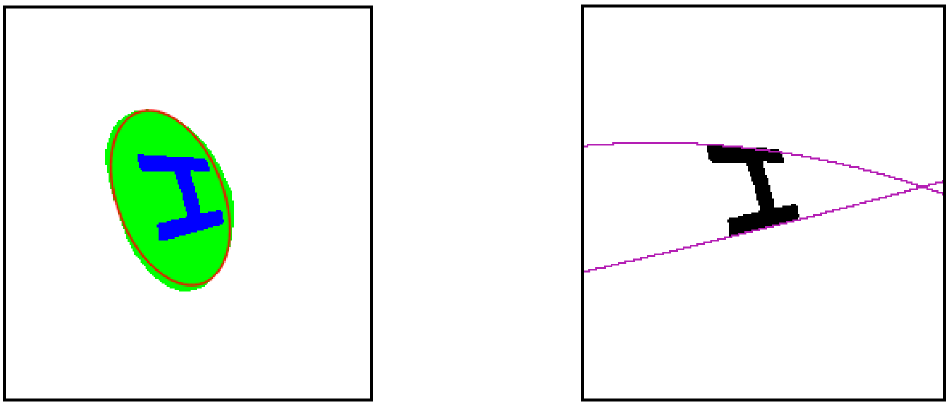

(a) (b) (c) (d)

Fig. 4.11 Four images of docking stage of helicopter landing process. (a)-(d) Images 9-12.



(a) Circular shape detection result. (b) Boundary line detection result.

Fig. 4.12 Circular shape and boundary line detection results of image 11.

Table 4.3 Experimental result of stage-3 simulation.

Data	**Image 9**	**Image 10**	**Image 11**	**Image 12**
real height h	26.5	26.5	26.5	26.5
computed height h	27.6	26.7	26.8	26.7
error ratio of height h	4.1%	0.8%	0.9%	0.8%
real distance d_C	44.5	34.1	24.9	15.2
compute distance d_C	43.1	33.0	24.4	15.4
error ratio of distance d_C	3.1%	3.5%	1.9%	1.3%

Chapter 5
Omni-vision Based Localization of Lateral Vehicles for Car Driving Assistance

5.1 Idea of Proposed Method

Car driving assistance using traditional cameras has been studied intensively in the past two decades, as surveyed in Chapter 1. Recently, omni-cameras with wider FOVs become popular. They are more suitable for car driving assistance because fewer cameras need be equipped. For example, some studies like Lai and Tsai [52] affixed a traditional camera on the right-frontal side of a *host car* to take the image of a *lateral car*. To acquire a larger frontal view, more traditional cameras should be used. Instead, one frontal omni-camera is sufficient. In addition, car wheels are circular-shaped, providing geometric hints for lateral car localization [52]. However, when a circle appears in an omni-image, it becomes irregular in shape, leading to difficulty of extending the existing vehicle localization methods for omni-images.

In this study, we try to solve this problem. The omni-camera is equipped on the frontal bumper *at the height of the wheel* so that the mathematics involved in circular-shape image analysis becomes maneuverable to get analytic solutions for fast computation.

In the remainder of this chapter, the proposed method is described in Section 5.2, followed by some experimental results in Section 5.3 and conclusions in Section 5.4.

5.2 Lateral Car Localization by Frontal Omni-camera

The basic idea of the proposed method is to utilize the geometric properties of a circular-shaped wheel image of the lateral car taken by a single omni-camera to estimate the position and orientation of the lateral car with respect to the host car. The omni-camera, affixed to the frontal bumper of the host car at the height of the wheel, includes a hyperboloidal-shaped mirror. Also, the optical axis of the camera is set to be horizontal to the ground plane. Such an arrangement of the camera makes the resulting irregular shape of the wheel in the omni-image to be extractable as an ellipse using the Hough transform, as proved in [54]. More details are described by the following algorithm.

Algorithm 5.1 ***Lateral car localization by a frontal omni-camera on a host car.***

Step 1. Affix an omni-camera to the frontal bumper of the host car at the height of the wheel center with the camera's optical axis adjusted to be horizontal to the ground plane and pointing to the frontal direction of the host car.

Step 2. Take an image of a wheel of the lateral car and find out the vertical height h of the wheel in the image.

Step 3. With the radius of the wheel and the value of h as input, estimate the position of the lateral car (details described in Section 5.2.1).

Step 4. Find out the farthest and the closest points, I_f and I_c, of the wheel in the image with respect to the image center.

Step 5. With I_f and I_c as input, derive the direction of the lateral car with respect to the host car (details described in Section 5.2.2).

5.2.1. Estimation of Lateral Car Position Using Rotational Invariance Property

The coordinate systems involved in an omni-camera system, including a traditional perspective camera and a hyperboloidal-shaped mirror, are depicted in Fig.

5.1(a), where the omni-camera and the image coordinates are specified by (X, Y, Z), and (u, v), respectively. The perspective camera and the mirror are properly aligned, as assumed, so that the omni-camera becomes a *single-viewpoint* system, and that the optical axis of the perspective camera coincides with the mirror axis which is the line going through the mirror center and perpendicular to the mirror base.

The middle point between the camera lens center O_l and the mirror focus point O_m is taken to be the origin O_a of the omni-camera coordinate system. The hyperboloidal mirror shape may so be described by Eq. (2.1), and O_m is located at (0, 0, $-c$) and O_l at (0, 0, $+c$) in the camera coordinate system where $c = \sqrt{a^2 + b^2}$. The relationship between (u, v) and (X, Y, Z) may be described [54] by Eqs. (2.2) where f is the camera's focal length, and b, c, and f are parameters assumed to be known in advance.

Also, as illustrated in Fig. 5.1(b), the omni-camera is affixed to the car bumper with the Z-axis of the omni-camera adjusted to be at the height of the wheel center, the negative Z-axis directed to the car driving direction, and the Y-axis set perpendicular to the ground surface. A wheel coordinate system x-y-z is defined on the left-frontal wheel of the lateral car with its origin O_w being the wheel center and its x-y plane being the wheel plane. The orientation of the wheel plane is denoted by θ with $\theta = 0^\circ$ meaning that the lateral car moves in parallel. The wheel's radius is assumed to be R_e. Then, defining (X_c, Y_c, Z_c) as the wheel center's coordinates in the omni-camera coordinate system, we get

$$Y_c = 0; \tag{5.1}$$

$$Y = y. \tag{5.2}$$

From the above omni-camera geometry, it is not to difficult to figure out the validity of the so-called rotational *invariance* property, which means that the angle of an incoming light ray formed by a space point at coordinates (X, Y, Z) onto the mirror surface in the omni-camera coordinate system is identical to the angle of the corresponding image point at coordinates (u, v) in the image coordinate system, leading to the following equality:

$$\frac{v}{u}=\frac{Y}{X}. \tag{5.3}$$

The lateral car localization problem now is to derive the wheel position and orientation parameters X_c, Z_c, and θ of the lateral car in the omni-camera coordinate system. First, define P_1 through P_4 as the four extreme points on the wheel circle so that the segments $\overline{P_1P_2}$ and $\overline{P_3P_4}$ are perpendicular and parallel to the ground surface, respectively, as illustrated by Fig. 5.2. Obviously, P_1 and P_2 are at (X_c, $+R_\text{e}$, Z_c) and (X_c, $-R_\text{e}$, Z_c), respectively. Next, it can be figured out that the Y-coordinates of P_3 and P_4 are equal to Y_c, which is zero, because the wheel center O_w at (X_c, Y_c, Z_c) is at the height of the omni-camera coordinate system origin O_a at coordinates (0, 0, 0), as assumed. Denote the image point corresponding to P_i as I_i and its coordinates as (u_i, v_i), i = 1, 2, 3, 4. Applying (5.3) to I_1 and I_2, we get

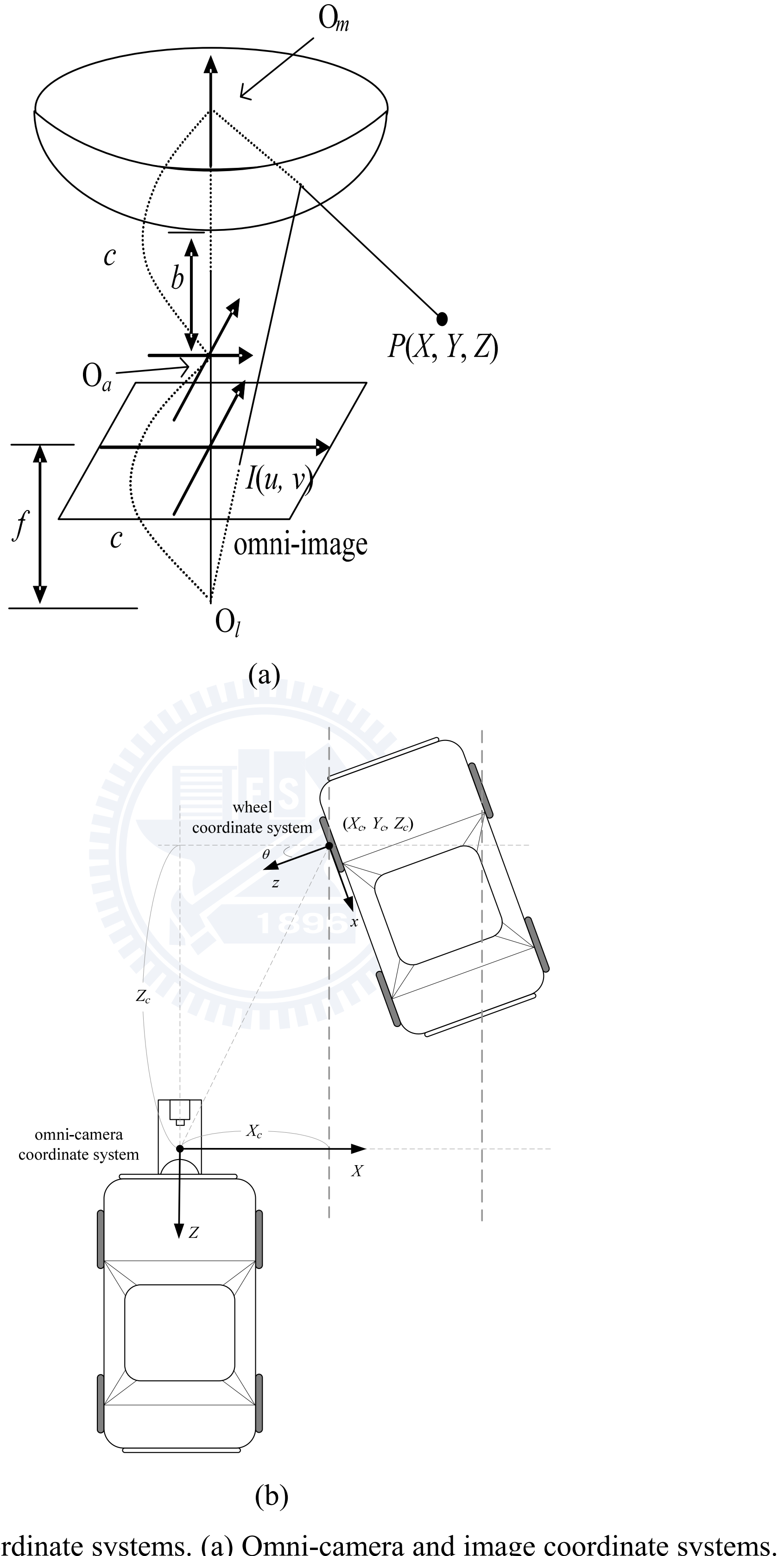
$$\frac{v_1}{u_1}=\frac{R_\text{e}}{X_c};\quad \frac{v_2}{u_2}=\frac{-R_\text{e}}{X_c}. \tag{5.4}$$

Also, from (2.2) and the omni-camera coordinates of P_1 and P_2, we get

$$u_1 = u_2. \tag{5.5}$$

Combining (5.4) and (5.5), we get the solution for X_c as

$$X_c=\frac{2R_\text{e}}{v_1-v_2}u_1. \tag{5.6}$$



(a)

(b)

Fig. 5.2 Relative coordinate systems. (a) Omni-camera and image coordinate systems. (b) Omni-camera and wheel coordinate systems.

Note that $v_1 - v_2$ is just the value h mentioned in Step 3 of Algorithm 5.1.

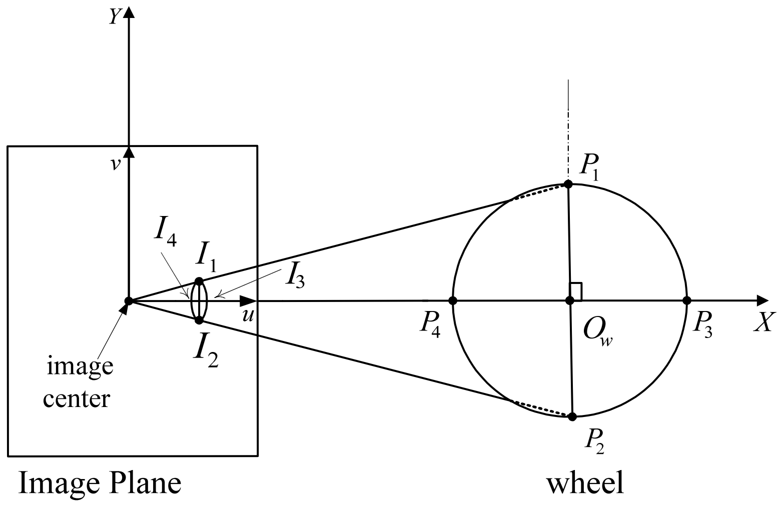


Fig. 5.2 Definition of corresponding image and space points.

To derive Z_c, let

$$Z_c' = Z_c - c. \tag{5.7}$$

Then, (2.2) for I_1 may be transformed into

$$[u_1^2(b^2 + c^2)^2 - u_1^2(-2bc)^2](Z_c')^2 - [2u_1X_cf(b^2 - c^2)(b^2 + c^2)](Z_c') \\ + [f^2(b^2 - c^2)^2X_c^2 - u_1^2(-2bc)^2X_c^2] = 0$$

which leads to

$$A(Z_c')^2 + B(Z_c') + C = 0 \tag{5.8}$$

where

$$\begin{aligned} A &= u_1^2(b^2 - c^2)^2; \\ B &= -2u_1X_cf(b^2 - c^2)(b^2 + c^2); \\ C &= X_c^2[f^2(b^2 - c^2)^2 - 4b^2c^2u_1^2]. \end{aligned} \tag{5.9}$$

So, we get

$$Z_c' = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{5.10}$$

which may be simplified to be

$$Z_c' = \frac{X_c[f(b^2+c^2)\pm 2bc\sqrt{f^2+u_1^{\ 2}}]}{u_1(b^2-c^2)}. \tag{5.11}$$

With the solution for X_c in (5.6), $Z_c' = Z_c - c$ in (5.7), and the equation of (5.11) above, we get finally the solution for Z_c as

$$\begin{aligned} Z_c &= Z_c' + c \\ &= \frac{X_c[f(b^2+c^2)\pm 2bc\sqrt{f^2+u_1^{\ 2}}]}{u_1(b^2-c^2)} + c \\ &= \frac{2R_\text{c}[f(b^2+c^2)\pm 2bc\sqrt{f^2+u_1^{\ 2}}]}{(v_1-v_2)(b^2-c^2)} + c. \end{aligned} \tag{5.12}$$

The sign (+ or −) in (5.12) may be decided experimentally.

5.2.2. Estimation of Lateral Car Orientation Using Wheel Shape Information

Now, we want to use the image coordinates (u_3, v_3) and (u_4, v_4) of I_3 and I_4 (denoted as I_f and I_c respectively in Step 4 of Algorithm 5.1) to derive the wheel orientation θ based on the values of X_c and Z_c obtained previously. First, according to (2.2) and because $Y_3 = Y_4 = Y_\text{c} = 0$, we get $v_3 = v_4 = 0$ and

$$u_3 = \frac{X_3 f(b^2-c^2)}{(b^2+c^2)(Z_3-c)-2bc\sqrt{(Z_3-c)^2+X_3^{\ 2}+Y_3^{\ 2}}}; \tag{5.13}$$

$$u_4 = \frac{X_4 f(b^2-c^2)}{(b^2+c^2)(Z_4-c)-2bc\sqrt{(Z_4-c)^2+X_4^{\ 2}+Y_4^{\ 2}}}. \tag{5.14}$$

Also, define

$$Z_3' = \text{Z}_3 - c; \tag{5.15}$$

$$Z_4' = \text{Z}_4 - c. \tag{5.16}$$

Using (5.13) and (5.14) and through a similar process to that for deriving (5.11), we get

$$Z_3' = \frac{X_3[f(b^2+c^2) \pm 2bc\sqrt{f^2+u_3{}^2}]}{u_3(b^2-c^2)}; \tag{5.17}$$

$$Z_4' = \frac{X_4[f(b^2+c^2) \pm 2bc\sqrt{f^2+u_4{}^2}]}{u_4(b^2-c^2)}. \tag{5.18}$$

Furthermore, with the middle point of $\overline{P_3P_4}$ as the wheel center O$_w$, we get

$$X_4 = 2X_c - X_3; \tag{5.19}$$

$$Z_4 = 2Z_c - Z_3. \tag{5.20}$$

Combining (5.15), (5.16) and (5.20), we have

$$Z_4' = 2Z_c - Z_3' - 2c. \tag{5.21}$$

Also, (5.17) and (5.18) may be transformed into

$$Z_3' = X_3A_3; \tag{5.22}$$

$$Z_4' = X_4A_4 \tag{5.23}$$

where

$$A_3 = \frac{[f(b^2+c^2) \pm 2bc\sqrt{f^2+u_3{}^2}]}{u_3(b^2-c^2)}; \tag{5.24}$$

$$A_4 = \frac{[f(b^2+c^2) \pm 2bc\sqrt{f^2+u_4{}^2}]}{u_4(b^2-c^2)}. \tag{5.25}$$

Combining (5.19) and (5.21) through (5.23), we get

$$X_3 = 2(Z_c - X_cA_4 - c)/(A_3 - A_4). \tag{5.26}$$

And from (5.15), (5.22), and (5.26), we get

$$Z_3 = (2Z_cA_3 - 2X_cA_3A_4 - cA_3 - cA_4)/(A_3 - A_4). \tag{5.27}$$

And from (5.20) and (5.27), we get

$$Z_4 = (2X_cA_3A_4 + cA_3 + cA_4 - 2Z_cA_4)/(A_3 - A_4). \tag{5.28}$$

Accordingly, from (5.23) we get

$$\begin{aligned} X_4 &= Z_4'/A_4 = (Z_4 - c)/A_4 \\ &= (2X_cA_3A_4 + cA_3 + cA_4 - 2Z_cA_4 - c)/[(A_3 - A_4)A_4]. \end{aligned} \tag{5.31}$$

With (5.26) through (5.31), we finally get the desired result

$$\theta = \tan^{-1}(\frac{X_3 - X_4}{Z_3 - Z_4}).$$

5.3 Experimental Results

In our experiments a set of real location data of a lateral car in different postures were measured before corresponding images were taken to estimate the posture parameters X_c, Z_c, and θ using the previously-derived equations. An example of the acquired images is shown in Fig. 5.3, in which detection of a wheel shape as an ellipse is also shown. Some estimation results are shown in Table 5.1. The error for X_c or Z_c is computed as the ratio of the difference between the estimated value and the real one with respect to the real value. And the angle error for θ is computed similarly but with respect to 180° which is the angle range of the lateral car. The table shows that the estimated values of X_c and Y_c are within 5% errors which are good enough for practical applications. But some angle errors are larger. The reason is that the wheel size in the images of these cases appeared to be small, so that the estimated angle θ is sensitive to the width of the horizontal wheel diameter $\overline{P_3 P_4}$.

