

國立交通大學

資訊科學與工程研究所

碩士論文

室內環境之三維模型重建

Three-Dimensional Surface Model
Reconstruction of Indoor Environments

研究生：張凱為

指導教授：陳永昇 教授

中華民國九十六年六月

室內環境之三維模型重建
**Three-Dimensional Surface Model
Reconstruction of Indoor Environments**

研 究 生：張凱為

Student : Kai-Wei Chang

指 導 教 授：陳永昇

Advisor : Yong-Sheng Chen



Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

摘要

從二維的影像資訊重建出三維的場景模型，一直是電腦視覺領域一個重要的研究主題，隨著電腦計算速度的進步，這項研究所延伸而出的應用更是琳瑯滿目。近年來蓬勃發展的電腦繪圖、虛擬實境等等，都會利用到影像重建的技術，比如將一些現成的玩具利用多個視角的照片，便可在電腦中產生玩具的模型。我們提出了一個透過影像來重建三維場景模型的方法，透過影像之中與影像之間的關係將攝影機的內外部參數算出之後，我們變能夠將影像間重複拍攝的部份的三維座標點算出。在算出三維座標點之後，使用Wendland將三維座標點之間缺乏的部份算出來形成場景的三維表面模型並將拍攝到的影像當作場景的材質貼到該模型上以達到擬真的室內環境重建。



致謝

本篇論文的完成，首先要感謝我的指導教授陳永昇老師，在老師熱心的協助之下，不論是在學術的研究上，或是在論文撰寫的過程中，都給了我莫大的幫助，也使得本篇論文能夠順利地完成。也謝謝口試委員莊仁輝教授及黃仲陵教授給予我的建議與指教，使得本篇論文能夠更加完善。另外，也感謝實驗室的同學和學長姊學弟妹，因為有他們的相互扶持與陪伴，使我的研究過程並不孤單。

最後，我要謝謝我的家人，因為他們的支持是我努力的最大動力，僅以此篇論文，獻給我最愛的父母親。



Three-Dimensional Surface Model Reconstruction of Indoor Environments

A thesis presented

by

Kai-Wei Chang

to

Department of Computer Science
and Information Engineering

in partial fulfillment of the requirements

for the degree of

Master of Science

in the subject of

Computer Science and Information Engineering

National Chiao Tung University

Hsinchu, Taiwan

2007

Three-Dimensional Surface Model Reconstruction of Indoor Environments

Copyright © 2007

by

Kai-Wei Chang



Abstract

In recent years computer hardware and computer graphics has made tremendous progress in visualizing 3D models of real objects. Many techniques have reached maturity and are being ported to hardware. This seems like in the area of 3D visualization, performance may increase even faster than Moor's law. Some job required a million dollar computer a few years ago can be now achieved by a custom computer, which cost a few hundred dollars. It is now possible to visualize complex 3D scenes in real time due to the advancement of computer hardware.

This speed of evolution causes an essential demand for more complex and realistic models. Even though we are now able to build three-dimensional models, the tools for three-dimensional modeling are getting more and more powerful, synthesizing realistic models is difficult and time-consuming. Many virtual objects are inspired by real objects, so we are interested in being able to build three-dimensional environment models directly from the real environments.

In the past, visual inspection and robot guidance were the main applications. We require more and more 3D content for computer graphics, virtual reality and communication nowadays. The visual quality becomes one of the main points of attention. Therefore not only the position of a small number of points have to be measured with high accuracy, but the geometry and appearance of all points of the surface have to be measured.

We proposed a semi-automatic 3D indoor environment reconstruction procedure using the thin-plate splines for surface modeling and texture mapping. First, the intrinsic parameters of the two cameras are calibrated. Second, calculate the fundamental matrix by using the well-known Eight-Point algorithm and the essential matrix is derived to be the combination of fundamental matrix and the two camera intrinsic matrices. Third, relative pose of the two cameras can be extracted from the essential matrix and sparse 3D point reconstruction can be performed. Forth, interpolate 3D surfaces among the reconstructed sparse 3D points with the thin-plate splines. Finally, we can add textures on the reconstructed 3D surface model with some texture mapping techniques. The 3D surface model established

with the proposed reconstruction system provides useful information for robot navigation and other applications.



Acknowledgements





Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	2
1.2 Thesis Scope	5
1.3 Thesis Organization	7
2 Related Works	9
2.1 Multi-View 3D Reconstruction	11
2.2 Single-Camera 3D Reconstruction	12
2.3 Other Reconstruction Methods and Applications	13
3 Projective geometry	15
3.1 Projective Geometry	16
3.1.1 The Projective Plane	17
3.1.2 The Projective 3D Space	18
3.1.3 Projective Transformations	18
3.2 Analysis of 3D Geometry	19
3.2.1 Projective Stratum	20
3.2.2 Affine Stratum	21
3.2.3 Metric Stratum	22
3.2.4 Euclidean Stratum	23

3.2.5	Comparison of the Different Strata	23
4	Camera Model and 3D Reconstruction Fundamentals	27
4.1	The Camera Model	28
4.1.1	A Simple Camera Model	28
4.1.2	Perspective Camera Intrinsic Calibration	29
4.1.3	The Projection Matrix	31
4.2	Multi-View Geometry	32
4.2.1	Two-View Geometry	33
4.2.2	Fundamental Matrix and Essential Matrix	34
4.2.3	The Eight-Point Linear Algorithm	35
4.2.4	The Essential Matrix and Extraction of Relative Pose Between Cameras	38
4.2.5	Calculation of Depth Information for Structure Reconstruction	41
5	The Thin-Plate Splines for 3D Surface Modeling	43
5.1	The Radio Basis Function(RBFs)	44
5.2	Bounded Linear Combinations of Radio Basis Functions	46
5.3	Algebra of the Thin-Plate Splines	47
5.4	The Wendland Radial Basis Function	48
6	Our Image-Based 3D Environment Reconstruction Procedure and Results	49
6.1	Perspective Camera Calibration with a 2D Plane	50
6.2	Extraction of Relative Pose Between Cameras from Images	53
6.3	Sparse 3D Points Reconstruction	53
6.4	3D Environment Surface Modeling and Texture Mapping Using Thin-Plate Splines	55
7	Conclusion	73
	Bibliography	75

List of Figures

1.1	A 3D Laser Scanner	2
1.2	3D Laser Scanning Result - Human Face	3
1.3	3D Laser Scanning Result - Shoe	4
1.4	An image of a scene with some features specified.	5
1.5	Back-projection of a point along the line of sight.	6
1.6	An image of a scene with some features specified.	7
2.1	Camera 3D Reconstruction System - 3D Dome Developed by Narayanan, Rander and Kanade.	10
3.1	Shapes which are equivalent to a cube under different geometric transforms.	25
4.1	Perspective Camera Model.	29
4.2	From Image Coordinates to Retinal Coordinates.	30
4.3	Correspondences Between Two Views.	32
4.4	Two-View Epipolar Geometry.	33
4.5	The Pose Recovery Twisted Pair Extracted from The Essential Matrix.	39
5.1	Radio Basis Function of the Thin-Plate Splines in Two-Dimensional Space.	44
5.2	A Mathematical Model of A Thin Steel Plate.	45
6.1	Our 2D Planar Camera Calibration Planar Pattern.	57
6.2	Result Image After Binarization.	57
6.3	Result Image After Grid Alignment.	58
6.4	Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 1.	58

6.5	Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 2.	59
6.6	Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 3.	59
6.7	Scene 1 - Epipolar Lines Calculated With The Fundamental Matrix.	61
6.8	Scene 2 - Epipolar Lines Calculated With The Fundamental Matrix.	61
6.9	Scene 1 Viewpoint 1 - Individually Reconstructed Surface Model.	62
6.10	Scene 1 Viewpoint 2 - Individually Reconstructed Surface Model.	62
6.11	Scene 1 Viewpoint 3 - Individually Reconstructed Surface Model.	63
6.12	Scene 1 Viewpoint 4 - Individually Reconstructed Surface Model.	63
6.13	Scene 2 Viewpoint 1 - Individually Reconstructed Surface Model.	64
6.14	Scene 2 Viewpoint 2 - Individually Reconstructed Surface Model.	64
6.15	Scene 2 Viewpoint 3 - Individually Reconstructed Surface Model.	65
6.16	Scene 2 Viewpoint 4 - Individually Reconstructed Surface Model.	65
6.17	Scene 3 Viewpoint 1 - Individually Reconstructed Surface Model.	66
6.18	Scene 3 Viewpoint 2 - Individually Reconstructed Surface Model.	66
6.19	Scene 3 Viewpoint 3 - Individually Reconstructed Surface Model.	67
6.20	Viewpoint 1 - Reconstructed Environment Using Our Method.	68
6.21	Viewpoint 2 - Reconstructed Environment Using Our Method.	68
6.22	Viewpoint 3 - Reconstructed Environment Using Our Method.	69
6.23	Viewpoint 4 - Reconstructed Environment Using Our Method.	69
6.24	Viewpoint 5 - Reconstructed Environment Using Our Method.	70
6.25	Viewpoint 6 - Reconstructed Environment Using Our Method.	70
6.26	Wireframe - Projecting all the synthesized views onto a sphere.	71
6.27	Wireframe - Reconstructed part of all the synthesized views.	71
6.28	Textured model - Projecting all the synthesized views onto a sphere.	72
6.29	Textured model - Reconstructed part of all the synthesized views.	72