Moreover, in Section 5.2.1 the radius of the wheel should be known in advance for estimating the vehicle location. However, in real cases of applying the proposed vehicle localization method, the type of the wheel on the vehicle is unknown in advance. A solution is to assume an average wheel radius, which is 20.75cm according to our measurement of a lot of car wheel radiuses. But this way will introduce errors in the estimated position values. Therefore, a simulation experiment was conducted to test whether the errors are tolerable or not. For this, first we project the wheel shape of a lateral car onto the image plane using a set of different real wheel radius parameters (ranging from 19.75cm to 21.75cm as measured by us). Then the proposed method was applied to estimate the position of the lateral car, under the assumption that the radius of the wheel of the car is of the above-mentioned average value 20.75cm. The results are shown in Table 5.2. The average error rates of the position parameters are all smaller than 5%, so the use of a fixed wheel radius in deriving the lateral car location is considered feasible in practice.

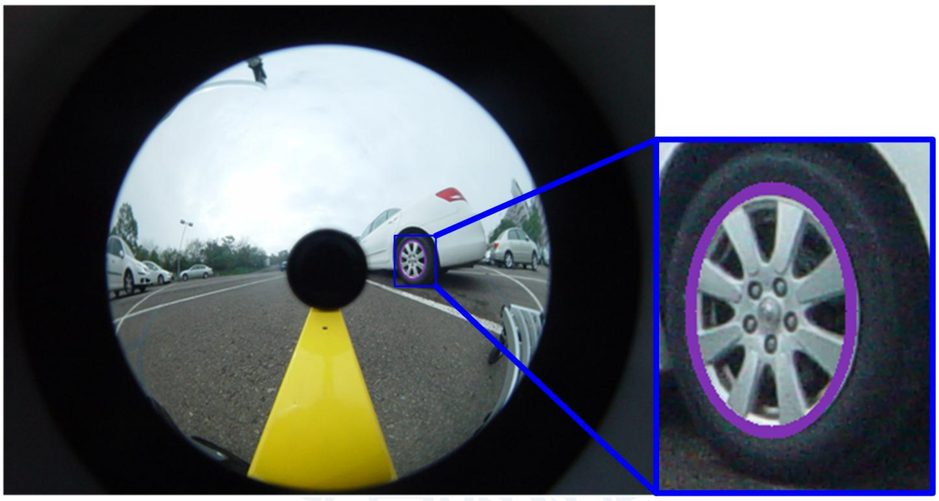


Fig. 5.3 A lateral car image with wheel shape detected as an elliptical shape.

5.4 Concluding Remarks

A lateral vehicle localization method by the use of a single frontal omni-camera has been proposed. The basic concept is to affix a single omni-camera on the bumper at the height of the wheel so that the mathematics involved in the analysis of the irregular shape of the circular wheel in the image becomes maneuverable, leading to the possibility of deriving analytic solutions. Experimental results show that most location estimation results are with error ratios smaller than 6%, which means that the proposed method is feasible for practical applications.

Table 5.2 Lateral car location estimation results.

real X_c (cm)	*estimated X_c (cm)*	*error ratio of X_c*	*real Z_c (cm)*	*estimated Z_c (cm)*	*error ratio of Z_c*	*real θ (degree)*	*estimated θ (degree)*	*angle error (degree)*
-104.3	-99.7	4.4%	-40.9	-39.3	4.0%	-5	-6	0.6%
-390.4	-382.4	2.2%	-87.2	-87.7	0.6%	15	13	1.1%
-464.9	-478.0	4.8%	-494.9	-502.7	1.6%	22	38	8.9%
-137.7	-138.4	0.7%	-159.9	-166.6	4.2%	9	21	6.7%
-407.0	-419.0	4.3%	-115.2	-112.5	2.3%	33	22	6.1%
608.0	606.8	0.6%	-235.5	-240.3	2.0%	-82	-73	5.0%
average error		2.8%			2.5%			4.7%

Table 5.2 Simulation results of estimating lateral car position using a fixed wheel radius value 20.75 cm.

input radius of wheel (cm)	19.75		20.41		21.08		21.75	
position coordinates	*X*	*Z*	*X*	*Z*	*X*	*Z*	*X*	*Z*
input real values (cm)	400	100	400	100	400	100	400	100
estimated values (cm)	419.24	104.81	405.55	101.39	392.73	98.18	380.69	95.17
error ratio	4.81%	4.81%	1.39%	1.39%	−1.82%	−1.82%	−4.82%	−4.82%

Chapter 6
Adaptation of Space-Mapping Methods for Object Location Estimation to Camera Setup Changes

6.1 Idea of Proposed Method

The use of video cameras for various purposes, like security surveillance, autonomous vehicle navigation, and mobile robot applications, has been studied intensively in recent years. One problem encountered in these applications is object localization in an indoor environment using an image of the object acquired by a camera on an autonomous vehicle or affixed to a wall or a ceiling. The result of such object localization is useful for target monitoring, range finding, vehicle or robot guidance, etc.

A conventional solution to the object localization problem is to conduct a work of *camera calibration* to obtain a set of camera parameters, followed by the use the parameters to compute the object location. In this sense, the object location estimation problem is regarded as equivalent to the camera calibration problem. Camera calibration methods often use specially landmarks or environment features to derive formulas or algorithms to compute camera parameters. The computation process is in general complicated and time-consuming. The camera used in such methods is usually mounted on a robot or vehicle and so is mobile, while the landmark or feature used by the methods is usually fixed in the environment.

An alternative way, as mentioned previously in Chapter 1, is to use a *space-mapping* approach which transforms the image space into the real-world space according to a *space-mapping table*. That is, the coordinates of an object point in an acquired image is mapped to the corresponding real-world coordinates of the point by a *table lookup* scheme. Thus, the above-mentioned camera calibration process is avoided, and the approach may be said to conduct *direct* object location estimation. The space-mapping table is constructed in advance, usually with the aid of a certain *calibration pattern*, before the camera is deployed in an application environment, as mentioned in the survey of related studies by Takeshita, Tomizawa, and Ohya [59], and Wang and Tsai [60].

The space-mapping approach to object location estimation, though faster and more convenient, is however *sensitive to camera setup changes*. That is, after a space-mapping table is constructed for a specific camera which was set up in an environment according to a certain camera-environment configuration, the camera should be used *in the same configuration thereafter*; otherwise the space-mapping table will not work. For example, if a space-mapping table was constructed for a downward-looking camera attached on a ceiling at a certain height from the floor, then the camera should be kept downward-looking and at the same height *all the time* when it is used for image acquisition; otherwise, object point locations in the real-world space obtained by table lookup will be incorrect. This weakness causes serious application flexibility problems in using the camera because the camera might be moved to different environments and set up in different ways.

To solve such a *camera-setup sensitive* problem, a commonly way is to abandon the original space-mapping table and reconstruct a new one in the new camera-environment configuration. But this solution is often difficult or impossible to carry out after the camera is delivered to a user who does not know the mapping table construction process or/and has no calibration pattern for use in the process.

In this study, we investigate the possibility of modifying the original space-mapping table for use in new environments with different camera setups for object location estimation, without using the original calibration pattern to repeat the space-mapping table construction process. The camera is assumed to be affixed to the ceiling in this study. Note that this problem of adapting the space-mapping table to new camera setups has not been studied so far.

More specifically, a method is proposed in this study to solve the space-mapping table adaptation problem. The original space-mapping table is not abandoned, but utilized instead, in such a way that its content of real-world coordinates is mapped further to correct values for use in the new camera setup environment. From the viewpoint of object location estimation, the proposed method may be regarded as a two-stage solution. In the first stage which is supposed to be conducted *in the factory*, the camera is set up according to a specific configuration, and a corresponding space-mapping table is constructed using a calibration pattern and a bilinear interpolation technique. Then the camera is allowed to be set up in a possibly different way, including ceiling height or camera orientation changes, and the second stage is carried out to deal with such changes by modifying the space-mapping table content using some formulas which are derived according to principles of image formation, geometry, and trigonometry. The proposed method is applicable to various types of cameras, including conventional camera, fish-eye camera, catadioptric cameras, etc.

In the remainder of this chapter, we first describe the idea of the proposed two-stage object location estimation method as an algorithm in Section 6.2. Techniques for constructions and modifications of space-mapping tables are presented in Section 6.3. Some experimental results showing the feasibility of the proposed method and the precision of location estimation data yielded by the method are given in Section 6.4, followed by conclusions and suggestions for further studies in Section 6.5.

6.2 Idea of Proposed Method

The proposed object location estimation method includes two stages, one conducted in in-factory environments and the other in in-field ones. In the first stage where a camera is supposed to be manufactured, a *basic space-mapping table* is constructed using a pre-selected calibration pattern with the camera being affixed to a ceiling at a certain height and with its optical axis pointing downward to the floor, as illustrated by Fig. 6.1. In the second stage, the basic mapping table is applicable if the camera setup is unchanged; if not, then the change of the camera setup is identified into two types, namely, a ceiling height change or a camera orientation change as illustrated by Fig. 6.2. The basic mapping table is modified accordingly. More details are described in the following algorithm.

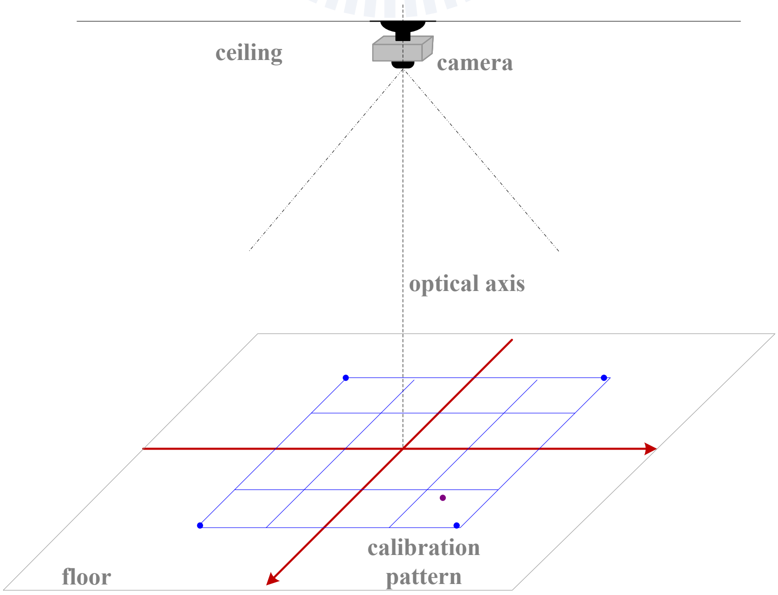


Fig. 6.1 Illustration of camera setup for space-mapping table construction in Stage 1 of proposed method.

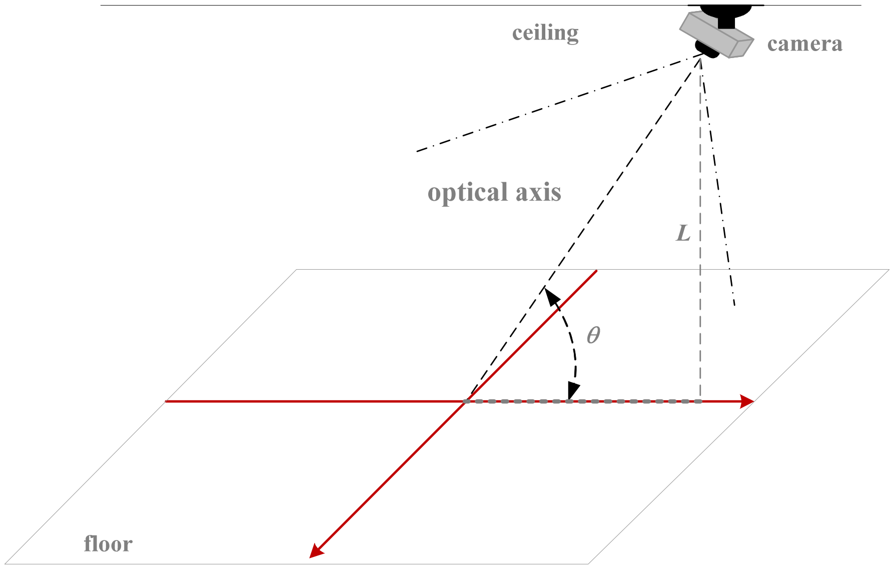


Fig. 6.2 Illustration of camera orientation change (with a tilt angle of θ).

Algorithm 6.1 Space-mapping table construction and modification for object location estimation.

Stage 1. Construction of a basic mapping table in the factory.

Step 1. Affix the camera to the ceiling at height H_0 with the camera's optical axis pointing to the floor perpendicularly, assuming that the surfaces of the floor and the ceiling are both flat and mutually parallel.

Step 2. Place a calibration pattern O right under the camera, take an image of it, extract all the feature points in the image, and find the image coordinates of them.

Step 3. Measure the real-world coordinates of the points in the calibration pattern, which correspond to the extracted feature points in the image.

Step 4. (*Quadrilateral mapping*) With the corresponding image and real-world coordinates obtained above as input, use a *quadrilateral mapping* technique to construct a *basic* space-mapping table T in a form like Table 6.1, which maps each image coordinate pair (u_i, v_j) to a real-world coordinate pair (x_{ij}, y_{ij}), that is, T: $(u_i, v_j) \rightarrow (x_{ij}, y_{ij})$.

Stage 2. Construction of a modified space-mapping table for a new ceiling height or a new camera orientation in in-field environment.

Step 5. (*Ceiling-height adaptation*) If the in-field camera setup to be carried out includes just a change of the original ceiling height H_0, then perform the following operations to modify the basic space-mapping table T; else, go to the next step (Step 6).

5.1 Affix the camera to the ceiling, measure the ceiling height with respective to the floor, and denote it as H_1.

5.2 Modify table T to construct a new one with H_1 as input by a technique called *ceiling height adaptation* using some error-correcting formulas.

5.3 Skip the next step and Go to Step 7

Step 6. (*Camera-orientation adaptation*) Perform the following operations to modify the basic space-mapping table T.

6.1 Affix the camera to the ceiling, measure the ceiling height and the camera's orientation with respective to the floor, and denote them as L and θ.

6.2 Modify table T with L and θ as input by a technique called *camera orientation adaptation* using some error-correcting formulas derived later.

6.3 Go to the next step.

Step 7. (Location estimation using space-mapping table T) Locate an object B in the real-world space using T_1 in the following way.

7.1 Acquired an image I of B with the camera.

7.2 Detect B in I and find a feature point p on it with image coordinates (u, v).

7.3 Use (u, v) to look up T to get the real-world coordinates (x, y) of the real-world point P corresponding to p as the desired object location.

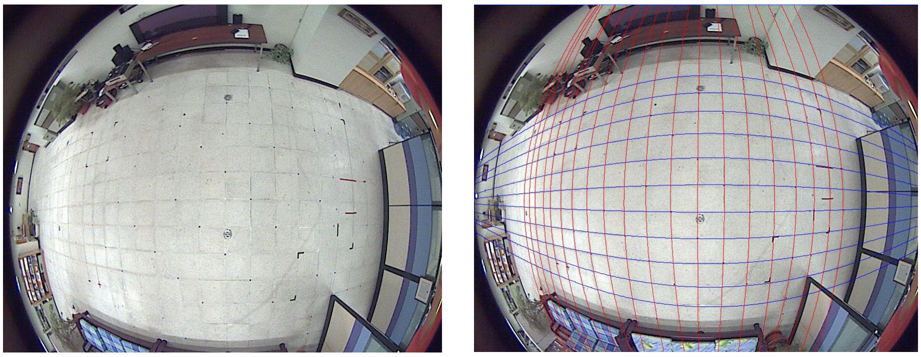
Table 6.1 A basic space-mapping table which records relations between coordinates of corresponding image points and real-world points.

	u_1	u_2	…	u_m
v_1	(x_{11}, y_{11})	(x_{21}, y_{21})	…	(x_{m1}, y_{m1})
v_2	(x_{12}, y_{12})	(x_{22}, y_{22})	…	(x_{m2}, y_{m2})
…	…	…	…	…
v_n	(x_{1n}, y_{1n})	(x_{2n}, y_{2n})	…	(x_{mn}, y_{mn})

6.3 Proposed Techniques for Basic Mapping Table Construction and Modifications for Ceiling Height and Camera Orientation Adaption

6.3.1. Basic Mapping Table Construction by Quadrilateral Mapping

The quadrilateral mapping technique mentioned in Step 4 of Algorithm 6.1 proposed in this study constructs a space-mapping table T like Table 1 by two steps: finding pairs of corresponding quadrilaterals in the calibration pattern in the image and in that in the real world, followed by transformations of image and real-world coordinates of corresponding points within the quadrilaterals based on the use of a bilinear interpolation technique. The calibration pattern used in this study is a grid pattern formed by tiles on the floor, as shown in Fig. 6.3(a). Each grid is of the shape of a rectangle. The camera used in this study is a fish-eye one. Each grid in an image acquired by the camera becomes a quadrilateral when the grid is small enough, as assumed in this study. The quadrilateral shapes in an acquired image are extracted by detecting the tile corners, connecting the extracted corner points with approximating quadratic curves, and segmenting areas enclosed by two orthogonal pairs of opposite boundary curve segments. As an example, the extraction result of Fig. 6.3(a) is shown in Fig. 6.3(b). in which each extracted quadrilateral consists of a pair of “vertical” red boundary segments and a pair of “horizontal” blue boundary segments.



(a) (b)

Fig. 6.3 Illustration of quadrilateral extraction using a grid pattern on floor. (a) An image of the grid pattern. (b) The lines approximating the grid lines.

Suppose that the quadrilateral in the image plane shown in Fig. 6.4 is one of the quadrilaterals so extracted, corresponding to a grid on the floor. The points p_1, p_2, p_3, and p_4 are the four corners of the quadrilateral within which a point p lies. The location of the point P in the real world corresponding to p is what we want to estimate. The image coordinates of the corners and the point p in the image are denoted by (u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4), and (u_p, v_p), respectively, and their corresponding real-world coordinates are denoted by(x_1, y_1), (x_2, y_1), (x_1, y_2), (x_2, y_2), and (x_P, y_P), respectively, as shown in Fig. 6.5. Note that we assume the grid is a rectangle.

To compute (x, y) of the real-world point P corresponding to p, we adopt the technique of inverse bilinear interpolation [62] We formulate first a forward bilinear transformation T from the real-world space to the image space and then find its reverse T^{-1}, as described in the following (see Fig. 6.5 for the notations).

(1) Compute the weights $x = (x_P - x_1)/(x_2 - x_1)$, $y = (y_P - y_1)/(y_2 - y_1)$.

(2) Perform linear interpolation in the x-direction for the upper and lower boundaries of the quadrilateral to get $\bar{u}_\mathrm{u} = (1 - x)\bar{u}_1 + x\bar{u}_2$ and $\bar{u}_\mathrm{L} = (1 - x)\bar{u}_3 + x\bar{u}_4$ where the vectors $\bar{u}_1 = [u_1\ v_1]^t$, $\bar{u}_2 = [u_2\ v_2]^t$, $\bar{u}_3 = [u_3\ v_3]^t$, $\bar{u}_4 = [u_4\ v_4]^t$ and "t" means the vector transpose operation.

(3) Perform linear interpolation in the y-direction for the line L_i going through p to get $\bar{u} = (1 - y)\bar{u}_\mathrm{u} + y\bar{u}_\mathrm{L}$ where $\bar{u} = [u_p\ v_p]^t$.

(4) Expand the equalities obtained in the above two steps to get $\bar{u} = (1 - y)\bar{u}_\mathrm{u} + y\bar{u}_\mathrm{L} = (1 - x)(1 - y)\bar{u}_1 + x(1 - y)\bar{u}_2 + (1 - x)y\bar{u}_3 + xy\bar{u}_4$, or equivalently,

$$u_p = axy + bx + cy + d;\ v_p = exy + fx + gy + h$$

where $a = u_3 - u_2 + u_1 - u_4$, $b = u_2 - u_1$, $c = u_4 - u_1$, $d = u_1$, $e = v_3 - v_2 + v_1 - v_4$, $f = v_2 - v_1$, $g = v_4 - v_1$, $h = v_1$.

(5) Use the two equalities in the above step with the known image coordinates (u_p, v_p) to solve the values of (x, y) as follows (the details omitted):

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A};\ y = \frac{u - gx - d}{ax + c}$$

where $A = af - be$, $B = eu_p - av_p + ah - de + cf - bg$, $C = gu_p - cv_p + ch - dg$.

(6) Solve the equalities established in Step (1) to get the desired values of x_P and y_P as

$$x_P = x(x_2 - x_1) + x_1;\ y_P = y(y_2 - y_1) + y_1.$$

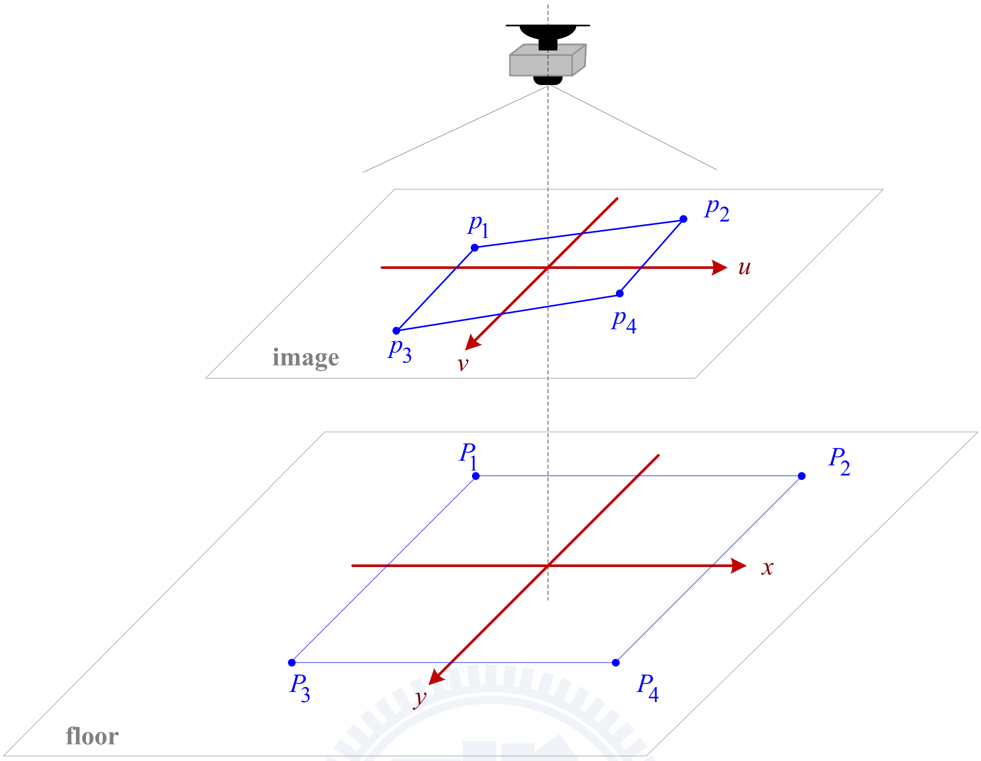


Fig. 6.4 Mapping of a pair of corresponding quadrilaterals in image and in calibration pattern.

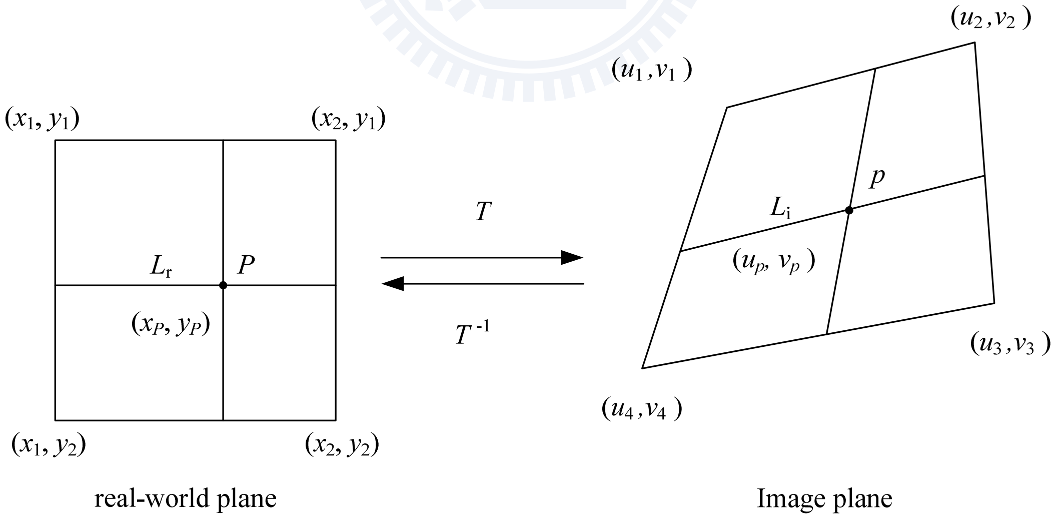


Fig. 6.5 Location estimation of a real-world point by inverse bilinear interpolation.

6.3.2. Mapping Table Modification According to Change of Floor Height

After the basic space-mapping table is obtained with the camera affixed to a ceiling at a certain height H_0 with respect to a parallel floor surface F_0, if the camera is used later in an environment with the ceiling height changed to another value H_1 with respect to a second parallel floor surface F_1, then the basic space-mapping table is no more applicable, the table lookup result for location estimation becomes incorrect, and modification of the table content is necessary, which we call *ceiling-height adaptation* in Step 6 in Algorithm 6.1. The idea behind the proposed technique for this purpose is described first in the following.

It is known that an image point p is formed in principle by *any* of the real-world points which all lie on a light ray R going into the camera's lens and then onto the image plane. As illustrated in Fig. 6.6, suppose that this light ray R intersects both the floor surface F_0 at P_0 and the floor surface F_1 at P_1. If the image coordinates of p are (u, v), then the real-world coordinates (x_0, y_0) in the basic mapping table corresponding to (u, v) actually are those of P_0 on F_0 instead of being the desired ones, (x_1, y_1), of P_1 on F_1. To correct this erroneous result, we have the following equalities according to the concept of side proportionality in a triangle:

$$\frac{x_0}{x_1} = \frac{H_0}{H_1}; \quad \frac{y_0}{y_1} = \frac{H_0}{H_1}, \tag{6.1}$$

or equivalently,

$$x_1 = x_0 \frac{H_1}{H_0}; \quad y_1 = y_0 \frac{H_1}{H_0}. \tag{6.2}$$

That is, the table lookup result (x_0, y_0) corresponding to the image coordinates (u, v) of a real-world point P_1 on floor surface F_1 should be magnified in proportion to H_1/H_0

to be (x_1, y_1) as the desired result. Note that in the above discussion, we assume that the real-world coordinate system *x*-*y*-*z* is set up at the camera's lens with the lens center as the origin and the optical axis as the z-axis. And the location of the object

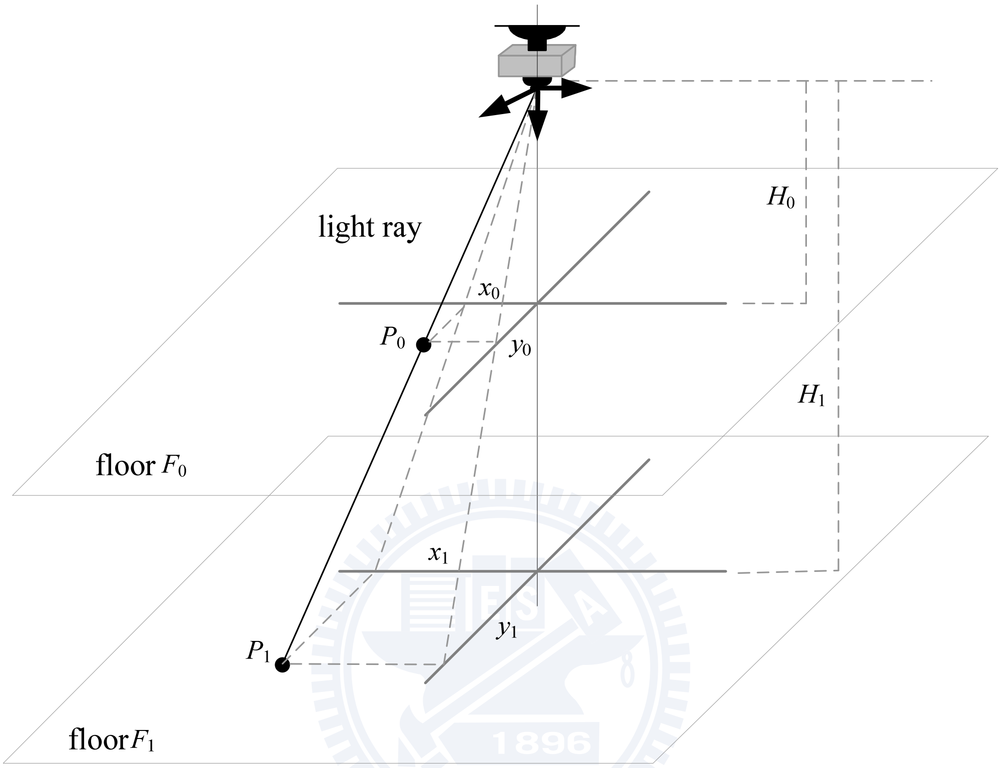


Fig. 6.6 Illustration of using side proportionality to compute coordinates of point P_1 on a floor F_1 with ceiling height H_1.

point *P* described by (x_1, y_1) is measured with respect to this coordinate system. Also, by a camera height, we mean more precisely the distance from the lens center right down to the floor.

6.3.3. Mapping Table Modification According to Change of Camera Orientation

Assume now that the camera is affixed to the ceiling with a tilt angle of θ and a height of *L* with respect to floor F_1, as illustrated in Fig. 6.7. Here, the location of the object point P_1 on F_1, which we want to estimate, is specified by the real-world coordinates (x_1, y_1) with respect to the downward projection point O of the camera's lens center onto F_1, where the x-axis is assumed to be coincident with the projection of the camera's optical axis on F_1. Let the coordinates of P_1 in the acquired image be (u, v). Again the space-mapping table is inapplicable here; the table lookup result, the real-world coordinates (x_0, y_0), are actually those of a real-world point on a floor surface F_0 at a distance of H_0 to the camera's lens center, instead of being the desired real-world coordinates (x_1, y_1) of P_1 on F_1. Table modification is necessary, which is called *camera orientation adaptation* in Step 6 of Algorithm 6.1.

To correct the values (x_0, y_0) into (x_1, y_1), we rotate floor surface F_1 through an angle of $90^\circ - \theta$ with P_1 as the rotation pivot point, such that the resulting surface plane F_1' becomes perpendicular to the camera's optical axis and the lateral view of the rotation result seen from the positive y-axis direction becomes the one shown in Fig. 6.8. The original floor surface F_0 is also shown in the figure.

Assume that the distance of P_1 on F_1' to the camera's optical axis is x'. Then, according to the concept of side proportionality again, we have

$$\frac{x_0}{x'} = \frac{H_0}{H_1}. \tag{6.3}$$

Also, by geometry and trigonometry we have

$$\sin\theta = \frac{x'}{M}; \tag{6.4}$$

$$\sin\theta = \frac{L}{N + H_0}; \tag{6.5}$$

$$\cos\theta = \frac{x_1 - M}{N + H_0}; \tag{6.6}$$

$$\cos\theta = \frac{H_1 - (N + H_0)}{M}. \tag{6.7}$$

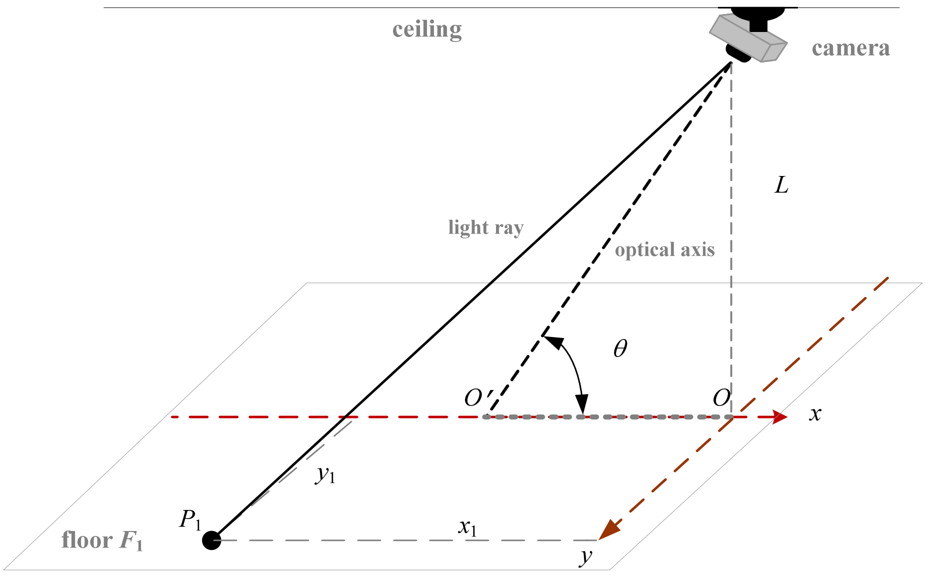


Fig. 6.7 Illustration of a tilted camera with angle θ with respect to the x-axis of the real-world coordinate system.

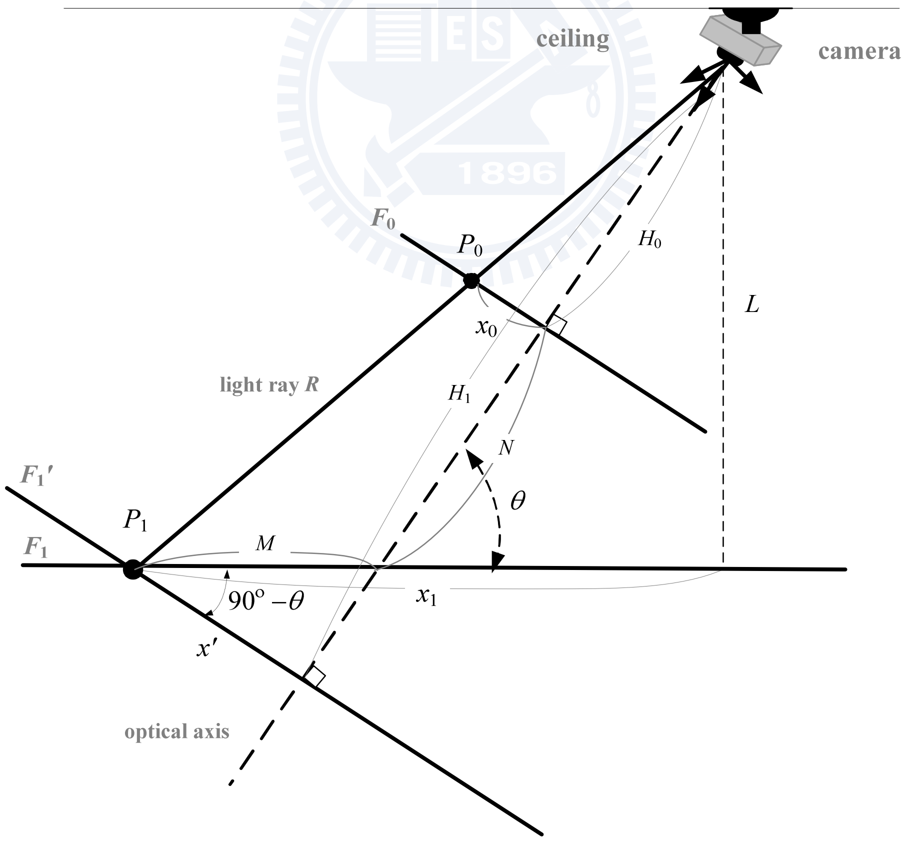


Fig. 6.8 Lateral view (from the positive y-axis direction) of rotation result of floor surface F_1 in Fig. 6.7 through an angle of $90^{\circ} - \theta$ with P_1 as the rotation pivot point.

From (6.5) and (6.6), we get $N + H_0 = L/\sin\theta = (x_1 - M)/\cos\theta$, or equivalently,

$$(x_1 - M)\sin\theta = L\cos\theta. \tag{6.8}$$

Also, from (6.3) and (6.4), we get $x_0M/\sin\theta = H_0/H_1$, or equivalently,

$$H_1 = \frac{H_0 \sin\theta}{x_0 M} . \tag{6.9}$$

And from (6.5), (6.7) and (6.9), we get

$$M\cos\theta = \frac{H_0 M \sin\theta}{x_0} - \frac{L}{\sin\theta} ,$$

or equivalently,

$$M = \frac{L}{\sin\theta} \times \frac{x_0}{H_0 \sin\theta - x_0 \cos\theta} . \tag{6.10}$$

From (6.8) and (6.10), we get

$$x_1 = L(\cot\theta - \csc\theta \times \frac{x_0}{x_0 \cos\theta - H_0 \sin\theta})$$

which can be reduced to be

$$x_1 = L \times \frac{H_0 \cos\theta + x_0 \sin\theta}{H_0 \sin\theta - x_0 \cos\theta} . \tag{6.11}$$

As a verification of the above equality, when $\theta = 90^{\circ}$, it reduces to $x_1 = L\times(x_0/H_0) = x_0\times(L/H_0)$, which exactly is the case described by Eqs. (6.2) and depicted by Fig. 6.6. The correctness of (6.11) may also be seen from Fig. 6.9 because according to trigonometry we have $H_0\cos\theta = a$, $x_0\sin\theta = b$, $H_0\sin\theta = c + d$, $x_0\cos\theta = d$, so that

$$L \times \frac{H_0 \cos\theta + x_0 \sin\theta}{H_0 \sin\theta - x_0 \cos\theta} = L \times \frac{a+b}{(c+d)-d} = L \times \frac{a+b}{c} = L\cot\theta' = x_1 \tag{6.12}$$

which is just (6.11).

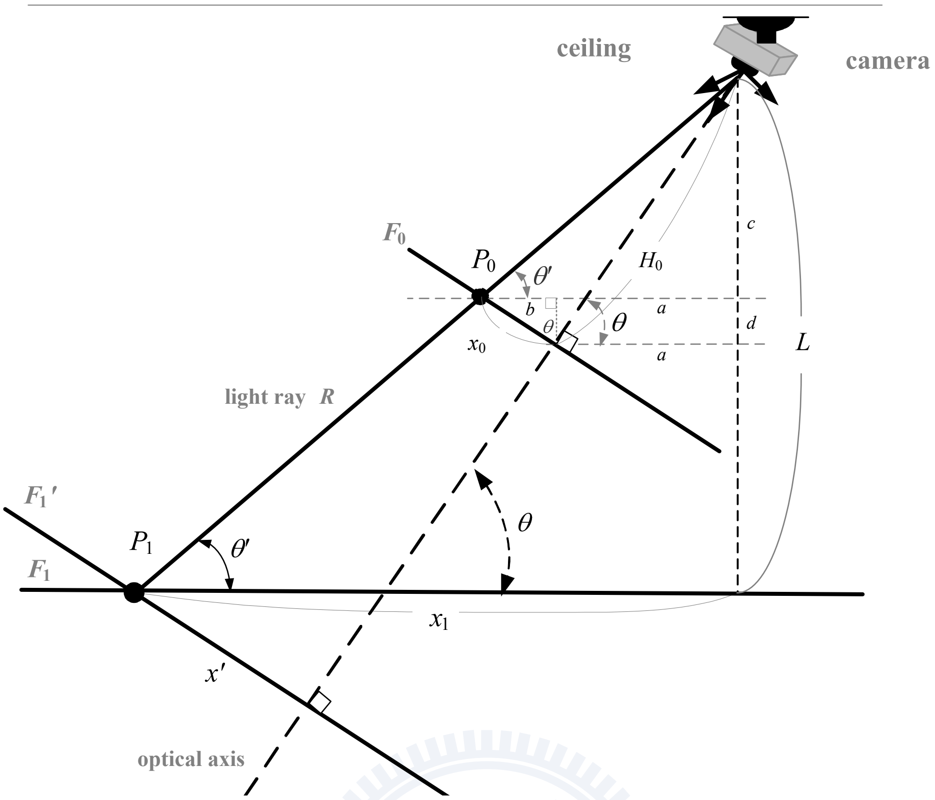


Fig. 6.9 Illustration for verification of correctness of Eq. (6.11).