List of Tables

3.1	Comparison of Different Geometric Strata	24
6.1	The Specification of Our 3D Environment Recontruction System.	51
6.2	The Calibrated Pan-Tilt-Zoom Camera Intrinsic Parameters Using Our Calibration Method.	60





Chapter 1

Introduction





Figure 1.1: **A 3D Laser Scanner.** "A 3D laser scanner which scans in the depth data in order to rebuild virtual objects of real ones." source: (<http://www.muellerr.ch/engineering/laserscanner/default.htm>)

1.1 Background

Building 3D environment models using information from 2D images is always a main issue in computer vision. With the progress in computational speed, more and more applications were developed using these techniques. Building models suitable for use in interactive Virtual Environments (VEs) has always been a difficult problem. When the environment must be synthesized into an existing scene, this requires obtaining accurate three-dimensional environment models and poses, as well as surface materials or textures.

In addition to the appearance of the reconstructed environment, modeling the behaviour of objects is also very important if the system and the environment allow any kind of nonpassive user interaction. Generally, a scene hierarchy is constructed by specifying the relationships between objects in the scene. These relationships can then be used to assist the user in interacting with the environment.

Traditional methods of reconstructing environment models involve a skilled user and a three-dimensional CAD (Computer Aided Design) program. Accurately modeling a real environment in such a way can only be done if the user has obtained blueprints is able



Figure 1.2: **3D Laser Scanning Result of A Human Face.** "A reconstruction result of human face using 3D laser scanner." source: (<http://www.muellerr.ch/engineering/laserscanner/default.htm>)

to take precise physical measurements of the real environment. In either way mentioned above, the process is slow and exhausting even if the content of the real environment is simple. Manually obtaining surface materials and textures is also very difficult. These problems stimulates human think about how to reconstruct environment with assistance of hardware and algorithms in order to rebuild the scenes automatically.

To rebuild virtual scenes more automatically with aid of instruments, existing 3D rebuilding systems are often built with specialized hardware (e.g. laser range finders or stereo rigs) and these systems cost extremely expensive. Many new applications however demand cheaper acquisition systems. This requirement stimulates the use of consumer photo- or video cameras. Moores law also tells us that more and more can be done in software because of the recent progress in digital imaging instruments.

Due to the factors mentioned above, many techniques using informations captured from cameras have been developed over the last few years. Many of these techniques do not require more than some cameras and a computer to rebuild three-dimensional models of real objects.

An image like in Figure 1.4 tells us a lot about the observed scene. There is how-



Figure 1.3: **3D Laser Scanning Result of A Shoe.** "A 3D rebuilding result of a shoe using 3D laser scanning data." source: (<http://www.muellerr.ch/engineering/laserscanner/default.htm>)

ever not enough information to reconstruct the 3D scene without doing an sufficient number of assumptions on the structure of the scene. This is due to the nature of the image formation process which consists of a projection from a three- dimensional scene onto a two-dimensional image. During this process the depth information of the 3D point is lost. Figure 1.5 illustrates this projection problem. The three-dimensional point corresponding to a specific image point is constraint to be on the associated line of sight. From a single image it is not possible to determine which point on this line corresponds to the image point. If two or more images are available, then Figure 1.6 shows that the coordinate of the three-dimensional point can be obtained as the intersection of the two back-projected rays. This process is called *triangulation*. Notice that, however, some prior knowledge must be required for triangulation:

- Corresponding image points
- Relative pose of the camera for the different views
- Relation between the image points and the corresponding line of sight



Figure 1.4: **An image of a scene with some features specified.**

The relation between an image point and its back-projected ray is given by the camera model (e.g. perspective camera) and its calibration parameters. These parameters are often called the *intrinsic* camera parameters while the position and orientation of the camera are usually called *extrinsic* parameters. In the following of this thesis we will learn how all these elements that can be retrieved from the images. The key for this are the relations between multiple views which tell us that corresponding sets of points must contain some particular structure and that this structure is related to the poses and the calibration of the camera.

1.2 Thesis Scope

In this thesis, we proposed a 3D reconstruction procedure using images captured by cameras at different poses. The relation between an image point and its corresponding ray of sight is given by the camera model (e.g. perspective camera) and the camera calibration parameters. These parameters are often called the intrinsic camera parameters while the position and orientation of the camera are in general called camera extrinsic parameters. In the following chapters we will learn how all these elements can be acquired from the

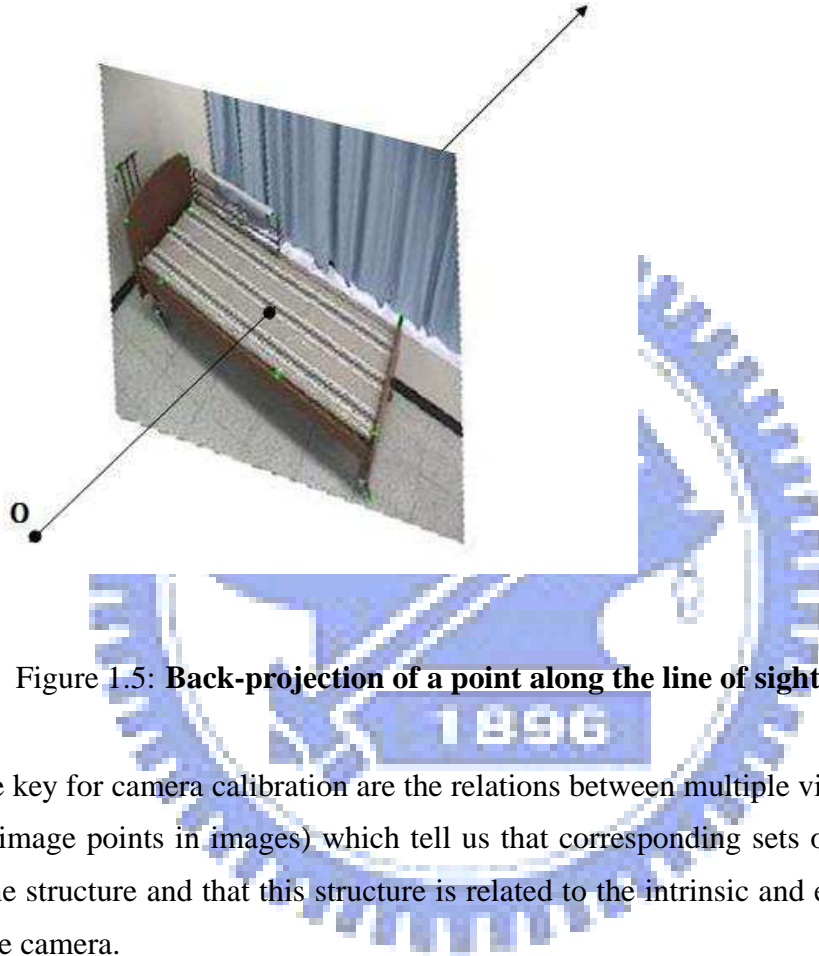


Figure 1.5: **Back-projection of a point along the line of sight.**

images. The key for camera calibration are the relations between multiple views (e.g. corresponding image points in images) which tell us that corresponding sets of points must contain some structure and that this structure is related to the intrinsic and extrinsic parameters of the camera.