On the other hand, because the *x*-axis on F_1 is assumed to be coincident with the projection of the camera's optical axis on F_1, and because the rotation of F_1 into F_1' is pivoted in the *y*-direction, we have $y' = y_1$. Also, according to Eqs. (6.2) we have $\frac{y'}{y_0} = \frac{H_1}{H_0} = \frac{x'}{x_0}$. Therefore,

$$y_1 = y' = y_0 \frac{x'}{x_0} . \qquad (6.13)$$

From (6.4), (6.10) and (6.13), we can get

$$y_1 = L \times \frac{y_0}{H_0 \sin\theta - x_0 \cos\theta} . \qquad (6.14)$$

As a verification of (6.14), when $\theta = 90^0$, it reduces to $y_1 = L\times(y_0/H_0) = y_0\times(L/H_0)$ which is again exactly the case described by Eqs. (6.2) and depicted by Fig. 6.6. The correctness of (6.14) may also be seen from Fig. 6.10 which is a lateral view of Fig.

6.8 from the positive x-axis direction, because from the previous analysis of (6.12), we have $H_0\sin\theta - x_0\cos\theta = c$ so that

$$L\times\frac{y_0}{H_0\sin\theta - x_0\cos\theta} = L\times\frac{y_0}{c} = L\cot\theta_1 = y_1 \tag{6.15}$$

which is just (6.14).

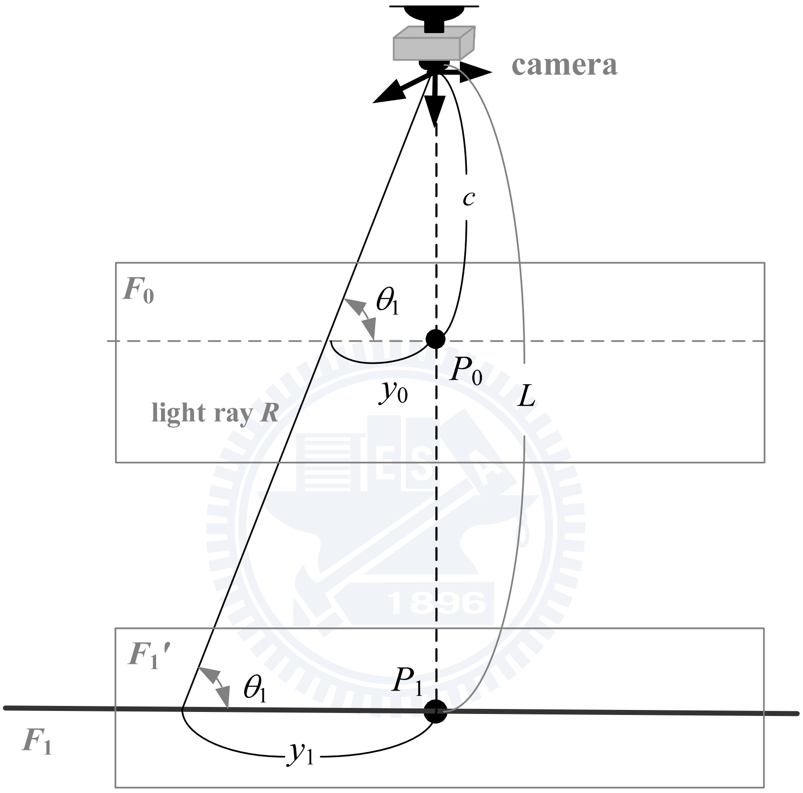


Fig. 6.10 Lateral view of Fig. 6.8 from direction of positive x-axis for verification of correctness of Eq. (6.14).

6.4 Experimental Results

A series of experiments have been conducted to test the correctness and precision of the proposed method for object location estimation. The fish-eye camera used in the experiments is shown in Fig. 6.11. It is attached to a rotator connected to a rod with an adjustable length. The camera can so be tilted arbitrarily by rotating the rotator, and raised to any height by adjusting the rod length, to simulate environments of different ceiling heights and camera orientations. An image taken with the camera looking right downward (i.e., with the tilt angle of 90°) is shown in Fig. 6.3. We show additionally here three images (Figs. 6.12(a) through 12(c)) taken with the camera in three distinct setups, which are used in our experiments reported in this section: (1) looking downward at the ceiling height of 200cm; (2) looking downward at the ceiling height of 250cm; (3) tilted for the angle of 50° at the ceiling height of 200cm. The images are all of the resolution of 1280×1024.

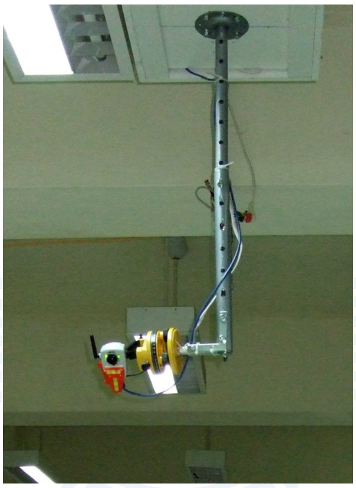


Fig. 6.11 Fish-eye camera used in this study, which is attached to a rod fixed on ceiling and can be tilted and moved up and down.

Case (1) is regarded as the *original* camera setup configuration used in the factory for building a basic space-mapping table. After the image of Fig. 6.12(a) was taken with the downward-looking camera at the ceiling height of 200cm, all the grid points in the image are extracted to get their image coordinates, forming a set denoted by I_c. Also, the real-world coordinates of each grid point are measured manually to form a set denoted by W_c. The two sets I_c and W_c of coordinate data are then used to construct a basic space-mapping table T by the process described in Section 6.3.1. To test the precision of the constructed table T, nine non-grid points among the grid ones, which also appear in Fig. 6.12(a), were selected and their image coordinates collected to form a set I_c'. Also, the real-world coordinates of these non-grid points are measured manually to form another set W_c'. The set I_c' then is used to obtain their corresponding real-world coordinates by table lookup using T, forming a set denoted W_c''. Finally, the two sets W_c' and W_c'' are compared and two types of error ratio measures are defined to compute the similarity between them as follows.

(1) Type 1 --- location error ratio with respect to the *distance from the real-world point to the camera's lens center*:

$$location\ error\ ratio = \frac{\sqrt{(real\ x_i - estimated\ x_i)^2 + (real\ y_i - estimated\ y_i)^2}}{\sqrt{real\ x_i^2 + real\ y_i^2 + L^2}}$$

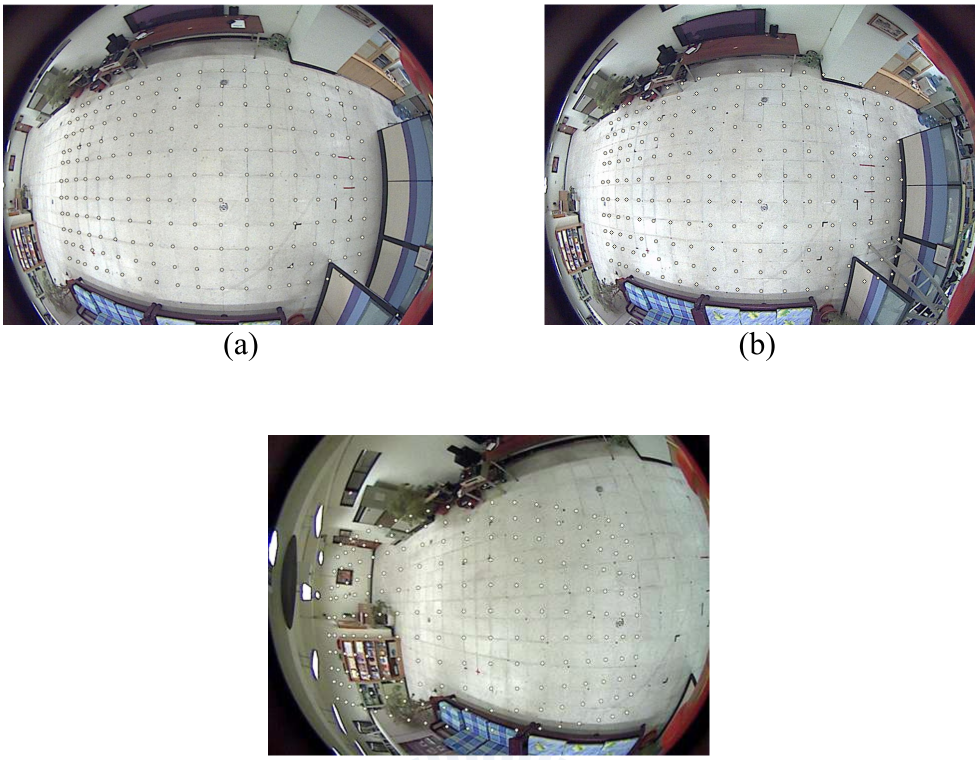
where *real* x_i and *real* y_i are data in W_c' and *estimated* x_i and *estimated* y_i are data in W_c''.

(2) Type 2 --- location error ratio with respect to the *effective field of view of the camera (see Fig. 6.13)*:

$$location\ error\ ratio = \frac{\sqrt{(real\ x_i - estimated\ x_i)^2 + (real\ y_i - estimated\ y_i)^2}}{\text{radius of effective camera's field of view}}.$$

The computed results for the two types of error ratios are summarized as a table as shown in Table 6.2, from which we can see that the ratios are all small then 5% which is practical for object location estimation applications like robot or vehicle guidance in indoor environments.

For Case (2), the camera, still looking downward, was affixed at a different height of 250cm and the previously-mentioned process of error ratio computation was



(a) (b)

(c)

Fig. 6.12 Images used for experiments reported here. (a) Taken with camera looking downward at ceiling height of 200cm. (b) Taken with camera looking downward at ceiling height of 250cm. (c) Taken with camera tilted for 50^o at ceiling height of 200cm.

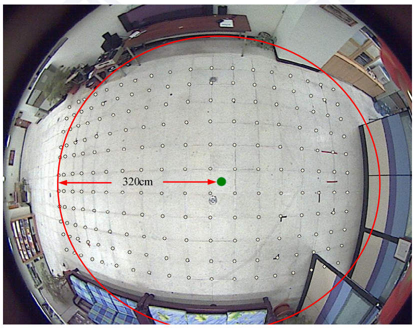


Fig. 6.13 Effective field of view of camera measured by radius of an enclosing red circle.

repeated after the proposed method was applied to the image of Fig. 6.12(b). The results were again summarized as a table shown in Table 6.3, from which we can see that the ratios are all small then 5% as well.

Table 6.2 Error ratios with camera looking downward at ceiling height 200cm.

real x (cm)	*estimated x (cm)*	*real y (cm)*	*estimated y (cm)*	*distance to origin (cm)*	*type-1 error ratio*	*type-2 error ratio*
-7	-8	-24	-23	25	0.7%	0.4%
-37	-36	36	36	52	0.5%	0.3%
-20	-20	96	94	98	0.9%	0.6%
-45	-44	-107	-106	116	0.6%	0.4%
-111	-112	-55	-56	124	0.6%	0.4%
-140	-140	62	60	153	0.8%	0.6%
-229	-228	-101	-104	250	1.0%	1.0%
-253	-257	76	82	264	2.2%	2.3%
-320	-317	-15	-15	320	0.8%	0.9%

Table 6.3 Error ratios with camera looking downward at ceiling height 250cm.

real x (cm)	*estimated x (cm)*	*real y (cm)*	*estimated y (cm)*	*distance to origin (cm)*	*type-1 error ratio*	*type-2 error ratio*
-7	-9	-24	-23	25	0.9%	0.7%
-37	-38	36	37	52	0.6%	0.4%
-20	-21	96	94	98	0.8%	0.7%
-45	-48	-107	-109	116	1.3%	1.1%
-111	-117	-55	-57	124	2.3%	2.0%
-140	-145	62	62	153	1.7%	1.6%
-229	-238	-101	-110	250	3.6%	4.0%
-253	-264	76	80	264	3.2%	3.7%
-320	-335	-15	-14	320	3.9%	4.7%

Similarly, for Case (3) where the camera was affixed at the ceiling height 200cm and tilted for 50°, the error ratio table constructed for the image of Fig. 6.12(c) is shown in Table 6.4, from which we see that the ratios are *not* all small then 5% this time; some are larger (6.0% and 7.1% for the last row in the table). The reason for this phenomenon is that the object point dealt with is located at (-320, -15) which is quite far away from the center of the image, falls within a distorted-shaped quadrilateral, and so incurs a larger error in the process of quadrilateral mapping described in Section 6.3.1.

Table 6.4 Error ratios with camera looking downward at ceiling height 250cm.

real x (cm)	*estimated x (cm)*	*real y (cm)*	*estimated y (cm)*	*distance to origin (cm)*	*type-1 error ratio*	*type-2 error ratio*
-7	-7	-24	-22	25	1.0%	0.6%
-37	-38	36	34	52	1.1%	0.7%
-20	-21	96	90	98	2.7%	1.9%
-45	-42	-107	-103	116	2.2%	1.6%
-111	-110	-55	-60	124	2.2%	1.6%
-140	-141	62	58	153	1.6%	1.3%
-229	-238	-101	-110	250	4.0%	4.0%
-253	-271	76	81	264	4.5%	4.6%
-320	-340	-15	-26	320	6.0%	7.1%

For Case (4) where the camera was affixed at the ceiling height 200cm but tilted for 90^{o} (looking down), 70^{o}, and 50^{o}, respectively, the error ratio table is shown in Table 6.5, from which we see that the average error ratios are slightly increased by 1~2% as the titled angle changes. For the similar reason as Case (3), the larger change occurs only in the positions far away from the image center.

Two possible applications of the proposed method are guidance of autonomous vehicles for human tracking and security patrolling, which were conducted in our laboratory. Two images taken in such application studies are shown in Fig. 6.14. The vehicle location estimation results using the proposed method were used for path correction and planning in these studies.

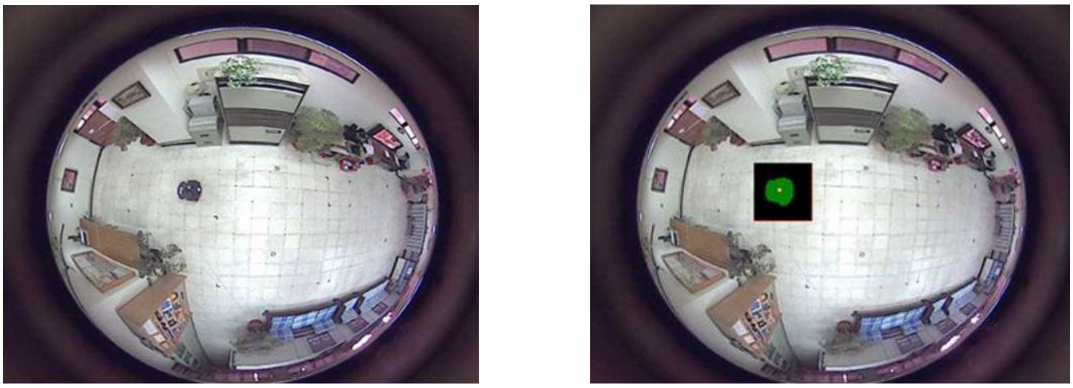
6.5 Concluding Remarks

A general space-mapping method for object location estimation by modifications of the basic space-mapping table to adapt it to camera setup changes has been proposed. The method does not require the conventional camera calibration process, and is general for any type of camera. The method estimates the location of an object by mapping the image coordinates of an object point to the real-world coordinates of the point using a space-mapping table. An algorithm is designed to construct the table, which consists of two stages with the first stage for constructing a basic space-mapping table using a bilinear interpolation technique and the second stage for modifying the basic table to adapt it to changes of camera setups, including camera height and orientation adjustments, which often occur after the camera is delivered to a user for uses in the application environment. The proposed techniques for table modifications are based on a concept of image formation by light rays as well as several properties of geometry and trigonometry. Such a problem of adapting the space-mapping method to camera-setup changes has not been studies before. Experimental results show that the method yields location estimation results with error ratios smaller than 5% in most cases, which means that the proposed method is practical for applications like robot or vehicle guidance. Future studies may be directed to applying the proposed method to more application fields, as well as

Table 6.5 Error ratios with camera at ceiling height 200cm for different tilted angle 90° (looking down), 70°, and 50°.

real x,y (*cm*)	Titled for 90°		Titled for 70°		Titled for 50°	
	estimated (*x,y*) (c*m*)	*error ratio* (*type-1, type-2*)	*estimated* (*x,y*) (cm)	*error ratio* (*type-1, type-2*)	*estimated* (*x,y*) (cm)	*error ratio* (*type-1, type-2*)
(-7,-24)	(-8,-23)	(0.7%,0.4%)	(-7, -22)	(1.0%,0.6%)	(-7,-22)	(1.0%,0.6%)
(-37,36)	(-36 36)	(0.5%,0.3%)	(-37,52)	(1.5%,0.9%)	(-38,34)	(1.1%,0.7%)
(-20,96)	(-20,94)	(0.9%,0.6%)	(-22,94)	(1.3%,0.9%)	(-21,90)	(2.7%,1.9%)
(-45,-107)	(-44,-106)	(0.6%,0.4%)	(-45,103)	(1.7%,1.2%)	(-42,-103)	(2.2%,1.6%)
(-111,-55)	(-112,-56)	(0.6%,0.4%)	(-115,-58)	(2.1%,1.6%)	(-110,-60)	(2.2%,1.6%)
(-140,62)	(-140,60)	(0.8%,0.6%)	(-144,66)	(2.2%,1.8%)	(-141,58)	(1.6%,1.3%)
(-229,-101)	(-228,-104)	(1.0%,1.0%)	(-224,-95)	(2.4%,2.4%)	(-238,-110)	(4.0%,4.0%)
(-253,76)	(-257,82)	(2.2%,2.3%)	(-259,82)	(2.6%,2.6%)	(-271,81)	(4.5%,4.6%)
(-320,-15)	(-317,-15)	(0.8%,0.9%)	(-330,-16)	(2.7%,3.1%)	(-340,-26)	(6.0%,7.1%)
average error *ratio* (*type-1, type-2*)		(0.9%,0.7%)		(1.9%,1.7%)		(2.8%,2.6%)

extending the method to outdoor environments. Modifications of the method for more complicated camera structures like omni-camera pairs [16] or two-mirror omni-cameras [63] are also worth investigations.



(a) (b)

Fig. 6.14 Illustrative images of applications of proposed location estimation method for autonomous vehicle guidance in an indoor environment (a laboratory where this study was conducted). (a) An image acquired by a downward-looking camera affixed on ceiling. (b) A processed image in which autonomous vehicle center (white point) was detected for vehicle location estimation.

Chapter 7
Unwarping of Images Taken by Misaligned Omni-cameras without Camera Calibration by Curved Quadrilateral Morphing Using Quadratic Pattern Classifiers

7.1 Idea of Proposed Method

As mentioned previously, omni-cameras are getting popular for various applications owing to their advantage in providing greater FOVs in acquired omni-images. A dioptric omni-camera captures incoming light through the camera lens to form images. An example is the fish-eye camera [13]. A catadioptric omni-camera has, in addition to a CCD camera, a reflective mirror, and captures indirect light reflected by the mirror to form images. The mirror surface may be of various shapes, like conic, parabolic, hyperbolic, spheric, etc. A catadioptric omni-camera with a parabolic mirror used in this study is shown in Fig. 1.1(a), in which a transparent plastic hollow cylinder is used to support the mirror at a distance from the CCD camera placed on a platform. The structure of the camera is illustrated in Fig. 7.1(a). If all the reflected light rays go through a *common point*, the camera is said additionally to have a *single-viewpoint* (*SVP*) [64]; otherwise, a *non-single-viewpoint* (*non-SVP*) [3].

Omni-images, though providing wider FOVs, are *warped* in nature. In many applications, it is necessary to transform them into *unwarped images*. Such an *image unwarping* work usually involves camera calibration, in which the intrinsic and extrinsic camera parameters are estimated, followed by the derivation of equations to transform image coordinates into unwarped versions [71]. The camera calibration process, presumably conducted in the camera manufacturing process, is in general complicated and time-consuming. After a calibrated camera is equipped in an application environment (e.g., installed on a vehicle, attached on a house ceiling, etc.) and used for application purposes, it is usually assumed that the camera structure is fixed stably forever, incurring no change of the camera parameters.

However, in real applications like vision-based autonomous vehicle navigation or security surveillance, a camera equipped on a vehicle might be shaken due to vehicle vibrations or one installed on a wall might be removed due to re-employment, causing possibly camera misalignment as mentioned previously, which causes displacements or/and re-orientations of the CCD camera with respect to the reflective mirror. The previously-mentioned non-SVP property is actually a type of camera misalignment with both the optical axis through the lens center and the mirror axis through the mirror center being *axially displaced* with respect to each other, resulting in destruction of the SVP into a locus called a *caustic surface* [3]. We will call such a kind of camera structure change *axial-directional* camera misalignment. An illustration is shown in Fig. 1(b). Note that usually the optical axis is assumed to be coincident with the mirror axis and that the distance of the CCD camera to the mirror surface is usually adjusted properly in advance to form the SVP property.

Another type of camera misalignment is *re-orientation* of the CCD camera with respect to the mirror surface, resulting in destruction of the coincidence of the optical axis with the mirror axis. Such misalignment, seldom studied, not only destructs the SVP property [34] but also the *rotational invariance property* in omni-images, used by almost all existing image unwarping methods to simplify computation [71]. We will call such a kind of camera structure change *lateral-directional camera misalignment*. An illustration is shown in Fig. 7.1(c). Note that the rotational invariance property says that the angle of an incoming light ray of a scene point is identical to that of the corresponding image point in the image space. An image taken by a correctly-aligned catadioptric omni-camera and another taken by a lateral-directionally misaligned one are shown in Figs. 3(a) and 3(b), respectively, for illustration.

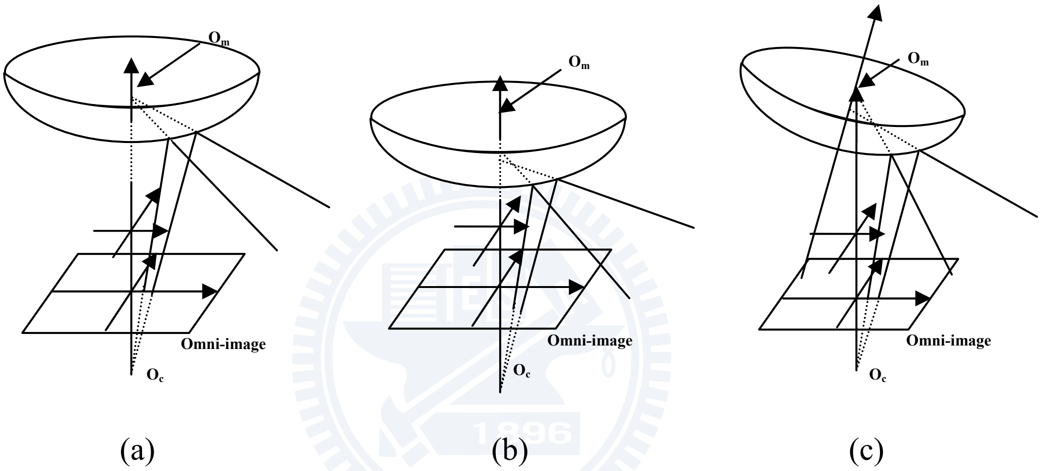


Fig. 7.1 Alignment of catadioptric omni-camera. (a) Correct alignment. (b) Axial-directional misalignment. (c) Lateral-directional misalignment.

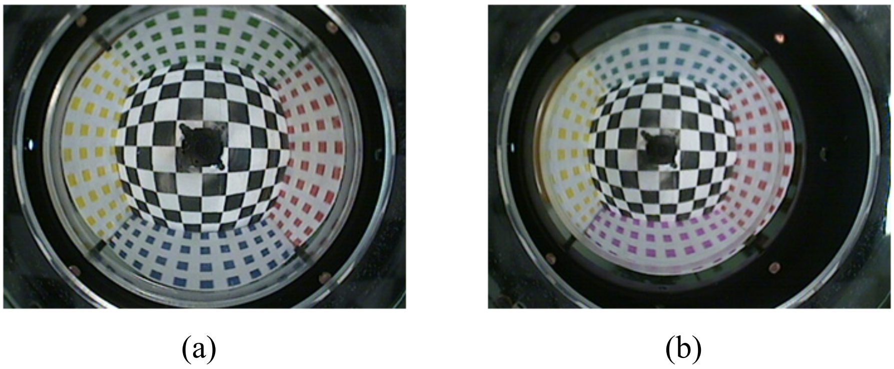


Fig. 7.2 Images of a color pattern acquired by a catadioptric omni-camera. (a) Image taken with the camera correctly-aligned. (b) Image taken the camera misaligned.

Camera misalignment causes conventional image unwarping methods inapplicable because of the resulting changes of the camera parameters. It is desired in this study to design a general image unwarping method which can solve this problem *in the application environment without camera calibration* which is usually done in the factory. Such an *in-field* method is useful for applications where sending misaligned cameras back to factories for re-calibration is undesirable or impractical.

For this goal, the idea of a *mapping-based* approach proposed recently by Jeng and Tsai [36] is adopted. This approach does not conduct camera calibration to estimate camera parameters, but creates a so-called *pano-mapping table* as a substitute of camera parameters for image unwarping. It is *unified* and *integrated* in nature, applicable to unwarping of images taken by any type of CCD camera as well as any type of reflective mirror surface.

More specifically, the proposed method has two stages, the first being assumed to be conducted in the factory and the second in the in-field environment. In the first stage, it assumed that the camera is correctly aligned to take images, which we call *undistorted* images. A pano-mapping table is then created according to Jeng and Tsai [36], which defines a coordinate mapping function from the real-world space to the omni-image space. It can be used to unwarp an omni-image into a panoramic or a perspective-view image. In the second stage where the camera is lateral-directionally misaligned, a *distortion correction table* is created first, which maps undistorted images to *distorted* ones taken by the camera. The table then is combined with the pano-mapping table to create a composite mapping from the real-world space to the distorted image space, in the form of a third table, called *misalignment adjustment table*. Such a table is finally used for unwarping distorted images into panoramic images in the real-world space.

In generating the distortion correction table which is essentially an image mapping between patches of a distorted image and those of an undistorted one, a new *image morphing* technique proposed in this study is applied. The technique is based on the use of the quadratic classifier in pattern recognition theory for two-class pattern classification. The use of such quadratic classifiers improves the precision of the morphing result of the conventionally-adopted bilinear mapping technique, because the corresponding patches in this study have *curved* boundaries instead of linear ones. Furthermore, the misalignment adjustment table is *invariant* in nature with respective to the camera position, so that the table is applicable wherever the camera is moved.

In the remainder of this chapter, we describe the proposed two-stage mapping-based image unwarping method as an algorithm in Section 7.2, present the proposed image patch morphing technique using quadratic classifiers in Section 7.3, show some experimental results in Section 7.4, and make concluding remarks finally in Section 7.5.

7.2 Proposed Mapping-based Image Unwarping Method

In this section, Jeng and Tsai's method [36] used in the proposed image unwarping method are reviewed first, followed by the description of the proposed method.

A. Review of a mapping-based image unwarping method

The *pano-mapping table* proposed by Jeng and Tsai [36] is created *once forever* by a simple learning process for a *non*-lateral-directionally misaligned omni-camera with any type of reflective mirror surface as a summary of the information conveyed by all the camera parameters. The learning process takes as input a set of landmark points on a *calibration object* in the world space and the set of corresponding points in a given image. For example, as illustrated in Fig. 7.3, P_1 and P_2 are two landmark points in the real world, and p_1 and p_2 are the corresponding image points, respectively. More generally, let the coordinates of each real-world point P be denoted as (θ, ρ), and those of its corresponding image pixel p as (u, v). The pair (θ, ρ) describes the azimuth angle and the elevation angle of an incident light ray coming from P and reflected by the mirror surface to go through the lens center, yielding the corresponding image pixel at (u, v) on the image plane. Accordingly, the pano-mapping table is designed to be 2D in nature with the horizontal and vertical axes specifying the possible ranges of θ and ρ in M and N increments, respectively, as shown in Table 7.1. Each entry E_{ij} with indices (i, j) in the table specifies a pair (θ_i, ρ_j), which defines an infinite set S_{ij} of real-world points on the light ray with azimuth angle θ_i and elevation angle ρ_j. These real-world points in S_{ij} are all projected onto an identical pixel p_{ij} in an omni-image taken by the camera, forming a *pano-mapping*, denoted as f_{pm}, from S_{ij} to p_{ij}. An illustration of this mapping is shown in Fig. 7.4. This mapping is shown in the table by filling entry E_{ij} with the coordinates (u_{ij}, v_{ij}) of pixel p_{ij} in the omni-image.

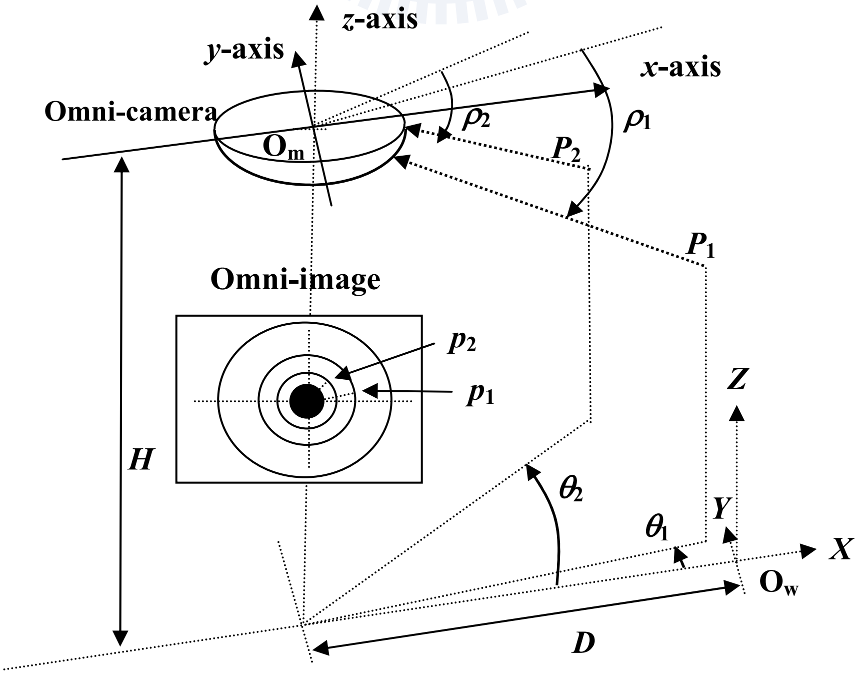


Fig. 7.3 Omni-camera system.

Table 7.1 A pano-mapping table of size $M \times N$.

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

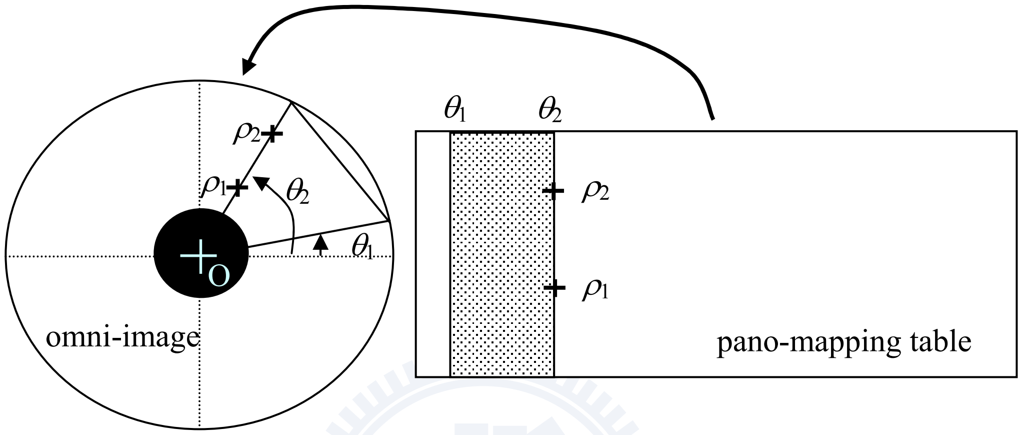


Fig. 7.4 Mapping between pano-mapping table and omni-image.

Under the assumption of correct camera alignment which leads to the rotational invariance property, Jeng and Tsai [36] derived the following equations for computing the values (u_{ij}, v_{ij}) of each entry in the table:

$$\theta_i = i \times (2\pi/M), \text{ for } i = 0, 1, \ldots, M - 1;$$
$$\rho_j = j \times [(\rho_e - \rho_s)/N] + \rho_s, \text{ for } j = 0, 1, \ldots, N - 1;$$
$$r_j = f_{\mathrm{r}}(\rho_j) = a_0 + a_1 \times \rho^1 + a_2 \times \rho^2 + a_3 \times \rho^3 + a_4\rho^4;$$
$$u_{ij} = r_j \times \cos\theta_i;$$
$$v_{ij} = r_j \times \sin\theta_i$$

where ρ_e and ρ_s specify the maximum and the minimum of the elevation angles of the omni-camera, respectively; $f_{\mathrm{r}}(\rho)$ is a nonlinear function specifying the relation between the elevation angle ρ of a real-world point P and the radial distance r from the corresponding image pixel p at coordinates (u, v) in the omni-image to the image center; and the coefficients a_0 through a_4 of $f_r(\rho)$ are estimated using the image and real-world coordinate data of the previously-mentioned corresponding landmark point pairs. Note that the rotational invariance property notationally means the azimuth angle θ of each real-world point P on the light ray is identical to the angle ϕ of the corresponding image pixel p with respect to the u-axis in the input image. That is, the azimuthal mapping is just an *identity function* f_a such that $f_a(\theta) = \phi = \theta$.

As illustrated laterally in Fig. 7.5, with the pano-mapping table T_{pm} generated as above and a given omni-image G, a panoramic image Q of size $M_Q \times N_Q$ with height H at distance D from the omni-camera may generated by mapping first each image pixel q_{kl} in Q at coordinates (k, l) to an entry E_{ij} in T_{pm} filled with coordinates (u_{ij}, v_{ij}) using the parameters M_Q, N_Q, D, and H, followed by assigning the color value of the image pixel p_{ij} of G at (u_{ij}, v_{ij}) to pixel q_{kl}. The formulas for computing the indices i and j in this process are as follows:

$$i = k \times \frac{M}{M_Q}; \quad H_q = l \times \frac{H}{N_Q}; \quad \rho_q = \tan^{-1}(\frac{H_q}{D}); \quad j = \frac{(\rho_q - \rho_s) \times N}{(\rho_e - \rho_s)}$$

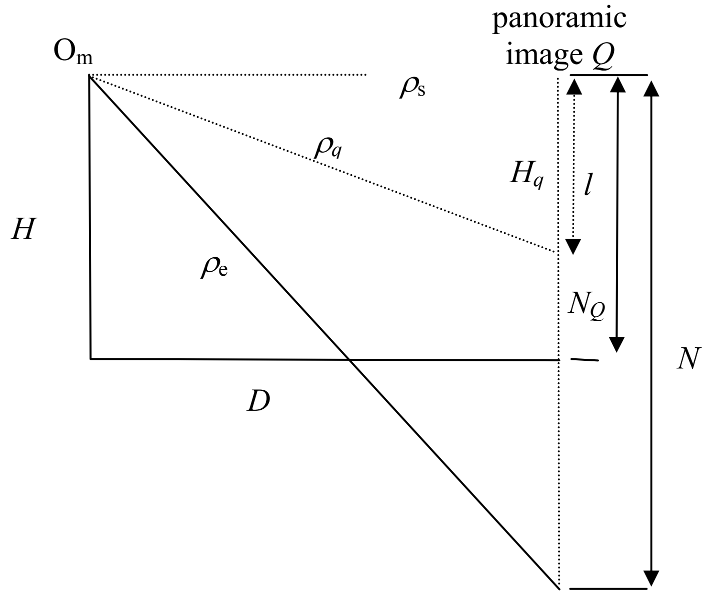
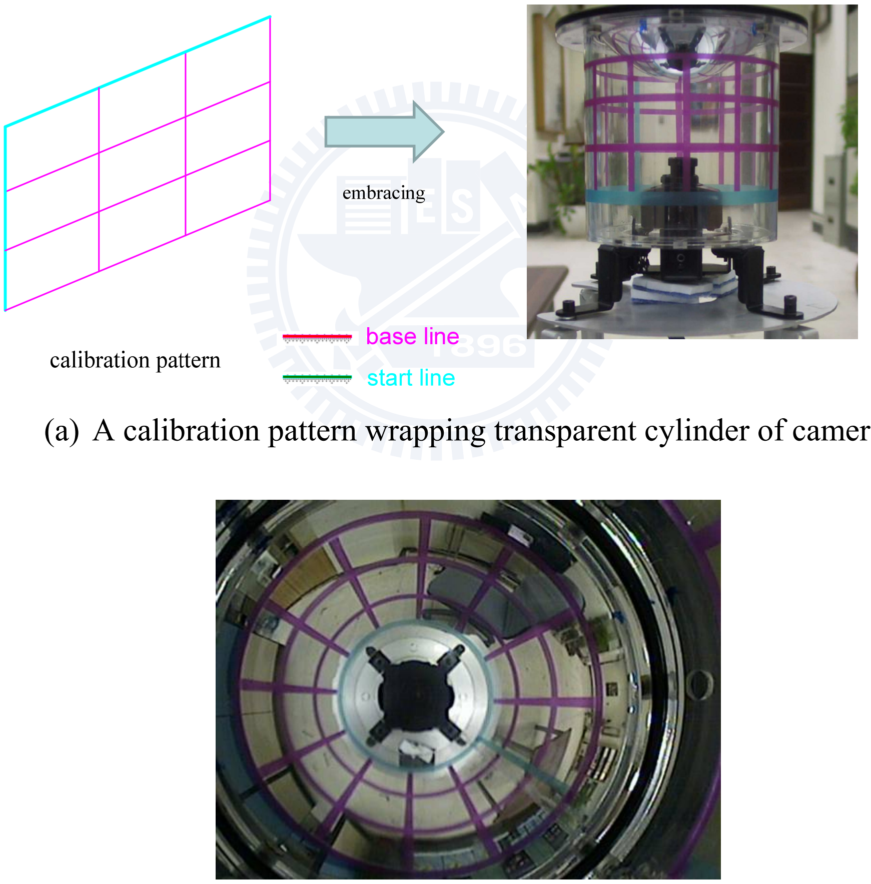


Fig. 7.5 Lateral-view configuration for generating a panoramic image.

B. Proposed method for unwarping images taken by a misaligned omni-camera

When the omni-camera is lateral-directionally misaligned as illustrated in Fig. 7.1(c), the taken image is *distorted* with respect to the undistorted image taken with a correctly-aligned or axially-misaligned omni-camera as illustrated in Figs. 1(a) and 1(b), respectively. For the above Jeng and Tsai method [36], which essentially is the pano-mapping f_{pm}: $(\theta_q, \rho_q) \rightarrow (u_{ij}, v_{ij})$, to be applicable, the distorted image coordinates (u_{ij}', v_{ij}') must be corrected in advance. From the viewpoint of image unwarping, each undistorted image pixel has a corresponding distorted one, so that there exists a *d̲istortion-m̲apping* function f_{dm}: $(u_{ij}, v_{ij}) \rightarrow (u_{ij}', v_{ij}')$, resulting in a composite mapping $f_{\text{dm}} \circ f_{\text{pm}}$: $(\theta_q, \rho_q) \rightarrow (u_{ij}, v_{ij}) \rightarrow (u_{ij}', v_{ij}')$, or integrally, f_{ma}: $(\theta_q, \rho_q) \rightarrow (u_{ij}', v_{ij}')$ as the overall solution to the image unwarping problem investigated in this study, where $f_{\text{ma}} = f_{\text{dm}} \circ f_{\text{pm}}$. The function f_{ma} will be called *misalignment adjustment function*.