In our 3D reconstruction procedure, first, the intrinsic parameters of the two cameras are calibrated. Second, we calculate the fundamental matrix by using the well-known Eight-Point algorithm and the essential matrix is derived to be the combination of fundamental matrix and the two camera intrinsic matrices. Third, relative pose of the two cameras can be extracted from the essential matrix and sparse 3D point reconstruction can be performed. Fourth, interpolate 3D surfaces among the reconstructed sparse 3D points with the thin-plate splines. Finally, we can add textures on the reconstructed 3D surface model with some texture mapping techniques. The 3D surface model established with the proposed

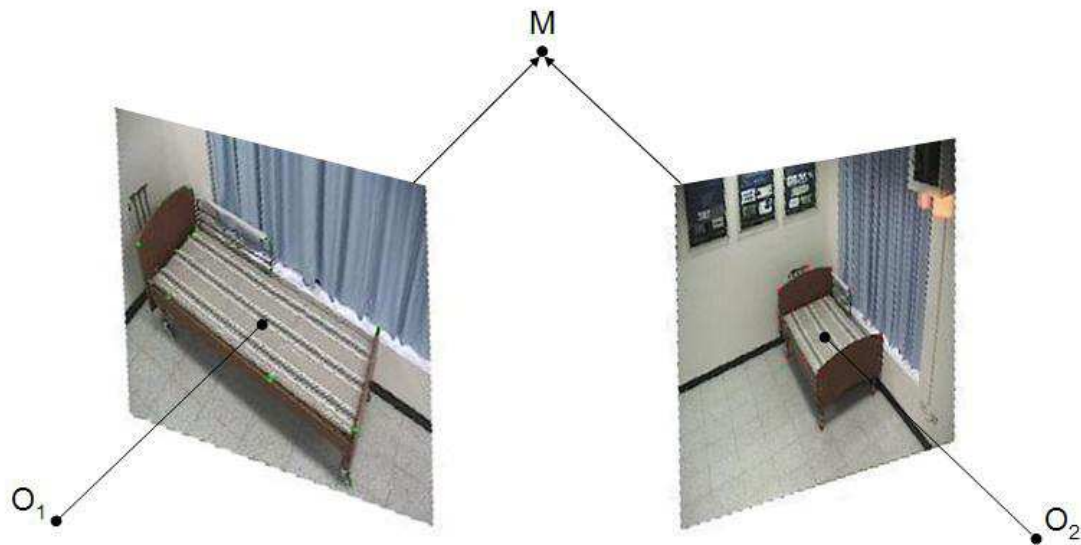


Figure 1.6: An image of a scene with some features specified.

reconstruction system provides useful information for robot navigation and other applications.

1.3 Thesis Organization

After this chapter, we will introduce some relative works during the pass few years. Chapter 3 describes projective geometry and the stratification of geometric structure. After some geometric fundamentals are introduced, we turn into the perspective camera model and some geometric calculation of the relation between multiple view cameras in chapter 4. Chapter 5 tells the main 3D surface modeling method we use to reconstruct photo texture

mapped 3D models in this thesis and then run the way through to perform our 3D model reconstruction procedure in chapter 6. Some reconstruction and experiment results are shown in chapter 7 and finally, we have some conclusion and future works discussed in chapter 8.



Chapter 2

Related Works



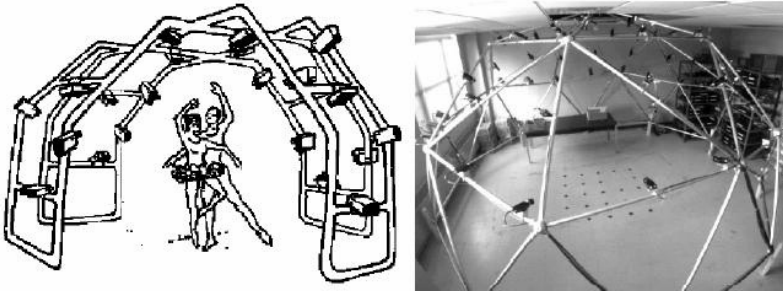


Figure 2.1: **Camera 3D Reconstruction System - 3D Dome Developed by Narayanan, Rander and Kanade.**

The technique of 3D reconstruction from stereo images of real scenes has been studied for many years. The focus points of 3D reconstruction studies vary due to different requirements of various applications such as robot navigation, 3D model reconstruction of architectures, computer graphics, virtual reality, etc.

Take robot navigation for example, robot vision systems demand no sophisticated or realistic reconstruction results but only the accuracy of depth information and some principle parts of the environment, therefore, the researchers of robot vision systems focus on how to calculate depth information from images precisely and efficiently.

Another example is the 3D virtual model reconstruction of a specific real object, the most common way nowadays is to put the object on a rotating plate and keep capturing images with a stationary camera while the plate rotates. The camera can be calibrated first in order to acquire the relationship between image points and its reprojection rays. Camera motion can be formulated since the rotation speed and the radius of the rotation are prior knowledge under the model reconstruction system. The main purpose of 3D virtual model reconstruction systems is to build photo-realistic models from a sequence of images.

In the following sections of this chapter, we will introduce several methods about how to rebuild 3D virtual models from images captured by various poses of cameras.

2.1 Multi-View 3D Reconstruction

Multi-view 3D reconstruction systems rebuild the model from photos captured by several cameras of different poses. The corresponding features are found among cameras in order to calculate 3D coordinates of the real object. Cameras in multi-view rebuilding systems are often fully calibrated so that their relative poses are known. Acquire enough 3D information by tracking the motion of an moving object with multiple calibrated cameras is the main advantage of these systems.

Narayanan, Rander and Kanade proposed a multi-view photographic reconstruction system called *3D Dome* [12]. As illustrated in Figure 2.1, the system 3D Dome is a semi-sphere multi-capturing system formed by fifty-one synchronous and fully calibrated cameras. Since all the cameras are all fully calibrated, which means in the Equation 4.6 the camera intrinsic matrix \mathbf{K} and relative poses among cameras are all known. Therefore, when a person is taking some actions in the 3D dome, every camera around the 3D dome will capture images from different point of views and then obtain a dense depth graph for each camera by running through a multiple-baseline stereo reconstruction procedure. Mapping the texture onto the dense depth graph forms a simple reconstructed 3D human model. The author called this a *visible surface model (VSM)*. But VSM is a surface model reconstructed from each camera, there is some part of the human have inevitable reconstruction difficulties due to occlusion. The author solved this problem by synthesizing all the VSMs together with a optimized integration procedure in order to reconstruct the *complete surface model (CSM)* of the scene.

In the 3D dome system, cameras are all fully calibrated with relative poses known. Therefore, in the 3D dome system, calculation of 3D coordinates from the photos needs no complicated computation. Since the system is equipped with 51 cameras, the main problem of the system is to have cameras capture images synchronously.

Fua and Leclerc have proposed a similar system which goal is to rebuild the real scenes in virtual reality [6]. Differ from Narayanan, Rander and Kanade, they use only two calibrated cameras to capture images of static scenes. Fua and Leclerc turn the calculated 3D points into meshes in order to reconstruct the 3D surface model of the scenes.

2.2 Single-Camera 3D Reconstruction

Single-camera 3D reconstruction systems often obtain images with a single camera but from different point of view or simply record videos while the camera is moving. The most often used method is the so-called *structure from motion*. Image processing methods, such as *multi-image intensity correlation*, can be used in single camera video systems in order to find out image correspondences since there should be small differences between frames in short-term intervals.

Pollefeys and Van Gool [13] have implemented a single camera reconstruction system similar to described systems above. The input of their system is a sequence of images captured from the same scene by single camera. After specifying some distinct features in each image, similarity comparison methods are used to find out correspondences among images. Since there are some errors in images due to camera projection hardware structure and some noises caused if the feature points were specified by human, Pollefeys and Van used a method called *random sampling consensus*(RANSAC) to calculate several choices of the fundamental matrices from the image correspondences and picked the most stable fundamental matrix out from the computed matrices. The fundamental matrix encodes the transformation of every image points in two corresponding images, the definition of the fundamental matrix and epipolar geometry will be introduced in chapter 3 and 4. After finding out the fundamental matrix, a projective reconstruction can be computed. If the cameras are calibrated, the intrinsic parameters of the camera matrix are all known and thus a metric reconstruction can be performed, which differs from the real world by only a scalar factor. After the metric reconstruction is done, the author computed the dense depth graph in order to calculate depth for every pixel in the image and then performed texture mapping to reconstruct the whole 3D model.

Fitzgibbon and Zisserman have proposed a similar system, but besides finding feature points, they used the informations of line segments in the images by using image processing techniques such as edge detection for 3D reconstruction. Therefore, their reconstruction results are not only sparse 3D points but with the information support of line segments, which can assist in reducing the depth error of the 3D reconstruction. In addition, Zisserman used

both the two-view reconstruction method and the trifocal tensor method, which is to cut down depth ambiguity by using three-view image correspondences.

2.3 Other Reconstruction Methods and Applications

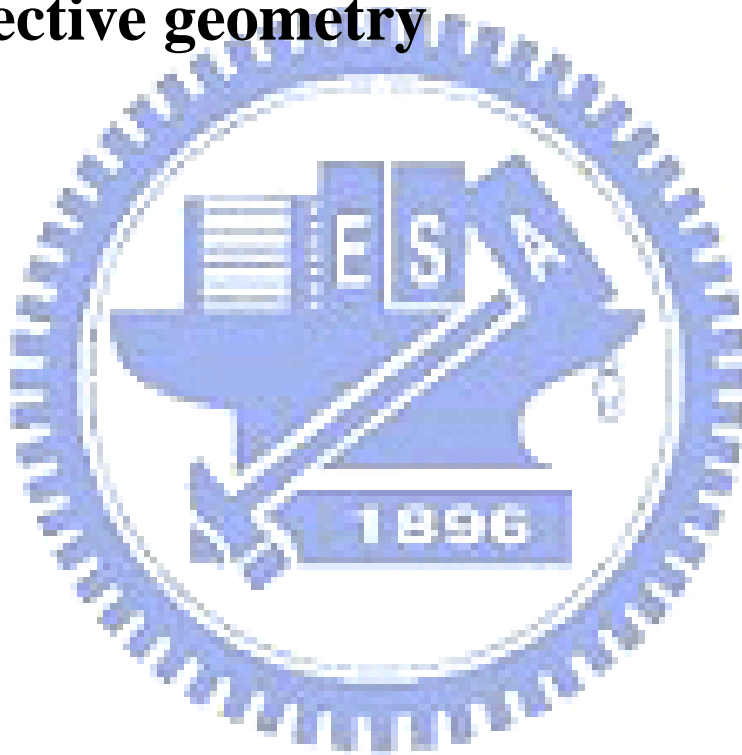
There are lots of applications require image-based 3D reconstruction systems for assistance. Schreer [15] has developed a robot navigation system with photographic 3D reconstruction algorithms built in. The two cameras used for robot vision are both fully calibrated in order to calculate 3D coordinates in real-time while the robot is moving around in the environment. But this system uses only the distribution condition of the reconstructed 3D points with some prior knowledge and experiences in order to avoid obstacles. But lack of considering the structure of indoor environments may cause the robot vision system inflexible.

Some reconstruction systems use some characteristics of the scenes to refine the reconstructed model. Cipolla and Robertson [3] used the prior knowledge such as the perpendicular relations among walls and floors of the buildings to find the vanishing point in the image. The vanishing point is then transferred into a 3D vector form in order to reduce computational error of the vanishing point. After the vanishing point is found, the camera intrinsic parameters can be calculated with the vanishing point in order to simplify the calculation process of camera intrinsic parameters and 3D model reconstruction.



Chapter 3

Projective geometry



The concepts presented in the following two chapters concentrates on concepts of projective geometry. This chapter and the next one introduce most of the geometric concepts used in the rest of the thesis. This chapter focuses on projective geometry and introduces concepts as points, lines and planes in two or three dimensions. A lot of attention goes to the analysis of geometry in projective, affine, metric and Euclidean layers. Projective geometry is used for its simplicity in formalism, additional structure and properties that can then be introduced were needed through this hierarchy of geometric strata. This section was inspired by the introductions on projective geometry found in Faugeras' book [5]. A detailed description on the subject can be found in the recent book by Hartley and Zisserman [8].

3.1 Projective Geometry

A point in projective n -space \mathcal{P}^n is given by a $(n + 1)$ -vector of coordinates $x = [x_1 \dots x_{n+1}]^T$. At least one of these entries of the vector should differ from zero. These coordinates are called *homogeneous* coordinates. In this thesis the coordinate vector and the point itself will be denoted with the same symbol. Two points denoted by $(n + 1)$ -vectors x and y are equal if and only if there exists a nonzero scalar λ such that $x = \lambda y$. This will be indicated by $x \sim y$.

A *collineation* is a mapping between projective spaces. A collineation from \mathcal{P}^m to \mathcal{P}^n can be mathematically denoted by a $(m + 1) \times (n + 1)$ matrix H , where points are transformed linearly: $x' \sim Hx$. Matrices H and λH with a nonzero scalar λ represent the same collineation.

A *projective basis* is the extension of a coordinate system to projective geometry. A projective basis is a set of $n + 2$ points such that no $n + 1$ of them are linearly dependent. The set $e_l = [0, \dots, 1, \dots, 0]^T$, $\forall l, 1 \leq l \leq n + 1$, where 1 is in the l th position and $e_{n+2} = [1, 1, \dots, 1]^T$ is the standard projective basis. A projective point of \mathcal{P}^n can be described as a linear combination of any $n + 1$ points of the standard basis. For example:

$$m = \sum_{l=1}^{n+1} \lambda_l e_l$$

It can be shown [4] that any projective basis can be transformed into a unique collineation of the standard projective basis. Similarly, if two sets of points m_1, \dots, m_{n+2} and m'_1, \dots, m'_{n+2} both form a projective basis, then there exists a uniquely resolved collineation T such that $m'_l \sim Tm_l, \forall l, 1 \leq l \leq n+2$. This collineation T describes the different combination of projective basis. Notice that T is invertible.

3.1.1 The Projective Plane

The projective plane is the projective space \mathcal{P}^2 . A point in \mathcal{P}^2 is represented by a 3-vector $m = [x, y, z]^T$. A line l is also represented by a 3-vector. A point m is located on a line l if and only if

$$l^T m = 0 \quad (3.1)$$

This equation, however, can also be described as the expression that "the line l passes through the point m " or "the point m is on the line l ". This symmetry in the equation shows that there is no formal difference between points and lines in the projective plane. This is known as the principle of *duality*. A line l passing through two points m_1 and m_2 is given by their vector product $m_1 \times m_2$. This can also be written as

$$l \sim [m_1]_{\times} m_2, \text{ with } [m_1]_{\times} = \begin{bmatrix} 0 & z_1 & -y_1 \\ -z_1 & 0 & x_1 \\ y_1 & -x_1 & 0 \end{bmatrix} \quad (3.2)$$

The dual formulation gives the intersection of two lines. All the lines passing through a specific point form a *pencil of lines*. If two lines l_1 and l_2 are distinct elements of the pencil, all the other lines can be obtained through the following equation:

$$l \sim \lambda_1 l_1 + \lambda_2 l_2 \quad (3.3)$$

for some scalars λ_1 and λ_2 . Note that the ratio $\frac{\lambda_1}{\lambda_2}$ is important.

3.1.2 The Projective 3D Space

A projective 3D space typically means the dimension of the projective space is 3, where is the projective space \mathcal{P}^3 . An element in \mathcal{P}^3 is represented by a 4-entry vector $M = [X, Y, Z, W]^T$. In \mathcal{P}^3 the duality of an element is a plane, which is also denoted as a 4-entry vector. A point M lies on a plane Π can be denoted mathematically as:

$$\Pi^T M = 0 \quad (3.4)$$

A line can be written into a linear combination of two points as:

$$\lambda_1 M_1 + \lambda_2 M_2$$

or can be produced by the intersection of two planes $\Pi_1 \cap \Pi_2$.

3.1.3 Projective Transformations

We can denote a transformation between the images as a *homography* of $\mathcal{P}^2 \rightarrow \mathcal{P}^2$, which can be represented by a 3×3 -matrix H . With the same properties of matrices, H and λH represent the same homography for all nonzero scalars λ . A point is transformed as follows:

$$m \mapsto m' \sim Hm \quad (3.5)$$

The corresponding transformation of a line can be obtained by transforming the points which are on the line and then finding the line defined by these points:

$$l'^T m' = l'^T H^{-1} Hm = l^T m = 0 \quad (3.6)$$

From the previous equation it is easy to derive a transformation equation for a line ($H^{-T} = (H^{-1})^T = (H^T)^{-1}$):

$$l \mapsto l' \sim H^{-T} l \quad (3.7)$$

Similar reasons can be considered in \mathcal{P}^3 gives the following equations for transformations of points and planes in 3D space:

$$M \mapsto M' \sim TM, \quad (3.8)$$

$$\Pi \mapsto \Pi' \sim T^{-T}\Pi \quad (3.9)$$

where T is a 4×4 -matrix.

3.2 Analysis of 3D Geometry

Usually we define the real world as a Euclidean 3D space. But in some particular cases it is not sufficient to use the full Euclidean structure of 3D space. Euclidean 3D space is only suitable for less structured and thus simpler projective geometry. Intermediate layers are formed by the affine and metric geometry. These structures can be thought of as different geometric layers which can be overlaid on the world for different transformations. The most complicated is Euclidean, then metric, next affine and finally projective structure.

The concept of stratification is closely related to the groups of transformations acting on geometric entities and leaving some properties of configurations of these elements invariant. Attached to the projective stratum is the set of projective transformations, attached to the affine stratum is the set of affine transformations, attached to the metric stratum is the set of similarities and attached to the Euclidean stratum is the set of Euclidean transformations. It is important to notice that these groups are subgroups of each other, e.g. the metric group is a subgroup of the affine group and both are subgroups of the projective group.

An important aspect related to these groups are their invariants. An *invariant* is a property of a derivation of geometric entities that is not altered by any transformation belonging to a specific group. Invariants therefore can guild us what measurements we can do considering a specific stratum of geometry. These invariants are often related to geometric entities which stay unchanged after applying the transformations to a specific group. These geomet-

ric entities with invariants related play an important role in part of this thesis. Recovering them allows us to upgrade the structure of the geometry to a higher level of the geometric stratification.

In the following sections, different strata of geometry are discussed. The associated groups of transformations, their invariants and the corresponding invariant structures are presented.

3.2.1 Projective Stratum

The simplest stratum is the projective stratum. It is the less structured one and has the least number of invariants and the largest group of transformations related to it. The group of projective transformations or collineations is composed with the most general group of linear transformations.

A projective transformation of 3D space can be denoted by a 4×4 -matrix, where the matrix is invertible:

$$T_P \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \quad (3.10)$$

This transformation matrix is only defined up to a nonzero scale factor and has therefore 15 degrees of freedom.