One way to construct the *nonlinear* distortion-mapping function f_{dm} is to decompose the involved image part *piecewisely* into very small patches so that the resulting subimage mappings become approximately *linear*. This requires implicitly the creation of a lot of feature points in the involved image part for use in the image decomposition. Instead of adopting this linearization technique, the solution proposed in this study allows the subimages to be processed *nonlinearly*. Such subimages come from the segmentation of the image of a *calibration pattern* designed for use in this study, consisting of parallel straight lines and attached on the transparent cylinder of the camera, as shown in Fig. 7.6(a). Each subimage is a “fan-shaped” curved quadrilateral appearing in image part of the calibration pattern, as shown in Fig. 7.6(b).



(a) A calibration pattern wrapping transparent cylinder of camera.

(b) Image of calibration pattern consisting of “fan-shaped” curved quadrilaterals.

Fig. 7.6 Configuration of an omni-camera wrapped with a calibration pattern.

The calibration pattern consists of two sets of *calibration lines*, one set horizontal and the other set vertical. The lines are drawn in two colors: blue and purple. The blue ones, called *start lines*, are provided for facilitating line correspondence. The proposed image unwarping procedure are described in the following.

Algorithm 7.1 Creation of misalignment adjustment table and image unwarping.

Stage 1. Generation of a pano-mapping table in the factory for a non-lateral-directionally misaligned omni-camera.

Step 1. Wrap the transparent cylinder of the camera with the calibration pattern O_c shown in Fig. 7.6(a), take an image of O_c as a *reference image*, and denote it as I_o.

Step 2. Apply Jeng and Tsai [36] to I_o to yield a pano-mapping table with mapping function f_{pm}.

Stage 2. Generation of a misalignment adjustment table and unwarping of input distorted omni-images in the field.

Step 3. Wrap the transparent cylinder of the omni-camera, already lateral-directionally misaligned, with the calibration pattern O_c, take an image of O_c as a *working image*, and denote it as I_w.

Step 4. (*Calibration line correspondence*) Perform the following steps to find corresponding calibration lines in I_o and I_w.

4.1 Segment the calibration lines in images I_o and I_w (appearing as curves), based on the color information (blue and purple) and the edge strengths of the lines --- classify a pixel with a sufficiently large weighted sum of its color and edge values as belonging to a calibration line.

4.2 Find corresponding horizontal and vertical calibration lines in I_o and I_w respectively by numbering the lines starting from the blue start lines ---

decide two lines numbered the same from two corresponding blue start lines as a corresponding pair.

Step 5. (*Curved Quadrilateral correspondence*) Perform the following steps to find corresponding curved quadrilaterals in I_o and I_w, like those illustrated in Fig. 7.7.

5.1 Cut each corresponding horizontal calibration line pairs in I_o and I_w into corresponding curve segments using the intersection points of each horizontal line with *all* the vertical calibration lines.

5.2 Find in order every corresponding curved quadrilateral pair in I_o and I_w by use of the corresponding curve segments.

Step 6. (*Curved Quadrilateral morphing*) For each pair of corresponding curved quadrilaterals, find *corresponding points* between them with a quadratic classification scheme (described later in the next section), resulting in a *quadrilateral-morphing function*.

Step 7. (*Creation of a distortion-mapping function*) Perform the following steps to create a distortion-mapping function.

7.1 Collect all the quadrilateral-morphing functions to create a *distortion-mapping function* f_dm, which maps the coordinates of I_o to those of I_w.

7.2 Compose f_dm and f_pm to create a *misalignment adjustment function* $f_\mathrm{ma} = f_\mathrm{dm} \circ f_\mathrm{pm}$ in the form of a table T_ma, called *misalignment adjustment table*, as the desired mapping from the real-world space to the distorted image space.

Step 8. (*Unwarping input distorted omni-images*) Perform the following steps to unwarp input distorted images into panoramic ones.

8.1 Remove the calibration pattern O_c from the omni-camera and take an image I_f.

8.2 Define a panoramic image I_p to be generated and compute the azimuth angle θ and elevation angle ρ for each pixel P in I_p according to the posture of I_p.

8.3 Acquire from the misalignment adjustment table T_ma the coordinate pair (u', v') at the entry indexed by the pair (θ, ρ).

8.4 Fill the pixel P in I_p with the color value of the pixel at coordinates (u', v') in I_f.

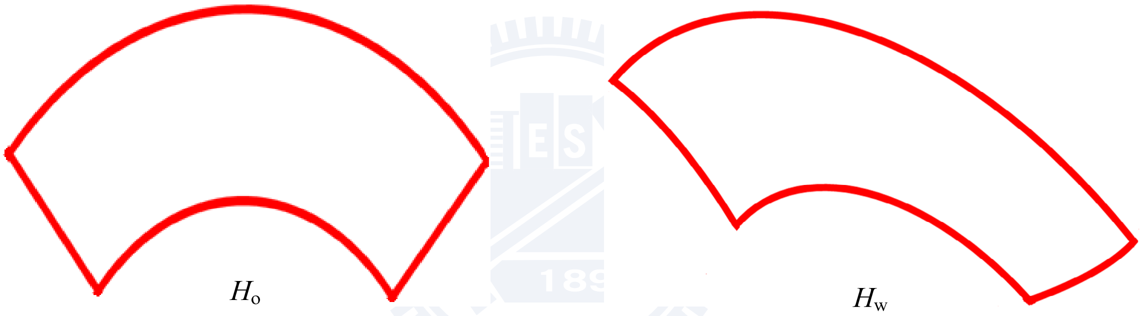


Fig. 7.7 Curved quadrilaterals forming a mutual corresponding pair.

The misalignment adjustment table created in Step 7 of the above algorithm is designed to be of the same form as that of the pano-mapping table as shown in Table 6.1, except that the table entries are filled with “distorted” coordinates (u', v') of the working image and that the index for the entries is the coordinate pair (u, v) of the reference image, like Table 7.2.

7.3 Curved Quadrilateral Morphing Using Quadratic Classifiers

In this section, the idea of the proposed technique for curved quadrilateral morphing using quadratic classifiers mentioned in Step 6 of Algorithm 7.1 is presented first. Then the adopted quadratic classification technique [73] is reviewed, followed by a detailed description of the proposed curved quadrilateral morphing technique as an algorithm.

A. Basic idea

A curved quadrilateral in the reference image I_{o} or in the working image I_{w} is a region enclosed by four curve boundaries. Fig. 7.7 illustrates two corresponding curved quadrilaterals H_{o} and H_{w} from I_{o} and I_{w}, respectively. The goal of curved quadrilateral morphing is to find corresponding pixels in H_{o} and H_{w} for use in the distortion mapping function mentioned Step 7 in Algorithm 7.1.

Table 7.2 A misalignment adjustment table of size M×N.

	v_1	v_2	v_3	v_4	…	v_M
u_1	(u_{11}', v_{11}')	(u_{21}', v_{21}')	(u_{31}', v_{31}')	(u_{41}', v_{41}')	…	(u_{M1}', v_{M1}')
u_2	(u_{12}', v_{12}')	(u_{22}', v_{22}')	(u_{32}', v_{32}')	(u_{42}', v_{42}')	…	(u_{M2}', v_{M2}')
u_3	(u_{13}', v_{13}')	(u_{23}', v_{23}')	(u_{33}', v_{33}')	(u_{43}', v_{43}')	…	(u_{M3}', v_{M3}')
u_4	(u_{14}', v_{14}')	(u_{24}', v_{24}')	(u_{34}', v_{34}')	(u_{44}', v_{44}')	…	(u_{M4}', v_{M4}')
...	…	…	…	…	…	…
u_N	(u_{1N}', v_{1N}')	(u_{2N}', v_{2N}')	(u_{3N}', v_{3N}')	(u_{4N}', v_{4N}')	…	(u_{MN}', v_{MN}')

Since the boundaries of the curved quadrilaterals here are all curves, the usual way of bilinear transformation for morphing quadrilaterals with line boundaries [65] is inapplicable here. It is desired to generate *interpolating curves* between the two opposite curves of each boundary pairs, with the interpolating curves dividing each boundary curve into equal-distanced segments, as illustrated by Fig. 7.8. That is, for example, the curve segments a, b, c, and d of the upper boundary of the quadrilateral resulting from such boundary division in Fig. 7.8 are all of equal lengths, and so are the four curve segments *e*, *f*, *g*, and *h* of the left boundary of the quadrilateral. Furthermore, the curve segments formed by mutual intersections of the interpolating curves within the four boundaries also all have this equal-lengthed property.

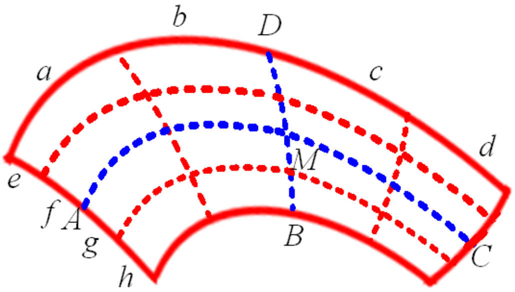


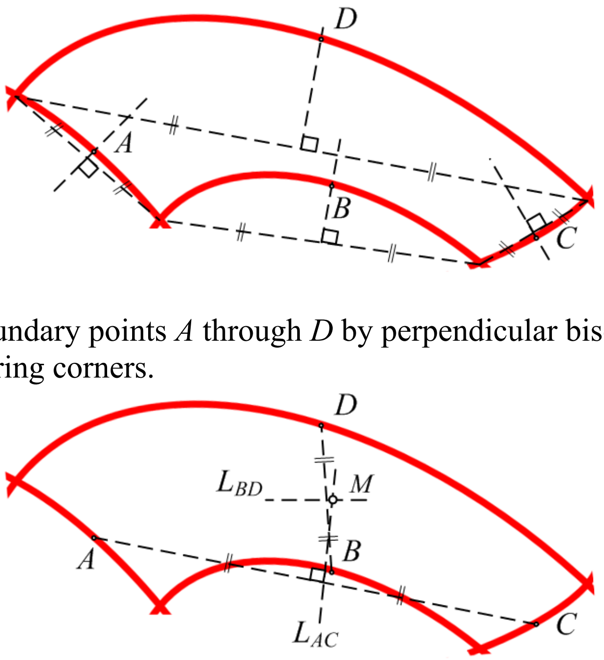
Fig. 7.8 Illustration of a curved quadrilateral with boundaries and interpolating curves segmented into equal-lengthed segments (*a* through *d* are all of equal lengths; *e* through *h* are similar, and so on).

We propose to accomplish the above idea of interpolating curve generation in a recursive manner instead of directly dividing each boundary curve into a number of equal-lengthed segments. That is, we divide *recursively* each of every pair of corresponding curved quadrilaterals into smaller *quarter ones* and consider the centers of the resulting quarter curved quadrilaterals as *corresponding points* in the original corresponding curved quadrilaterals. For this purpose, we try to find the middle point of each boundary curve, resulting in two pairs of "opposite" middle boundary points. For example, in Fig. 7.8 the two pairs are (*A*, *C*) and (*B*, *D*). The center of a curved quadrilateral is defined to be the point at equal distances to the two middle boundary points in each pair. This point in Fig. 7.8 is *M*. It will be called the *central quadrilateral point* in the sequel.

To find the middle boundary point of a curve boundary *V* of a curved quadrilateral, say, with two end points *E* and *F*, we adopt an approximation method as follows: (1) connect *E* and *F* into a line segment $\overline{EF}$; (2) find the bisecting point *G* of $\overline{EF}$; (3) find the line L_{EF} going through G and perpendicular to $\overline{EF}$, and call it the *perpendicular bisecting line of E and F*; and (4) find the intersection point S of L_{EF} and V as the desired middle boundary point. Note that S is at *equal* distances to A and C because of the bisection and perpendicularity property of L. The point S found in this way is just an *approximation* of the real middle boundary point, but it will become more accurate when the boundary curve is more symmetric with respect to the perpendicular bisecting line, as can be easily figured out. For our study here, since the distortion owing to camera misalignment is usually not too serious, this approximation is within allowable tolerance according to our experimental experience. An illustration of finding the middle boundary points of a quadrilateral is shown in Fig. 7.9(a).

To find the central quadrilateral point, say, for the case shown in Fig. 7.9(a), we adopt another approximation process which finds the perpendicular bisecting line L_{AC} of the middle boundary points A and C as well as the perpendicular bisecting line L_{BD} of the middle boundary points B and D, and then compute the intersection point M of L_{AC} and L_{BD} as the desired result. An illustration of the result of this process for Fig. 7.9(a) is shown in Fig. 7.9(b).

After the central quadrilateral point is found, the next step is to cut the original curved quadrilateral into four quarter ones. For this purpose, we have to find the curves which enclose each quarter curved quadrilateral. Such curves should go through the central quadrilateral point, as illustrated by the two blue curves in Fig. 7.8. To find such interpolating curves, we adopt the quadratic classification technique used in pattern recognition theory, as reviewed in the following.



(a) Finding middle boundary points *A* through *D* by perpendicular bisecting lines of every two neighboring corners.

(b) Finding central quadrilateral point *M* by perpendicular bisecting line of *A* and *C*, and that of *B* and *D*.

Fig. 7.9 Illustrations of finding the central quadrilateral point M in a curved quadrilateral.

B. Review of quadratic classification technique

The design of a two-class quadratic classifier in pattern recognition takes as input two sets of patterns and draws a curve as the decision boundary in the pattern space to separate the patterns into two classes in the sense of minimizing the Bayes probability of erroneously assigning the patterns into wrong classes. A good property of the classifier is its capability to generate a quadratic boundary curve which takes into consideration the *shapes* of the two pattern sets. That is, the decision boundary curve is roughly a *blending* result of the two shapes.

For our problem here, if we take each pair of "opposite" curve boundaries of a curved quadrilateral as two pattern sets with the coordinates of each boundary point as a pattern, we can design a quadratic classifier to find a decision boundary curve *going through the central quadrilateral point* as a desired interpolating curve mentioned above. An illustration is shown in Fig. 7.10, where the red coordinate data points

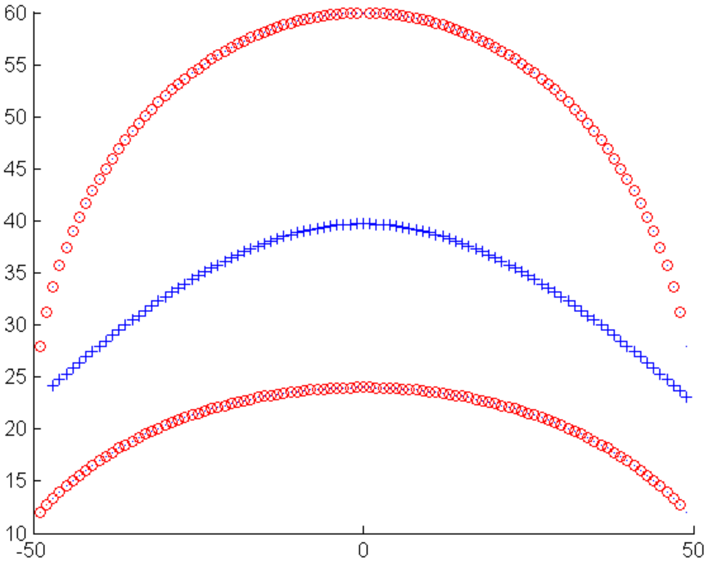


Fig. 7.10. Interpolating curve (blue) for two curve boundaries (red) found by a quadratic classifier using coordinate data as patterns (the axes specify *x*- and *y*-coordinates).

represent two simulated boundary curves of a curved quadrilateral as input pattern sets, and the blue curve is the decision boundary of a quadratic classifier designed for the two pattern sets. Comparing the shapes of the three curves, we can see the effect of shape blending mentioned previously.

Formally, a quadratic classifier for two pattern classes ω$_a$ and ω$_b$ in vector form $X = [x_1\ x_2]^T$ is as follows:

$$h(X) = X^T QX + V^T X + v_o$$

$$= \sum_{i=i}^{2}\sum_{j=1}^{2} q_{ij}x_i x_j + \sum_{i=1}^{2} v_i x_i + v_o$$

which may be transformed into a *linear form* as follows [73]:

$$h(X) = \sum_{i=i}^{3} \alpha_i y_i + \sum_{i=1}^{2} v_i x_i + v_o$$

$$= [\alpha_1\ \alpha_2\ \alpha_3\ v_1\ v_2][y_1\ y_2\ y_3\ x_1\ x_2]^T + v_0$$

$$= AZ^T + v_0$$

$$= h(Z)$$

where $Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$, $V = [v_1\ v_2]^T$, $A = [\alpha_1\ \alpha_2\ \alpha_3\ v_1,\ v_2]^T$=$[q_{11}\ q_{12}$+$q_{21}\ q_{22}\ v_1\ v_2]^T$, and $Z = [y_1\ y_2\ y_3\ x_1\ x_2]^T = [u^2 \quad uv \quad v^2 \quad u \quad v]^T$. By the linearity of $h(Z) = AZ^T + v_0$, we can use the design technique for the linear classifier to find the coefficient vector A and v_0. The result is as follows:

$$A = [sK_\text{a} + (1-s)K_\text{b}]^{-1}(D_\text{b} - D_\text{a})$$
$$v_0 = -V^T[sD_\text{a} + (1-s)D_\text{b}]$$

where s is a scaling factor between 0 and 1 for adjusting the location of the decision boundary (normally taken to be 0.5), and D_a and D_b and K_a and K_b are the means and variances of the new pattern vectors Z^a and Z^b for classes ω_a and ω_b, respectively, which are computed as follows:

$$D_\text{a} = \frac{1}{m}\sum_{i=1}^{m} Z_i^\text{a} = \frac{1}{m}\sum_{i=1}^{m}\left[(u_i^\text{a})^2 \quad u_i^\text{a}v_i^\text{a} \quad (v_i^\text{a})^2 \quad u_i^\text{a} \quad v_i^\text{a}\right]^T;$$

$$D_\text{b} = \frac{1}{n}\sum_{i=1}^{n} Z_i^\text{b} = \frac{1}{n}\sum_{i=1}^{n}\left[(u_i^\text{b})^2 \quad u_i^\text{b}v_i^\text{b} \quad (v_i^\text{b})^2 \quad u_i^\text{b} \quad v_i^\text{b}\right]^T;$$

$$K_\text{a} = \frac{1}{m}\sum_{i=1}^{m}(Z_i^\text{a} - D_\text{a})(Z_i^\text{a} - D_\text{a})^T;$$

$$K_\text{b} = \frac{1}{n}\sum_{i=1}^{n}(Z_i^\text{b} - D_\text{b})(Z_i^\text{b} - D_\text{b})^T$$

where m and n are the numbers of pattern vectors in ω_a and ω_b, respectively. By these equations, A and v_0 can be obtained and the quadratic decision boundary $h(Z) = 0$, or originally $h(X) = 0$, can be obtained.

C. Curved quadrilateral morphing

We are now ready to describe the algorithm we propose for curved quadrilateral morphing.

Algorithm 7.2 Quadrilateral morphing by quadratic classification.

Step 1. Acquire a curved quadrilateral H_w from the working image I_w, and its corresponding curved quadrilateral H_o from the reference image I_o.

Step 2. Compute the central quadrilateral points of H_w and H_o, denote them as M_c^w and M_c^o, respectively, and consider them as corresponding points.

Step 3. Design the quadratic classifier h_1^w for a pair of two opposite boundary curves of H_w so that the decision boundary curve of h_1^w goes through the central quadrilateral point M_c^w (i.e., $h_1^\mathrm{w}(M_c^\mathrm{w}) = 0$) by adjusting the scaling factor s mentioned previously. Do this similarly for the other pair of opposite boundary curves of H_w to derive another quadratic classifier h_2^w which goes through M_c^w as well.