Relations of collinearity, incidence and tangency are projectively invariant. The cross-ratio is an invariant property under projective transformations as well. It is defined as follows: Assume that the four points M_1, M_2, M_3 and M_4 are collinear. Then they can be expressed as $M_i = M + \lambda_i M'$ (assume none is coincident with M'). The cross-ratio is defined as

$$\{M_1, M_2; M_3, M_4\} = \frac{\lambda_1 - \lambda_3}{\lambda_1 - \lambda_4} : \frac{\lambda_2 - \lambda_3}{\lambda_2 - \lambda_4} \quad (3.11)$$

The cross-ratio does not depend on the choice of the reference points M and M' and is invariant under the group of projective transformations of \mathcal{P}^3 . It can be derived that a similar cross-ratio invariant for four line intersecting in a point or four planes intersecting in a line.

We can regard cross-ratio as the coordinate of the fourth point is the linear combination of the first three points, since three points form a basis for a projective line in \mathcal{P}^1 . Similarly, two invariants can be found for five coplanar points, three invariants for six coplanar points, all in general position.

3.2.2 Affine Stratum

The affine stratum has more structure than the projective one, but less structure than the metric or the Euclidean strata. Differs from projective stratum, the affine stratum identifies a special plane, which called the *plane at infinity*.

To define this plane at infinity, we have $W = 0$ and thus $\Pi_\infty = [0, 0, 0, 1]^T$. We can consider that the projective space contains the affine space under the one-to-one mapping: $\mathcal{A}^3 \rightarrow \mathcal{P}^3: [X, Y, Z]^T \mapsto [X, Y, Z, 1]^T$. The plane $W = 0$ in \mathcal{P}^3 can be thought as containing the limit points for $\|M\| = \infty$. The Affine transformation is usually denoted as the following:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}, \text{ with } \det(a_{ij}) \neq 0 \quad (3.12)$$

The affine transformation can be rewritten in the matrix form: $M' \sim T_A M$ with:

$$T_A \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

Therefore, the affine transformation has 12 independent degrees of freedom. All invariants under the projective stratum also exist under the affine stratum. For the more restrictive affine group, parallelism is added as a new invariant property. Lines or planes having their intersection at infinity are called *parallel*. Another new invariant property for affine group is the *ratio of lengths along some direction*.

3.2.3 Metric Stratum

The metric stratum resembles in the group of similarities. This stratum differs from the Euclidean stratum only up to a scale factor. The metric transformations correspond to Euclidean transformations complemented with a scaling. When no absolute measurement is available, reconstruction in the metric coordinate is the highest level of geometric structure that 3D reconstruction from images can achieve.

A metric transformation can be represented as the following:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \sigma \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_{14} \\ t_{24} \\ t_{34} \end{bmatrix} \quad (3.14)$$

with r_{ij} the coefficients of an orthonormal matrix, which is usually denoted by R such that $R^T R = R R^T = I$ and thus $R^{-1} = R^T$. Recall that R is a rotation matrix if and only if $R R^T = I$ and $\det(R) = 1$. In homogeneous coordinates, Equation 3.14 can be rewritten as $M' = T_M M$, with

$$T_M \sim \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_X \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_Y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & \sigma^{-1} t_X \\ r_{21} & r_{22} & r_{23} & \sigma^{-1} t_Y \\ r_{31} & r_{32} & r_{33} & \sigma^{-1} t_Z \\ 0 & 0 & 0 & \sigma^{-1} \end{bmatrix} \quad (3.15)$$

A metric transformation therefore has 7 independent degrees of freedom, 3 for translation, 3 for orientation and 1 for scale. In metric stratum there are two important new invariants properties: *relative lengths* and *angles*.

3.2.4 Euclidean Stratum

The only difference between Euclidean stratum and metric stratum is *absolute length*. Therefore, the Euclidean transformation has 6 independent degrees of freedom, 3 for translation and 3 for rotation. A Euclidean transformation has the following matrix form:

$$T_E \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

with r_{ij} the coefficients of an orthonormal matrix, as described previously. If $\det(R) = 1$ then, this transformation is simply the same as a rigid-body transformation in space.

3.2.5 Comparison of the Different Strata

In this chapter some concepts of projective geometry were introduced. Based on these concepts, some methods can be invented by doing the inverse of the projection process and obtain 3D reconstructions of the observed scenes, where is the main objective of this thesis. We can list a table in order to compare different strata described previously:

ambiguity	DOF	transformation in matrix form	invariants
projective	15	$T_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$	cross-ratio
affine	12	$T_A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances along direction parallism <i>plane at infinity</i>
metric	7	$T_M = \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_x \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances angles <i>absolute conic</i>
Euclidean	6	$T_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	<i>absolute distances</i>

Table 3.1: **Comparison of Different Geometric Strata.** "Number of degrees of freedom, transformation in matrix form and invariants corresponding to different geometric strata."

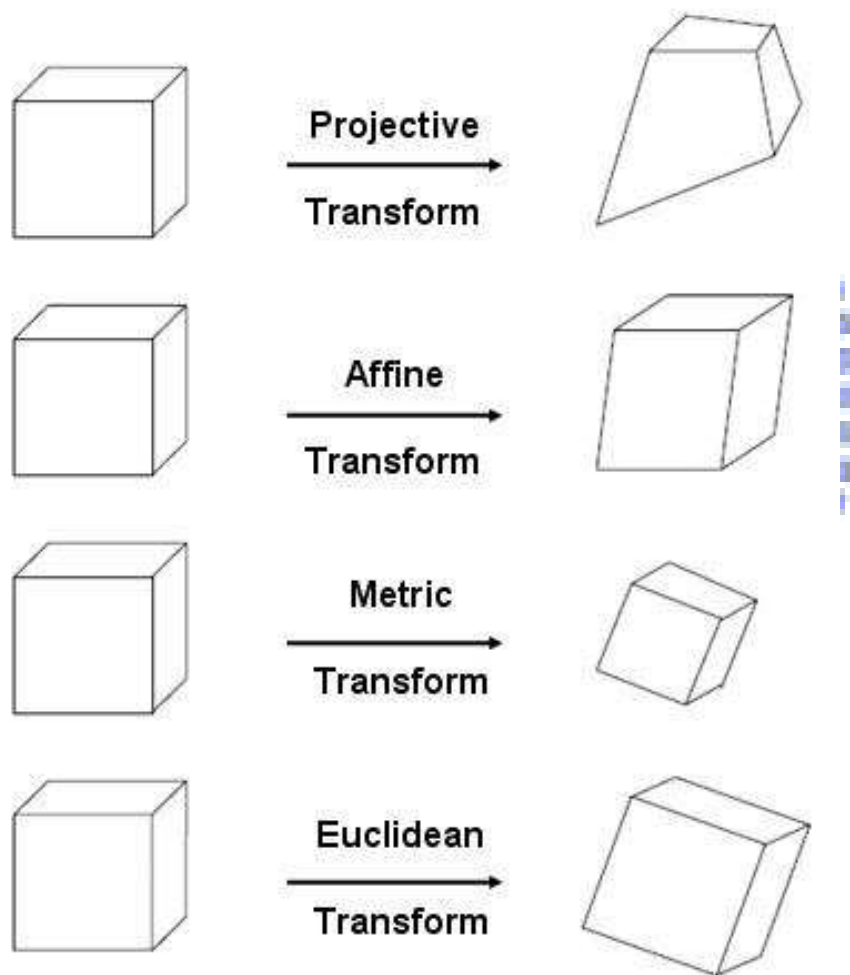


Figure 3.1: Shapes which are equivalent to a cube under different geometric transforms.



Chapter 4

Camera Model and 3D Reconstruction Fundamentals



Before discussing how to reconstruct 3D objects from relations of images captured from different poses of cameras, it is important to know how images are formed via the camera model. In the following sections, first, the perspective camera model is introduced. Second, some important relationships between multiple views are presented with some mathematics.

4.1 The Camera Model

In this thesis the model of perspective camera is used. The image-forming process is completely determined by having a perspective projection center point and a retinal plane. The projection of a real 3D point is then obtained as the intersection of a line passing through this real 3D point and the projection center C with the image plane \mathcal{R} .

4.1.1 A Simple Camera Model

In the simplest case, where the center of projection C is placed at the origin of the world frame and the image plane is at $Z = 1$, the projection process can be formulated as follows:

$$x = \frac{X}{Z}, y = \frac{Y}{Z} \quad (4.1)$$

For a world point (X, Y, Z) and its corresponding projected image point (x, y) . Using the homogeneous representation of the points, a linear equation is then obtained as the following:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.2)$$

This projection is illustrated in Figure 4.1, where the optical axis passes through the projection center C and is orthogonal to the retinal plane \mathcal{R} . The intersection of the optical

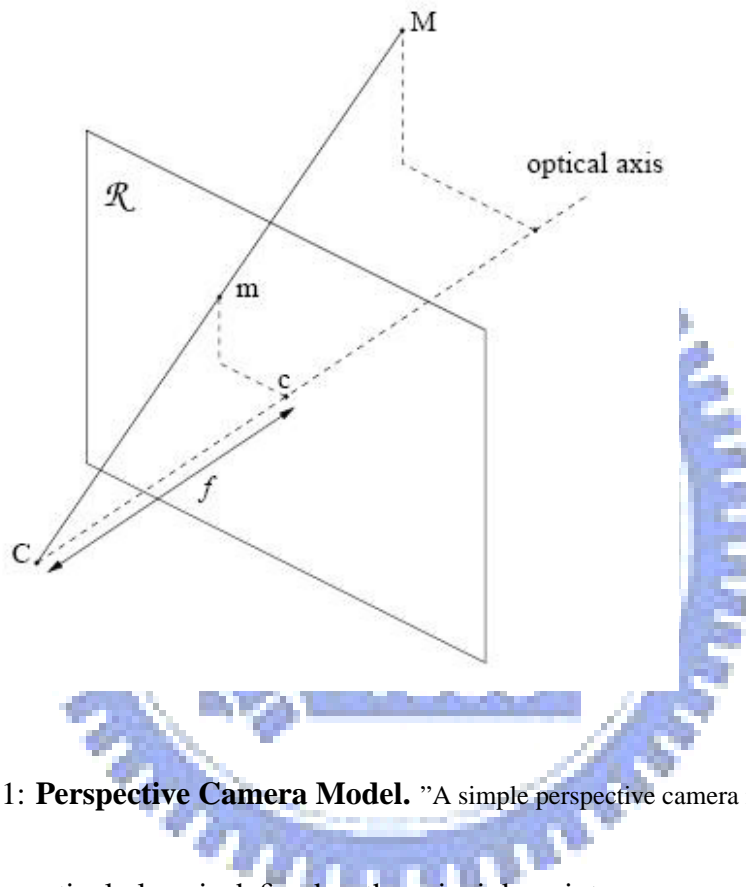


Figure 4.1: **Perspective Camera Model.** "A simple perspective camera model, cited from: [14]."

axis and the retinal plane is defined as the principle point c .

4.1.2 Perspective Camera Intrinsic Calibration

Now consider the case when actual camera is used, where the focal length f will be different from 1, the coordinates of Equation 4.2 should be scaled with f to take account.

In addition the coordinates in the image output on the screen do not match the physical coordinates in the retinal plane. Using a CCD camera the relation between the image coordinate and the retinal coordinate depends on the size and shape of the pixels and of the

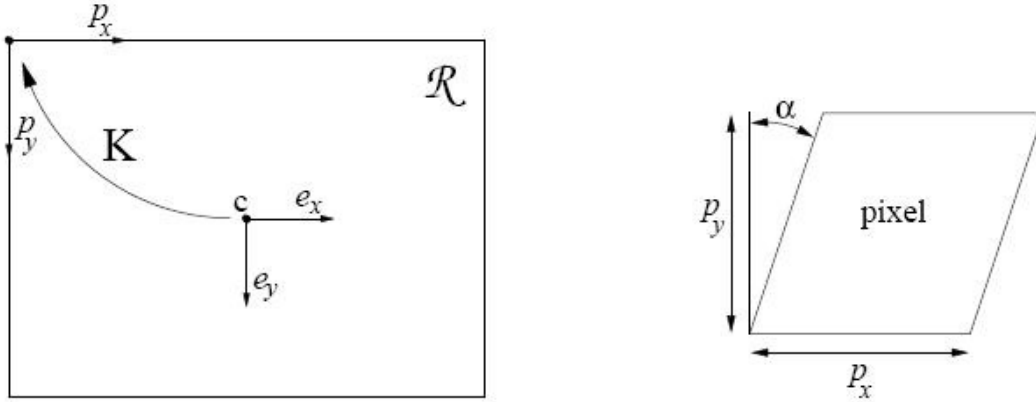


Figure 4.2: **From image coordinates to retinal coordinates.** "This figure illustrates how image coordinates transform to retinal coordinates, cited from: [14]."

position of the CCD chip placed in the camera. The projection process of actual perspective camera can be formulated in matrix form as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_x} & (\tan \alpha) \frac{f}{p_y} & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} \quad (4.3)$$

where p_x and p_y are the width and height of the pixels, the principle point $\mathbf{c} = [c_x, c_y, 1]^T$ and α the skew angle as shown in Figure 4.2. Since only the ratios $\frac{f}{p_x}$ and $\frac{f}{p_y}$ are important, we can write a simplified notation of Equation 4.3 as the following:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} \quad (4.4)$$

with f_x and f_y the focal length measured in width and height of pixels, and s a factor being the skew factor due to non-rectangular pixels. The above upper triangular matrix is

called the *intrinsic camera calibration matrix*, and the notation \mathbf{K} is usually used for the matrix. For a camera with fixed optics these parameters are identical for all the images taken with the camera. For cameras which have zooming and focusing capabilities the focal length can obviously change, but also the principal point can vary. In order to find out the camera intrinsic parameters, we use the calibration method proposed by Z.Zhang [23], which calibrates perspective cameras with a 2D plane with some features easily extracted by image processing techniques.

4.1.3 The Projection Matrix

Combining Equations(4.2), (4.4) and rigid-body transformation of the camera, the following expression can be written with camera intrinsic parameters defined previously and with a specific camera position and orientation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^T t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.5)$$

which can be simplified to:

$$m \sim K \begin{bmatrix} R^T & -R^T t \\ 0_3^T & 1 \end{bmatrix} M \quad (4.6)$$

or even

$$m \sim PM \quad (4.7)$$

The 3×4 matrix P is called the *camera projection matrix*, which determines how real world 3D points turn into image 2D points we saw on the monitor screen. With the Equation 4.7 the plane Π corresponding to a back-projected line l can also be derived: Since $l^T m \sim l^T P M \sim \Pi^T M$,

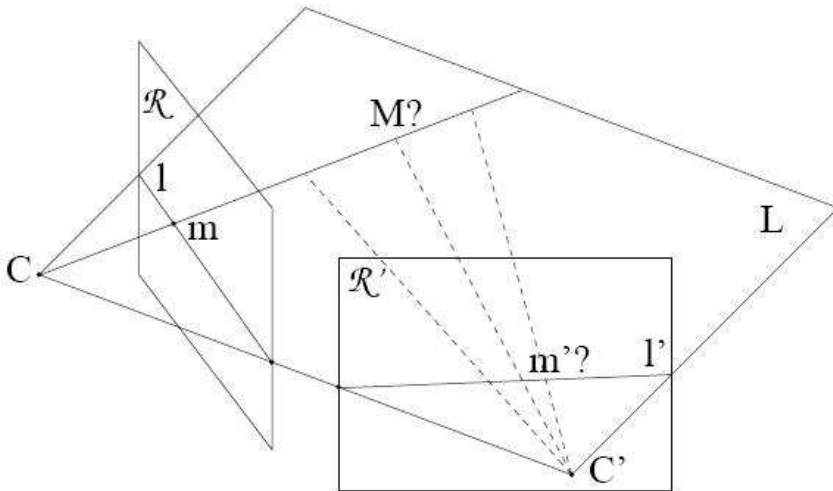


Figure 4.3: **Correspondences Between Two Views.** "Even the exact position of M is not known, it is bounded on the line of sight of the corresponding image point m . This line of sight can be projected on the other camera image plane as l' , cited from: [14]."

$$\Pi \sim P^T l$$

(4.8)

4.2 Multi-View Geometry

In the previous sections multi view relations were not discovered. Since several geometric relationships can be build between two, three or more images, these relationships are the essential parts for camera calibration and 3D reconstruction from images. Many insights of multiple view geometry are discovered over the last few decades.

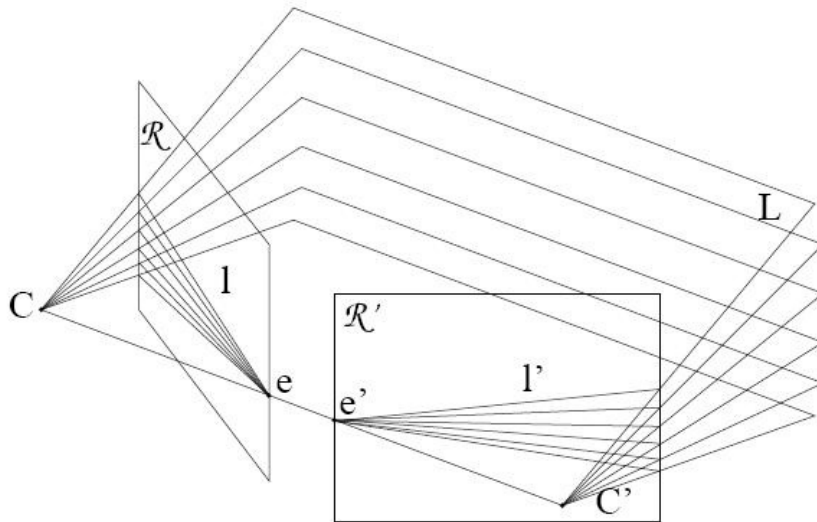


Figure 4.4: **Two-View Epipolar Geometry.** "This figure illustrates that different epipolar planes formed by 3D points and the two projection centers C and C' always include the baseline and the two epipoles e and e' . Each epipolar plane satisfies epipolar geometry and can be formulated in mathematical way (cited from: [14])."