Step 4. Use the decision boundary curves of h_1^w and h_2^w to divide the curved quadrilateral H_w into four quarter ones H_1^w, H_2^w, H_3^w, and H_4^w.

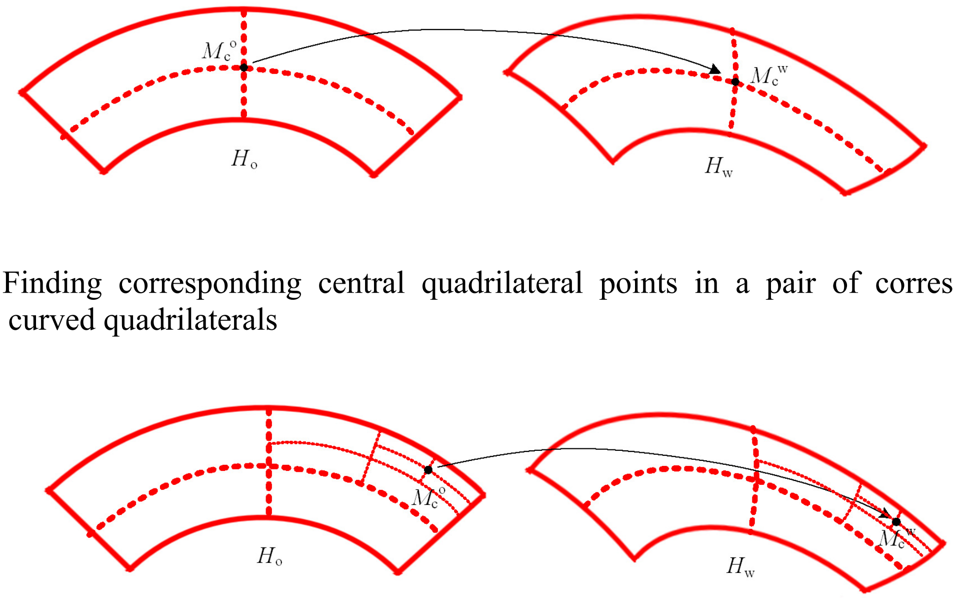
Step 5. Perform Steps 3 and 4 to the curved quadrilateral H_o similarly to cut H_o into four quarter curved quadrilaterals H_1^o, H_2^o, H_3^o, and H_4^o; and take H_i^o as the curved quadrilateral corresponding to H_i^w for $i = 1, 2, 3,$ and 4.

Step 6. For each pair of corresponding quarter curved quadrilaterals H_i^w and H_i^o for $i = 1, 2, 3,$ and 4, perform the previous steps recursively to find their respective corresponding central quadrilateral points, cut them into even smaller quarter curved quadrilaterals, and so on, until the area of any of the quarter curved quadrilaterals is smaller than a pre-selected threshold.

An illustration of the above algorithm is shown in Fig. 7.11. The result of the algorithm is a mapping from a set of the central quadrilateral points of sub-quadrilaterals of H_w to a set of the corresponding central quadrilateral points of sub-quadrilaterals of H_o. All the points are scattered in the original quadrilaterals H_w and H_o at *non-discrete* locations. For generation of a mapping between *discrete* coordinates of H_w and H_o, we apply the concept of nearest neighboring to substitute non-discrete locations with discrete ones. For example, given a point P_w in H_w with discrete coordinates (u_w, v_w), to find its corresponding point P_o in H_o with coordinates (u_o, v_o), we perform the following steps: (1) find the central quadrilateral point P_w' at non-discrete position (u_w', v_w') in H_w which is nearest to P_w; (2) find the central quadrilateral point P_o' in H_o at non-discrete position (u_o', v_o') corresponding to P_w'; and (3) find the point P_o at discrete position (u_o, v_o) in H_o which is nearest to P_o'. That is, we have the following series of mappings:

$$P_w(u_w, v_w) \rightarrow P_w'(u_w', v_w') \rightarrow P_o'(u_o', v_o') \rightarrow P_o(u_o, v_o);$$

and we take the overall mapping $(u_w, v_w) \rightarrow (u_o, v_o)$ as the final result.



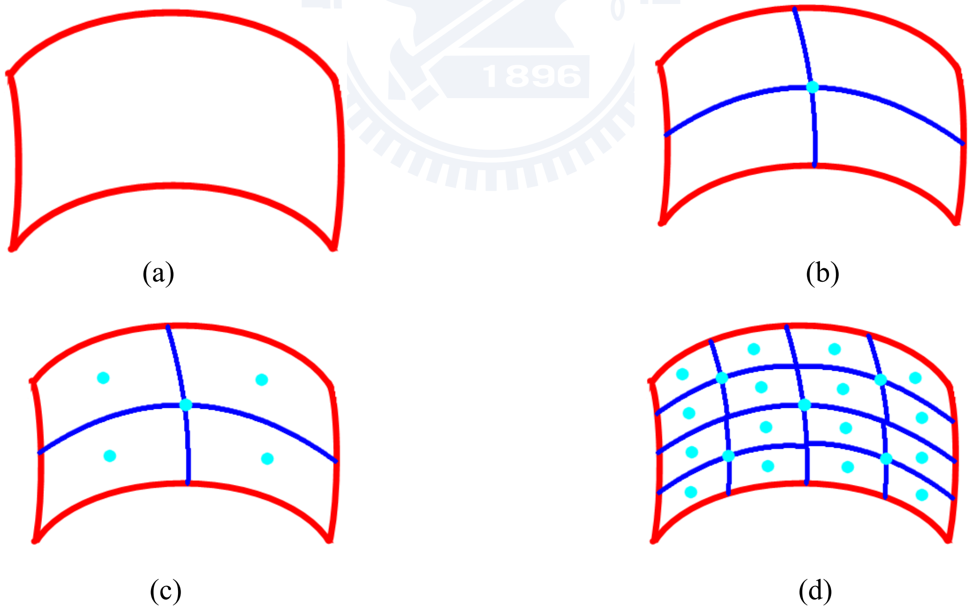
(a) Finding corresponding central quadrilateral points in a pair of corresponding curved quadrilaterals

(b) Finding more corresponding central quadrilateral points within recursively cut quarter curved quadrilaterals.

Fig. 7.11 Illustration of proposed algorithm for morphing one curved quadrilateral to a corresponding curved quadrilateral.

7.4 Experimental Results

A series of experiments have been conducted to verify the proposed method. The first experiment was conducted to test the proposed morphing algorithm (Algorithm 7.2) on some simulated data of curved quadrilaterals, which were drawn by hand and transformed into images. Fig. 7.12 shows one of the intermediate results of iterative central quadrilateral point computation, where a given input curved quadrilateral is shown in Fig. 7.12(a), and the results of the first three iterations are shown in Figs. 7.12(b) through 12(d), respectively. In each figure, the dark blue lines are the interpolating curves found by quadratic classifications, and the light blue points are the found central quadrilateral points.



(a) (b)

(c) (d)

Fig. 7.12 Results of curved quadrilateral morphing by Algorithm 7.2 using simulated data. (a) A simulated curved quadrilateral. (b) Result of 1st iteration. (c) Result of 2nd iteration. (d) Result of 3rd iteration.

In the second experiment, we use real data (a pair of curved quadrilaterals taken from real omni-images acquired by a hyperbolic catadioptric omni-camera manufactured by Micro-star International Co.) to conduct the same process of the first experiment described above. The results are shown in Fig. 7.13, with Figs. 7.13(a) through 7.13(d) corresponding respectively to Figs. 7.12(a) through 7.12(d) in meaning.

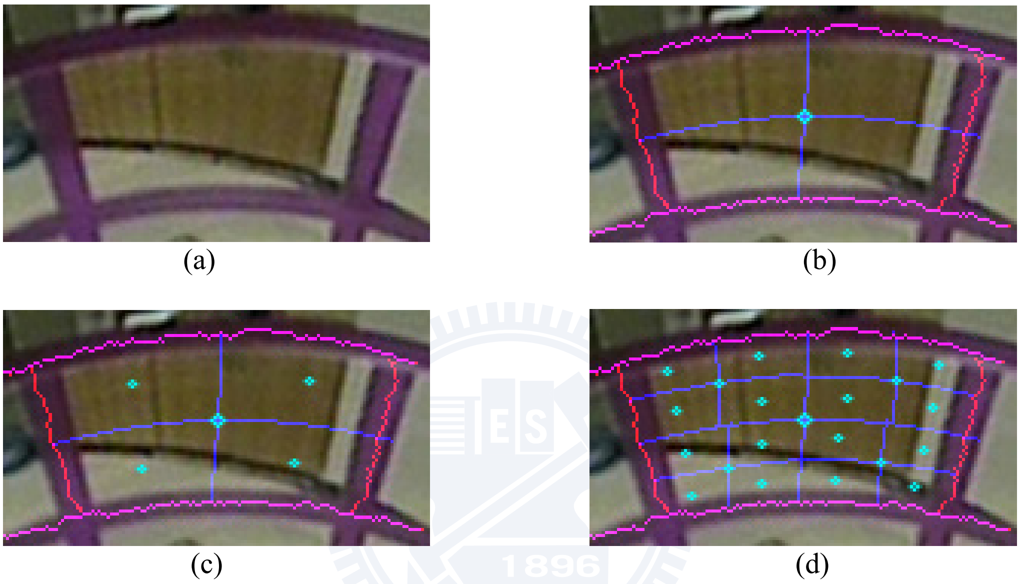


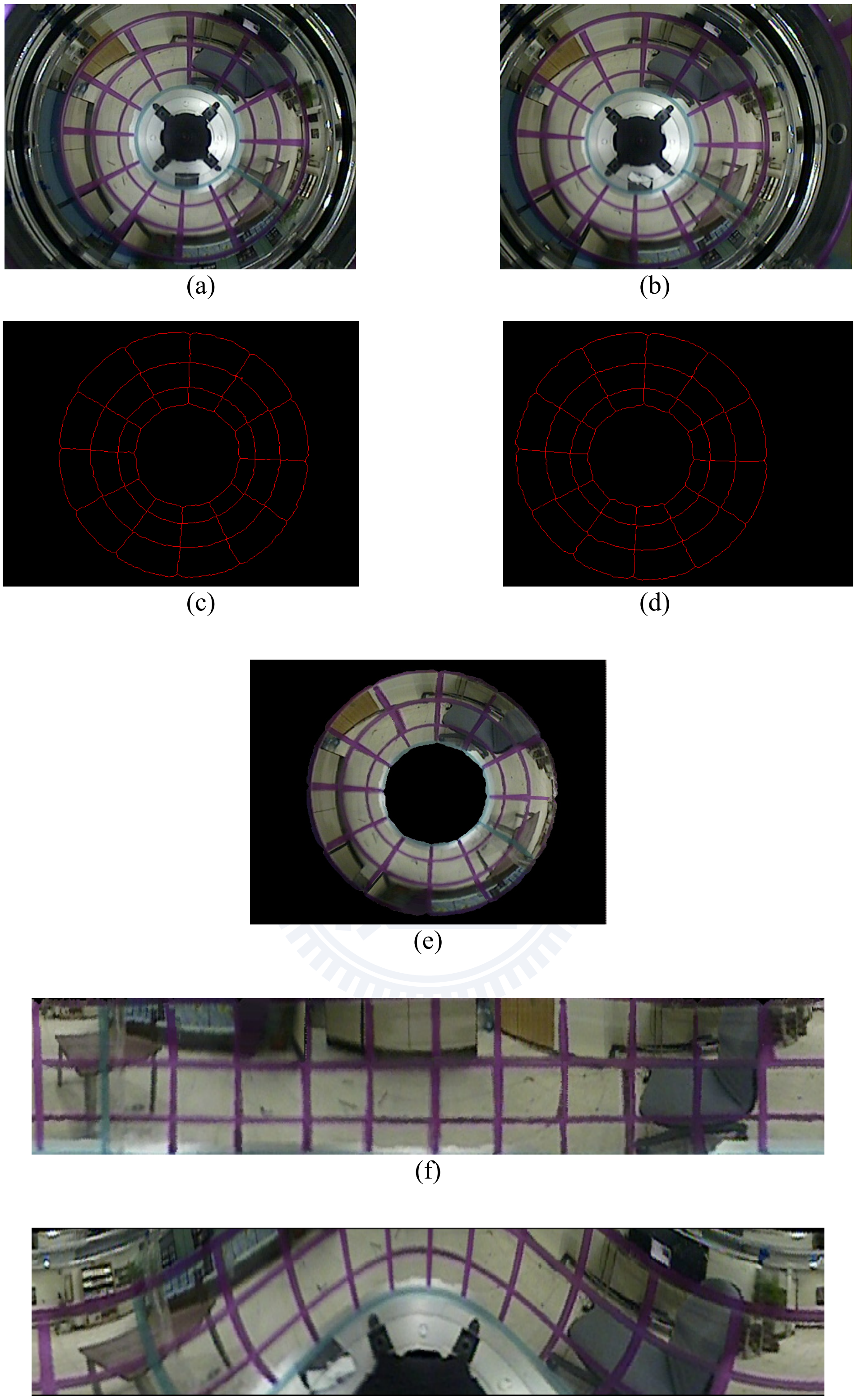
Fig. 7.13 Results of quadrilateral morphing for real data using Algorithm 7.2. (a) A curved quadrilaterals. (b) Result of 1st iteration. (c) Result of 2nd iteration. (d) Result of 3rd iteration with intermediate curves removed.

In the third experiment, we tested Algorithm 7.1 and Algorithm 7.2 together using real image data. The calibration pattern we used is as the one attached on the transparent cylinder of the omni-camera shown in Fig. 7.6(a). An undistorted image of the calibration pattern assumed to be taken in a factory for use as the reference image is shown in Fig. 7.14(a). And a distorted version of Fig. 7.14(a) taken in field using a lateral-directionally misaligned omni-camera for use as the working image is shown in Fig. 7.14(b). Figs. 7.14(c) and 7.14(d) are intermediate results which show the segmented calibration lines of Figs. 7.14(a) and 7.14(b), respectively. Using the two figures and subsequent results, a misalignment adjustment table was created by

Stage 2 of Algorithm 7.1. And Figs. 15(e) is the result of applying Algorithm 7.2 to Fig. 7.14(b) using the table. This figure was used further to generate a panoramic image, which is shown in Fig. 7.14(f).

As a contrast, we also generated a panoramic image from the distorted image of Fig. 7.14(b) without using the distortion mapping f_{dm} (i.e., using the pano-mapping f_{pm} only), and the result is shown in Fig. 7.14(g) which is quite unacceptable. This means that the proposed approach is quite significant in correcting image distortion caused by lateral-directional camera misalignment.

In the fourth experiment, we tested the effect of Algorithm 7.2 on real image data misalignment adjustment table obtained in the last experiment. A result is shown in Fig. 7.15. The input distorted image, shown in Fig. 7.15(a), was acquired from the lateral-directionally misalignment camera used in the third experiment with the calibration pattern removed. The panoramic image created from this distorted omni-image using Algorithms 7.1 and 7.2 is shown in Fig. 7.15(b). For comparison again, we create another panoramic image using the pano-mapping function only without distortion correction, and the result is shown in Fig. 7.15(c). Comparing Fig. 7.15(b) with Fig. 7.15(c), we see again that the proposed method is effect.



(a)

(b)

(c)

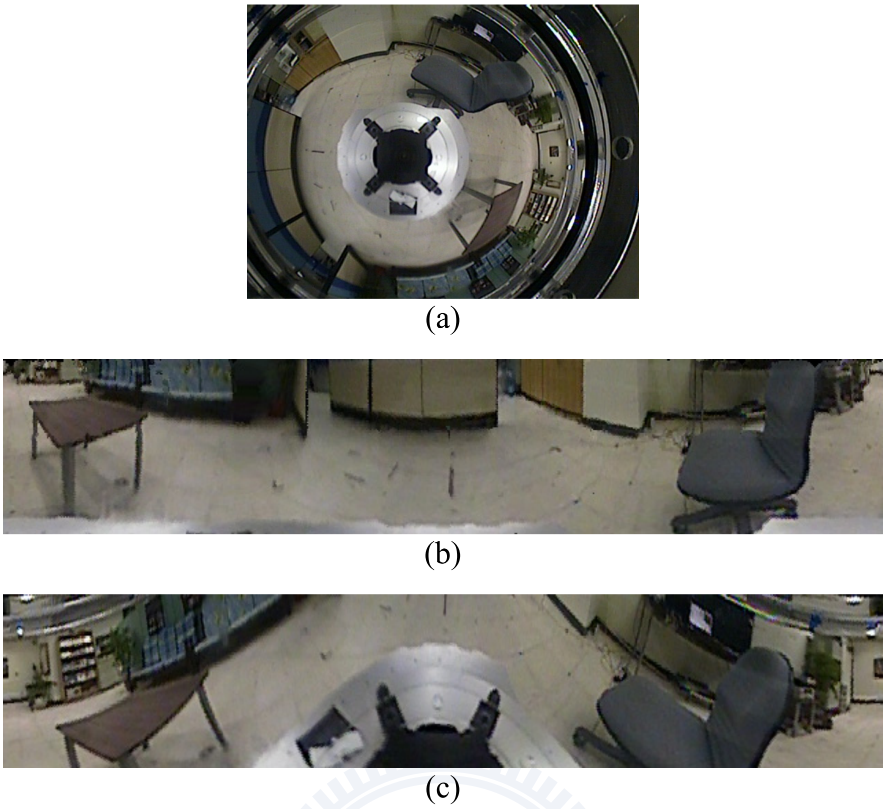
(d)

(e)

(f)

(g)

Fig. 7.14 Image unwarping results using Algorithm 7.1 and Algorithm 7.2. (a) Reference image. (b) Working image. (c) Segmented calibration lines in (a) in thinned form. (d) Segmented calibration lines in (b) in thinned form. (e) Result of applying Algorithm 7.2 to (b) using the misalignment adjustment table. (f) A panoramic image generated form (e). (g) A panoramic image generated from



(a)

(b)

(c)

Fig. 7.15 Results of distorted image unwarping using Algorithm 7.2. (a) Distorted omni-image. (b) Created panoramic image with correction by Algorithm 7.2. (c) Created panoramic image without misalignment correction.

7.5 Concluding Remarks

A method for solving the new problem of unwarping omni-images taken by lateral-directionally misaligned omni-cameras is proposed, which is based on the concept of direct pixel mapping instead of the conventional camera calibration approach. The method may be regarded as a generalization of that proposed by Jeng and Tsai [73] which is applicable to images taken by omni-cameras with no lateral-directional misalignment. The proposed method uses the misalignment adjustment table to map real-world space points to distorted image pixels in a table-lookup manner, thus speeding up the image unwarping process. The table is a composite of a distortion-mapping function defined in this study and a pano-mapping function generated by the Jeng and Tsai method. The distortion-mapping function is generated by a new technique of curved quadrilateral morphing based on quadratic classification in pattern recognition theory. Quadratic classifiers are used to generate interpolating curves within corresponding curved quadrilaterals extracted from the reference and working images. Such curves are used to generate corresponding pixels in distorted and undistorted images, which are then used for generating the content of the misalignment adjustment table. Experimental results show the feasibility of the proposed method. Future studies may be directed to investigating the possibility of deriving the misalignment adjustment table directly without combining two tables and to using the proposed curved quadrilateral morphing technique for unwarping other types of image distortion.

Chapter 8 Conclusions and Suggestions for Future Research

8.1 Conclusions

In this study, new localization and image analysis techniques based on omni-vision for various applications of autonomous vehicles, including indoor vehicle and robot guidance, helicopter landing, car driving assistance, have been proposed. Landmark used in the proposed vehicle localization techniques include:

(1) house corners, which include points, lines, and their combinations;

(2) circular shapes on ceilings;

(3) international standard helipads;

(4) car wheel shapes, etc.

Proposed new image analysis techniques for uses in the applications include:

(1) approximating irregular projections of circles in omni-images by ellipses;

(2) approximating irregular projections of lines in omni-images by conic-sectional shapes;

(3) warping distorted omni-images acquired by misaligned omni-cameras into undistorted images;

(4) adaptive adjustment of omni-images taken by posture-slanted omni-cameras into standard images, etc.

Novel combinations of the proposed vehicle localization and image analysis techniques have been proposed for various applications, including:

(1) location estimation for indoor autonomous vehicle navigation by

omni-directional vision using circular landmarks on ceilings;

(2) a systematic approach to indoor vision-based robot localization using corner features in omni images;

(3) an omni-vision based self-localization method for automatic helicopter landing on standard helipads;

(4) omni-vision based localization of lateral vehicles for car driving assistance

(5) adaptation of space-mapping methods for object location estimation to camera setup changes;

(6) unwarping of images taken by misaligned omni-cameras without camera calibration by curved quadrilateral morphing using quadratic pattern classifiers.

Good experimental results have been shown to prove the feasibility of the proposed techniques.

8.2 Suggestions for Future Research

The following topics may be investigated in the future:

1. use of vanishing point information formed by long parallel lines;
2. use of house beam information formed by mutually parallel lines;
3. use of parallel line information in an image acquired by a two-mirror omni-camera;
4. use of two perpendicular lines in an image acquired by a longitudinally-coxial omni-camera pair;
5. use of new-typed omni-cameras in various applications for more precise and effective image acquisition and analysis;
6. aplying applications of the proposed methods to other types of vehicles, such

as airplane, UAV, surveillance car, toy car, etc.;

7. exploitations of other applications of autonomous vehicles, like automatic car parking and driving, security patrolling, intelligent transportation, airplane landing, outer space planet exploration, etc.

References

[1] M. Betke and L. Gurvits, “Mobile robot localization using landmarks,” *IEEE Transactions on Robotics and Automation*, vol. 13, no.2, pp. 251-263, Apr. 1997.

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

[3] J. Gaspar, N. Winters, and J. Santos-Victor, “Vision-based navigation and environmental representations with an omni-camera,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, Dec. 2000.

[4] E. Menegatti, T. Maeda and H. Ishiguro, “Image-based memory for robot navigation using properties of the omni-images,” *Robotics and Autonomous Systems*, vol. 47, pp. 251-267, July 2004.

[5] 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.

[6] 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, June 2002, pp. 128-135.

[7] C. Becker, J. Salas, K. Tokusei, and J.-C. Latombe, “Reliable navigation using landmarks,” *IEEE Robotics and Automation Conference*, 1995, pp. 401 – 406.

[8] 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.

[9] 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.

[10] 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. (15 January 2008), pp. 31-43.

[11] 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, Aug. 23-26, 2004.

[12] 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, Nov. 1996

[13] Liu, Y. C., K. Y. Lin, and Y. S. Chen, “Bird’s-eye view vision system for vehicle surrounding monitoring,” *Proceedings of Conference on Robot Visio*n, Berlin, Germany, Feb. 20, 2008, pp. 207-218.

[14] 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.

[15] Zhihui, et al., “Catadioptric omni-directional stereo vision and its applications in moving objects detection,” in Computer Vision, Z. Xiong (ed.), Vienna, Austria, Nov. 2008

[16] H. Koyasu, J. Miura and Y. Shirai, “Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments,” *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 31-36, Maui, Hawaii, U. S. A., Oct. 29-Nov. 03, 2001.

[17] C. T. Ho and L. H. Chen, “A high-speed algorithm for elliptical object detection,” *IEEE Transactions on Image Processing*, Vol. 5, No. 3, pp. 547-550, Mar. 1996.

[18] L. Ma, Y. Chen, and K. L. Moore, “Flexible camera calibration using a new analytical radial undistortion formula with application to mobile robot localization,” *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 799-804, 2004.

[19] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no.11, pages 1330-1334, 2000.

[20] I. Shimshoni, “On mobile robot localization from landmark bearings,” IEEE

Transactions on Robotics & Automation, vol. 18, no. 6, December 2002.

[21] A. Merke, S. Welker, and M. Riedmiller, “Line based robot localization under natural light conditions,” *Proceedings of European Conference on Artificial Intelligence (ECAI) 2004-Workshop on Agents in Dynamic & Real-Time Environments*, Valencia, Spain, 2004.

[22] M. Dissanayake, P. Newman, M., S. Clark, H. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Transactions on Robotics & Automation*, vol. 17, no. 3, pp. 229-241,2001..

[23] H. L. Chou and W. H. Tsai, "A new approach to robot location by house corners," *Pattern Recognition*, vol. 19, No. 6, 1986, pp. 439-451.

[24] S. Y. Chen and W. H. Tsai, "Determination of robot locations by common object shapes," *IEEE Transactions on Robotics & Automatio*n, vol. 7, no. 1, pp. 149-156, 1991.

[25] O. Parlaktuna, T. Bilir, and A. Yazici, “Localization of a mobile robot using natural landmarks and sensor fusion,” *Proceedings of 4th International Conference on Electrical & Electronics Eng.* (*ELECO 2005*), Bursa,Turkey, pp.394-397, Dec. , 2005.

[26] S. Y. Park, S. C. Jung, Y. S. Song, and H. J. Kim, “jo4,” *Proceedings of 2008 IAPR Workshop on Cognitive Information Processing*, Santorini, Greece, June 9-10, 2008.

[27] J. P. Barreto and H. Araujo, “Fitting conics to paracatadioptric projections of lines,” *Computer Vision & Image Understanding*, vol. 101, 151–165, 2006.

[28] R. I. Hartley, “Camera calibration using line correspondences,” *Proceedings of DARPA Image. Understanding Workshop*, pp. 361-366, 1993.

[29] A. Ramisa, A. Tapus, R. L. de Mantaras, R. Toledo, “Mobile robot localization using panoramic vision and combinations of feature region detectors,” *Proceedings of IEEE International Conference on Robotics & Automation*, Pasadena, CA, USA, May 19-23, 2008, pp. 538-543.

[30] B. Siemiątkowska1 and R. Chojecki, “Mobile robot localization based on omnicamera,” *Proceedings of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisboa, Portugal, July 5-7, 2004.

[31] D. Cobzas and H. Zhang, “Mobile robot localization using planar patches and a stereo panoramic model,” *Proceedings of Vision Interface Annual Conference*,

Ottawa, Canada, pp. 94- 99.

[32] J. P. Barreto and H. Araujo, “Geometric properties of central catadioptric line images and their application in calibration,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 8, Aug. 2005.

[33] X. Ying and H. Zha, “Geometric interpretations of the relation between the image of the Absolute Conic and Sphere Images,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 28, no. 12, Dec. 2006.

[34] T. Mashita, Y. Iwai, and M. Yachida, “Calibration method for misaligned catadioptric camera,” *IEICE Transactions on Information & Systems*, vol. E89-D, no. 7, pp. 1984-1993, July 2006.

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

[36] S. W. Jeng and W. H. Tsai, "Using pano-mapping tables to unwarping of omni-images into panoramic and perspective-view Images," *IET Image Processing*, vol. 1, No. 2, pp. 149-155, June 2007.

[37] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Communication ACM*, vol. 15, no. 1, 1972 pp. 11-15.

[38] Z. F. Yang and W. H. Tsai, “Using parallel line information for vision-based landmark location estimation and an application to automatic helicopter landing,” *Robotics and Computer-Integrated Manufacturing*, vol. 14, pp. 297–306, 1998.

[39] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, “Visually-guided landing of an unmanned aerial vehicle,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp 371-381. June 2003.

[40] R. O. V. Shakernia, C. S. Sharp, Y. Ma, and S. Sastry, “Multiple view motion estimation and control for landing an unmanned aerial vehicle,” *Proceedings of 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, U. S. A., pp. 2793 - 2798, May 2002.

[41] P. Garcia-Pardo, G. Sukhatme, and J. Montgomery, "Towards vision-based safe landing for an autonomous helicopter", *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 19-29, 2001.

[42] L. Mejias, P. Campoy, S. Saripalli, and G. S. Sukhatme, “A visual servoing

approach for tracking features in urban areas using an autonomous helicopter," *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, U. S. A., pp. 2503-2508, May 2006.

[43] A. C. Tsai, P. W. Gibbens and R. H. Stone, "Visual position estimation for automatic landing of a tail-sitter vertical takeoff and landing unmanned air vehicle," *Mechatronics and Machine Vision in Practice,* J. Billingsley and R. Bradbeer (eds.) Springer-Verlag, New York, New York, U. S. A., pp. 181-191, 2008.

[44] O.A. Yakimenko, I. I, Kaminer, W. J. Lentz and P. A. Ghyzel, "Unmanned aircraft navigation for shipboard landing using infrared vision," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 1181-1200, 2002.

[45] J. M. Hespanha, O. A.Yakimenko, I. I. Kaminer, and A. M. Pascoal, " Linear parametrically varying systems with brief instabilities: an application to vision/inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 889-902, 2004.

[46] S. Hrabar and G. S. Sukhatme "Omnidirectional vision for an autonomous helicopter," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 14-19, 2003, pp. 3602-3609.

[47] *C. Demonceaux, P. Vasseur,* and C. Pegard, "Omnidirectional vision on uav for attitude computation," *Proceedings of 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, U. S. A. ,* pp. 2842 – 2847, May 2006.

[48] J. C. Bazin, I. Kweon, C. Demonceaux, and P. Vasseur, "UAV attitude estimation by vanishing points in catadioptric images," *Proceedings of 2008 IEEE International Conference on Robotics and Automation*, Pasadena, California, U. S. A. , pp. 2842 - 2847, May 2008.

[49] C. J. Wu and W. H. Tsai, "Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 546-555, May 2009.

[50] H. H. Lin and J. H. Lin "A Visual Positioning System for Vehicle or Mobile Robot Navigation," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 7, 2006, pp. 2109-2116

[51] K. Sakurai, S. Kyo and S. Okazaki "Overtaking Vehicle Detection Method and

Its Implementation Using IMAPCAR Highly Parallel Image Processor," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-D, no. 7, 2008, pp. 1899-1905.

[52] C. C. Lai and W. H. Tsai "Estimation of Moving Vehicle Locations Using Wheel Shape Information in Single 2-D Lateral Vehicle Images by 3-D Computer Vision Techniques. Robot," *Robotics and Computer-Integrated Manufacturing*, vol. 15, 1999, pp. 111-120.

[53] Y. Cao, A. Renfrew and P. Cook "Vehicle Motion Analysis Based on a Monocular Vision System," *Proc. RTIC 2008 Conference, Manchester*, UK, 2008, pp. 1-6.

[54] C. J. Wu and W. H. Tsai, Location Estimation for Indoor Autonomous Vehicle Navigation by Omni-Directional Vision Using Circular Landmarks on Ceilings, *Robotics and Autonomous Systems*, vol. 57, no. 5, 2009, pp. 546-555.

[55] O. A. Aider, P. Hoppenot, and E. Colle, "A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences," *Robotics and Autonomous Systems*, vol. 52, pp. 229-246, Aug. 2005.

[56] E. E. Hemayed, "A survey of camera self-calibration," *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 351- 357, Miami, Florida, USA, July 21-22, 2003.

[57] Z. F. Yang and W. H. Tsai, "Viewing corridors as right parallelepipeds for vision-based vehicle localization," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 3, pp. 653-661, June 1999.

[58] C. H. Ku and W. H. Tsai, "Obstacle avoidance in person following for vision-based autonomous land vehicle guidance using vehicle location estimation and quadratic pattern classifier," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 1, pp. 205-215, Feb. 2001.

[59] T. Takeshita, T. Tomizawa and A. Ohya, "A house cleaning robot system – path indication and position estimation using ceiling camera," *Proceedings of International Joint Conference on SICE-ICASE*, pp. 2653-2656, Busan, Korea, Oct. 18-21, 2006.

[60] Y. T. Wang and W. H. Tsai, "Indoor security patrolling with intruding person detection and following capabilities by vision-based autonomous vehicle navigation," *Proceedings of 2006 International Computer Symposium (ICS 2006)*

- *Workshop on Image Processing, Computer Graphics, and Multimedia Technologies*, Taipei, Taiwan, Dec. 4-6, 2006.

[61] H. C. Chen and W. H. Tsai, “Optimal security patrolling by multiple vision-based autonomous vehicles with omni-monitoring from the ceiling,” *Proceedings of 2008 International Computer Symposium,* Taipei, Taiwan, Nov. 13-15, 2008.

[62] J. Gomes, L. Darsa, B. Costa, and L. Velho, *Warping and Morphing of Graphical Objects*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1999.

[63] Z. Xiong, W. Chen and M. Zhang, “Catadioptric omni-directional stereo vision and its applications in moving objects detection,” in *Computer Vision*, Z. Xiong (ed.), Vienna, Austria, Nov. 2008

[64] S. Baker and S. K. Nayar, “A theory of single-viewpoint catadioptric image formation,” *International Journal of Computer Vision*, vol. 35, no. 2, pp. 175-196, 1999.

[65] G. Tarak and M. M. Trivedi, “Parametric ego-motion estimation for vehicle surround analysis using omni-camera,” *Machine Vision and Applications*, vol. 16, no. 2, pp. 85-95, 2005.

[66] M. Senoh, F. Kozawa, and M. Yamada, “Development of shape measurement system using an omnidirectional sensor and light sectioning method with laser beam scanning for Hume pipes,” *Optical Engineering*, vol. 45, no. 6, June 2006.

[67] T. Boult, “DOVE: Dolphin omni-directional video equipment,” *Proceedings of IASTED International Conference on Robotics and Automation*, Honolulu, Hawaii, USA, Aug. 14-16, 2000.

[68] S. K. Nayar, “Catadioptric omni-camera,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San-Juan, Puerto Rico, June 1997, pp. 482-488.

[69] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, “Visual surveillance and monitoring system using an omni-directional video camera,” *Proceedings of 14th International Conference on Pattern Recognition*, Brisbane, Australia, vol. 1, pp. 588-592, Aug. 16-20, 1998.

[70] X. H. Ying and Z. Y. Hu, “Catadioptric camera calibration using geometric invariants,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 26, no. 10, pp. 1260-1271, Oct. 2004.

[71] S. W. Jeng and W. H. Tsai, "Analytic image unwarping by a systematic calibration method for omni-cameras with hyperbolic-shaped mirrors," *Image and Vision Computing*, vol. 26, no. 5, pp. 690-701, May 2008.

[72] S. W. Jeng and W. H. Tsai, "Construction of perspective and panoramic images from omni-images taken from hypercatadioptric cameras for visual surveillance," *Proceedings of 2004 IEEE International Conference on Networking, Sensing*, *and Control*, Taipei, Taiwan, pp. 204-209, March 21-23, 2004.

[73] K. Fukunaga, *Introduction to Statistical Pattern Recognition* (2nd edition). Academic Press, New York, NY, USA, 1990.

[74] R. Swaminathan, M.D. Grossberg, S.K. Nayar, "Caustics of catadioptric cameras," *Proceedings of Eighth IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, vol. 2, July 7–14, 2001, pp. 2–9.

[75] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kauffmann Publishers, San Francisco, CA, USA, 1999.

Publication List

Journal Papers:

(1) C. J. Wu and W. H. Tsai (2009/05). "Location Estimation for Indoor Autonomous Vehicle Navigation by Omni-Directional Vision Using Circular Landmarks on Ceilings," *Robotics and Autonomous Systems,* Vol. 57, No. *5*, pp. 546-555. (SCI, EI)

(2) C. J. Wu and W. H. Tsai "Unwarping of Images Taken by Misaligned Omni-cameras without Camera Calibration by Curved Quadrilateral Morphing Using Quadratic Pattern Classifiers, " *Optical Engineering* (accepted and to appear) (SCI, EI).

(3) C. J. Wu and W. H. Tsai "A Systematic Approach to Indoor Vision-Based Robot Localization Using Corner Features in Omni Images," submitted to the *IEEE Transactions on Industrial Electronics*. (SCI, EI)

(4) C. J. Wu and W. H. Tsai " An Omni-vision Based Self-localization Method for Automatic Helicopter Landing on Standard Helipads," submitted to the *IEEE Transactions on Aerospace and Electronic Systems*. (SCI, EI)

(5) C. J. Wu and W. H. Tsai " A Space-Mapping Method for Object Location Estimation Adaptive to Camera Setup Changes for Vision-based Au-tomation Applications" submitted to *IEEE Transactions on. Circuits and Systems for Video Technology* (SCI, EI).

(6) C. J. Wu and W. H. Tsai "Omni-vision Based Localization of Lateral Vehicles for Car Driving Assistance," submitted to *Robotics and Computer-Integrated Manufacturing* (SCI, EI).

(7) C. J. Wu and W. H. Tsai "Automatic Airplane Landing by Omni-vision Based Self-localization," submitted to *IEEE Transactions on. Circuits and Systems for Video Technology* (SCI, EI)

Patents

(8) C. J. Wu, S. Y. Tsai, and W. H. Tsai. "Automatic Ultrasonic And Computer-Vision Navigation Device And Method Using The Same," *USA patent*, *No.12/365190*. (pending)

(9) C. J. Wu and W. H. Tsai(2007). “A Landmark-Based Vehicle Location System

And Its Technique,” *Republic of China Patent*, *No. I 274971*, 2007 (granted).

(10) K. F. Chien, C. J. Wu and W. H. Tsai. "A Video Surveillance System And Its Method Based On Data Hiding And Video Encoding," *Republic of China Patent*, *No. 200837661*. (pending)

(11) H. Y. Chen, M. J. Han, C. J. Wu, H. H. Lin, C. J. Kang, Y. Y. Yang, K. T, Song, W. H. Tsai, and J. H. Chuang. "A Mobile System of Image Acquisition And Its Controlling Method," *Republic of China Patent*, *No. 096127495*. (pending).

(12) C. J. Wu, S. Y. Tsai, and W. H. Tsai. ”A Method for Mobile Robot Navigation System with Simple Learning Procedures by Ultrasonic Sensing And Computer Vision Techniques,” *Republic of China Patent*, *No. 97131096*. (pending).

Conference Papers

(13) C. J. Wu and W. H. Tsai "A novel method for lateral vehicle localization by omni-cameras for car driving assistance," *Proceedings of 13th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems* (KES2009), Santiago, Chile, September 28-30, 2009.

(14) C. J. Wu and W. H. Tsai "Adaptation of Space-Mapping Methods for Object Location Estimation to Camera Setup Changes --- A New Study," *Proceedings of 13th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems* (KES2009), Santiago, Chile, September 28-30, 2009.

(15) C. J. Wu and W. H. Tsai. “Location estimation for indoor autonomous vehicle navigation by omni-directional vision using circular landmarks on ceilings,*”* *Proceedings of 2005 Conference on Computer Vision, Graphics and Image Processing*, Taipei, Taiwan, Republic of China, 21-23 Aug.,2005

Technology transfers and patent authorizations

(16) C. J. Wu, Y. T. Wang, W. H. Tsai, et al., Contract No. NCTU-05A032, Micro star international, Taiwan, April 2006.

(17) C. J. Wu, S. Y. Tsai, W. H. Tsai, et al., Contract No.NCTU-08A018, Micro star international, Taiwan, April 2008.

(18) C. J. Wu, S. Y. Tsai, W. H. Tsai, et al., Contract No. NCTU-08A051, Industrial technology research institute of Taiwan, Nov. 2008.

Vita

Chih-Jen Wu was born in Taichung, Taiwan, R.O.C., in 1978. He received the B.S. degree and the M. S. degree in engineering science from National Cheng Kung University, Tainan, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in the Institute of Computer Science and Engineering, College of Computer Science from National Chiao Tung University in 2009.

Mr. Wu worked as a research assistant in the Laboratory of System Integration in National Cheng Kung University from August 2000 to July 2002, and as a research engineer in the Computer Vision Laboratory in National Chiao Tung University since August 2002 till now. His current research interests include computer vision, robotics, pattern recognition, and their applications.