4.2.1 Two-View Geometry

After the intrinsic parameters of the perspective camera are known, we can calculate the corresponding ray of an specific image point passing itself and the projection center. Consider that there are two cameras with different positions and orientation capturing images from the same scene, is there any relations between the images formed by these cameras? A more specific question: *Given one image point in an image, can this point restrict the position of an image point in the other image?* It turns out that this relationship can be obtained from the process of camera calibration or even from a set of prior image point correspondences.

To answer this question, consider the projection relationships of a real world 3D point

M among two cameras, although the exact position of M is not known, it is bounded on the line of sight of the corresponding image point m . This line of sight can be projected on the other camera image plane as shown in Figure 4.3. In fact all the points on the plane Π defined by the two projection centers and M have their image on l' . The same reason that line l is formed by the projecting all the points on the plane Π onto the left image. l and l' are said to be in *epipolar correspondence*, the plane Π is usually named with *epipolar plane*.

All these epipolar planes pass through both projection centers C and C' , results in a set of corresponding epipolar lines as shown in Figure 4.4. All these epipolar lines pass through two specific points e and e' , which are commonly called *epipoles*.

This epipolar geometry can be represented mathematically. A point m on a line l can be expressed in the formula as $l^T m = 0$. The line passing through point m and the epipole e is:

$$l \sim [e]_{\times} m \quad (4.9)$$

with $[e]_{\times}$ the antisymmetric 3×3 matrix describing the cross-product of the epipole e .

4.2.2 Fundamental Matrix and Essential Matrix

After describing the basic two-view epipolar geometry, we can now go further into some derivations of the fundamental matrix and the essential matrix. From Equation 4.8 and Equation 4.9 the plane Π can be easily obtained as $\Pi \sim P^T l$ and similarly $\Pi \sim P'^T l'$. Combining these equations gives the following formula:

$$l' \sim (P'^T)^{\dagger} P^T l \equiv H^{-1} l \quad (4.10)$$

with \dagger denoting the pseudo-inverse. Substituting (4.9) in (4.10) we have the following equation:

$$l' \sim H^{-T}[e]_x m \quad (4.11)$$

defining $F = H^{-T}[e]_x$ and substitute in Equation 4.11, we have:

$$l' \sim Fm \quad (4.12)$$

and thus,

$$m'^T Fm = 0 \quad (4.13)$$

The matrix F is called the *fundamental matrix*. These definitions and concepts were introduced by Faugeras [4] and Hartley [7]. Many people have studied the properties of the fundamental matrix (e.g. Q.T. Luong [9] and [10]) and lots of efforts have been put in obtaining the fundamental matrix from two-view image pairs robustly [16–18].

When the calibration is not known, the fundamental matrix F can be calculated by Equation (4.13). Every pair of image correspondences gives one constraint on the fundamental matrix F . Since F is a 3×3 matrix which is determined only up to a scalar factor, it has $3 \times 3 - 1$ unknowns, which means eight pairs of image correspondences are sufficient to compute F with a linear algorithm. The linear algorithm is then introduced in the following section.

4.2.3 The Eight-Point Linear Algorithm

Linear Solution for the Fundamental Matrix

As described in the previous section, the fundamental matrix is defined by Equation 4.13, for any matching image pairs $m \leftrightarrow m'$. Given a sufficient number of image point matches (at least eight) $m_i \leftrightarrow m'_i$, Equation 4.13 can be used to compute the unknown fundamental matrix F . Let $m = [u, v, 1]^T$ and $m' = [u', v', 1]^T$, every point correspondence gives one constraint linear equation to an unknown entry of F . The coefficients of the equation can be easily derived in coordinates of m and m' as the following:

$$uu'F_{11} + uv'F_{21} + uF_{31} + vu'F_{12} + vv'F_{22} + vF_{32} + u'F_{13} + v'F_{23} + F_{33} = 0 \quad (4.14)$$

The coefficients of the equation can be written into a row vector as follows:

$$(uu', uv', u, vu', vv', v, u', v', 1) \quad (4.15)$$

Let the row vector in the Equation 4.15 be matrix A , and the nine-vector column vector f be the stacked-version matrix containing the entries of the fundamental matrix F . Then we obtain a set of linear equations of the form:

$$Af = 0 \quad (4.16)$$

Because the fundamental matrix F is defined up to an unknown scalar factor, to avoid the trivial solution f , an additional constraint can be used as follows:

$$\|f\| = 1 \quad (4.17)$$

where $\|f\|$ is the norm of f .

With the constraints described above, it is possible to find a solution to the linear system with as few as eight image pairs. If more than eight point correspondences are specified, we have an overspecified system of equations. Assuming that there exists a non-zero solution to this system of equations, A is derived to be rank-deficient. In other words, although A has nine columns, the rank of A must be at most eight. In fact, the rank of A is exactly eight, and there is a unique solution f .

The above discussion assumes that the given point correspondences are all perfect data and without the disturbance of noise. Actually, because of inaccuracies in the measurement or specification of the matched points, the matrix A will not be rank-deficient, which means it will have rank nine. In this case, there will not be any nontrivial solutions to the system of equations $Af = 0$. Instead of finding a non-zero solution, we seek a least-squares solution

to this equation set, where is well known to be the unit eigenvector corresponding to the smallest eigenvalue of $A^T A$. An appropriate algorithm for finding this eigenvector can refer to the algorithm of Jacobi [21] or the *singular value decomposition*(SVD) [1, 21].

The properties of the fundamental matrix will be introduced in the next paragraph.

The Singularity Constraint and The Eight-Point Algorithm

A important property of the fundamental matrix is that it is singular, which is in fact has rank of two. Furthermore, the left and right null-spaces of the fundamental matrix F can be generated by the vectors in homogeneous coordinate denoting the two epipoles in the two relative images. Most applications depends on the rank two constraint of the matrix F . But the matrix F found by solving the system of equations (4.16) will not in general have rank two due to the noise and the error of measurement. Therefore, a convenient method to enforce the singularity constraint and compute the fundamental matrix is to use the singular value decomposition. In particular, let $F = UDV^T$ be the SVD of F , where D is a diagonal matrix $D = \text{diag}(r, s, t)$ satisfying $r \geq s \geq t$. Let $F' = U\text{diag}(r, s, 0)V^T$, this method is suggested by Tsai and Huang [19] and has been proven to minimize the Frobenius norm $\|F - F'\|$ as required.

Thus, with the previous description we can now formulate the eight-point algorithm into two main steps for the computation of the fundamental matrix F as follows:

1. *Find Linear Solution*: Given image pairs $m \leftrightarrow m'$, solve the equations $m'^T F m = 0$ to find F . The solution is the eigenvector corresponding to the smallest eigenvalue of $A^T A$ with A the equation matrix.
2. *Enforce Rank Two Constraint*: Replace F by F' , where F' is the closest singular matrix to F under Frobenius norm. This can be done via singular value decomposition.

The algorithm is extremely simple and can be easily implemented, assuming with the availability of a suitable linear algebra library, for instance, Matlab or OpenCV.

4.2.4 The Essential Matrix and Extraction of Relative Pose Between Cameras

In this section we will introduce the definition of the essential matrix and how to find out the relative pose of the two cameras by extracting the essential matrix.

The Essential Matrix

In comparison to the fundamental matrix F , where F satisfies the equations $m'^T F m = 0$ in the homogeneous image(pixel) coordinate. The essential matrix E is defined in the metric coordinate satisfying the similar constraint $x'^T E x = 0$, with $m = Kx$ and K the intrinsic camera matrix. In fact, $E = T \times R = [T]_{\times} R$ with (R, T) the coordinate transformation between the two camera coordinates. In other words, the essential matrix E encodes the relative pose of the two cameras in one matrix with the two cameras intrinsically calibrated(K and K' are known). Compare to the fundamental matrix F with the cameras uncalibrated, the essential matrix and the fundamental matrix have the following relative equation:

$$E = K'^T F K \quad (4.18)$$

The meaning of the equation above is that after we calibrate the two cameras and the fundamental matrix F is calculated with the eight-point algorithm mentioned previously, the essential matrix E can be computed with Equation 4.18, which informs us the relative pose of the two cameras.

Pose Recovery of Cameras By Extracting The Essential Matrix

After the essential matrix E is computed, in the previous section we know that $E = [T]_{\times} R$, which is the matrix that relative pose of the two cameras (R, T) can be extracted from it. In order to extract the coordinate transformation from the essential matrix, the essential matrix E is decomposed with singular value decomposition at first. Let $E =$

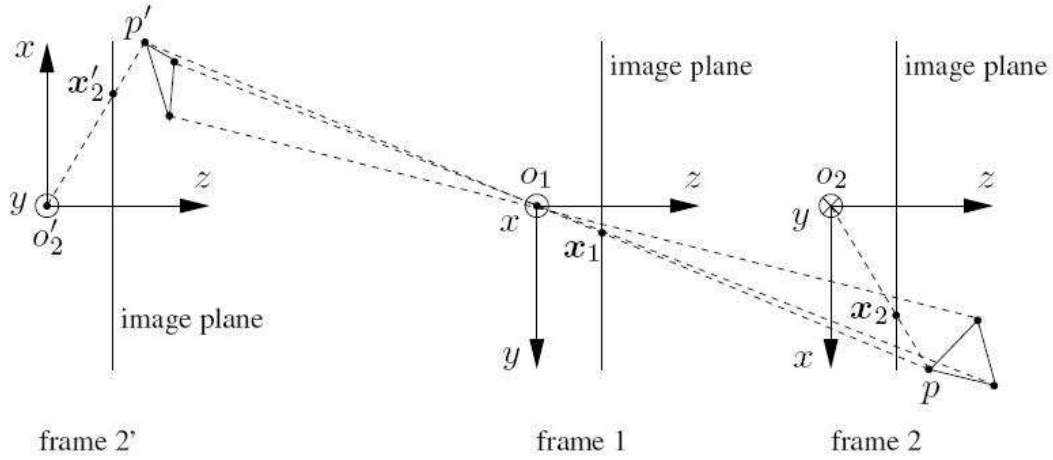


Figure 4.5: **The Pose Recovery Twisted Pair Extracted from The Essential Matrix.** "Two pairs of camera relative poses which generate the same essential matrix. It is shown that one of the two solutions will not satisfy the positive depth constraint. Cited from [22]"

UDV^T , where $D = \text{diag}(a, a, 0)$, define:

$$R_Z\left(+\frac{\pi}{2}\right) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.19)$$

then we can derive four solutions with two rotation matrices R_1, R_2 and two translation matrices in 3×3 cross-vector form T_1, T_2 with the following formula:

$$\begin{aligned} (R_1, T_1) &= (UR_Z^T\left(+\frac{\pi}{2}\right)V^T, UR_Z\left(+\frac{\pi}{2}\right)DU^T) \\ (R_2, T_2) &= (UR_Z^T\left(-\frac{\pi}{2}\right)V^T, UR_Z\left(-\frac{\pi}{2}\right)DU^T) \end{aligned}$$

In the formula above (R_1, T_1) and (R_2, T_2) are called the *twisted pair*. Figure 4.5 shows that one solution of the twisted pair will not satisfy the *positive depth constraint*, which

means all 3D points must lie in front of the image planes of both cameras. In fact, in order to extract the relative poses from the essential matrix with SVD robustly, Wang and Hung [20] have proven that there are four solutions for rotation (two additional solutions for rotation) and two solutions for translation, which permutes into eight solutions for the relative pose between the two cameras. Let:

$$R_{(-Z)}(+\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.20)$$

then after extracting the solutions of pose from the essential matrix we have the following eight solutions:

$$\begin{aligned} (R_1, T_1) &= (UR_Z^T(+\frac{\pi}{2})V^T, UR_Z(+\frac{\pi}{2})DU^T) \\ (R_2, T_2) &= (UR_Z^T(-\frac{\pi}{2})V^T, UR_Z(-\frac{\pi}{2})DU^T) \\ (R_1, T_2) &= (UR_Z^T(+\frac{\pi}{2})V^T, UR_Z(-\frac{\pi}{2})DU^T) \\ (R_2, T_1) &= (UR_Z^T(-\frac{\pi}{2})V^T, UR_Z(+\frac{\pi}{2})DU^T) \\ (R_3, T_1) &= (UR_{(-Z)}^T(+\frac{\pi}{2})V^T, UR_Z(+\frac{\pi}{2})DU^T) \\ (R_3, T_2) &= (UR_{(-Z)}^T(+\frac{\pi}{2})V^T, UR_Z(-\frac{\pi}{2})DU^T) \\ (R_4, T_1) &= (UR_{(-Z)}^T(-\frac{\pi}{2})V^T, UR_Z(+\frac{\pi}{2})DU^T) \\ (R_4, T_2) &= (UR_{(-Z)}^T(-\frac{\pi}{2})V^T, UR_Z(-\frac{\pi}{2})DU^T) \end{aligned}$$

After the above eight solutions are extracted from the essential matrix, six out of the eight solutions can be rejected by using the positive depth constraint and one solution is removed since the relative pose of the two cameras is not reasonable. Thus, after extracting the solutions of the pose recovery problem from the essential matrix with singular value decomposition, we can obtain eight solutions where one of them will be the correct relative pose between the cameras.

4.2.5 Calculation of Depth Information for Structure Reconstruction

With the rigid-body transformation of the cameras (R, T) and the intrinsic camera parameters known, we can calculate 3D coordinates of the paired image points in terms of the image coordinates and depths λ, λ' , let x_i, x_i' be the i th image pair points in homogeneous image coordinate system, the relation between depths and the transformations between two camera coordinates can be formulated as follows:

$$\lambda'_i x_i' = \lambda_i R x_i + \gamma T \quad (4.21)$$

notice that because (R, T) are known, the depths λ 's and the scale of translation γ in Equation 4.21 form a linear system of equations and thus they can be easily solved. For each 3D point, λ and λ' denote its depths with respect to the first and second camera frames, respectively. One of the depths λ or λ' is therefore redundant, for instance, if λ is known, λ' is simply a function of (R, T) . Hence we can eliminate one of the depths, say, λ' by multiplying both sides of the Equation 4.21 by $[x_i']_x$:

$$\lambda [x_i']_x R x_i + \gamma [x_i']_x T = 0 \quad (4.22)$$

this is equivalent to solving the following linear equation:

$$M_i \bar{\lambda}_i = \begin{bmatrix} [x_i']_x R x_i & [x_i']_x T \end{bmatrix} \begin{bmatrix} \lambda_i \\ \gamma \end{bmatrix} = 0 \quad (4.23)$$

where $M_i = \begin{bmatrix} [x_i']_x R x_i & [x_i']_x T \end{bmatrix} \in \mathbf{R}^{3 \times 2}$ and $\bar{\lambda}_i = \begin{bmatrix} \lambda_i \\ \gamma \end{bmatrix} \in \mathbf{R}^2$, for $i = 1, 2, \dots, n$.

Notice that all the n equations above shares the same coefficient γ , we define a vector

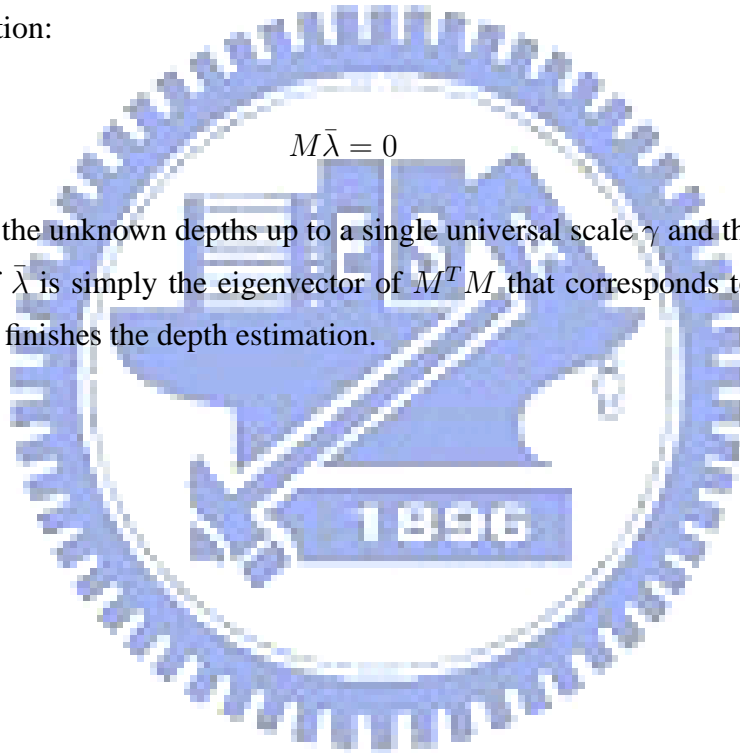
$\bar{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^T \in \mathbf{R}^{n+1}$ and a matrix $M \in \mathbf{R}^{3n \times (n+1)}$ as

$$M = \begin{bmatrix} [x_1']_{\mathbf{x}} R x_1 & 0 & 0 & 0 & 0 & [x_1']_{\mathbf{x}} T \\ 0 & [x_2']_{\mathbf{x}} R x_2 & 0 & 0 & 0 & [x_2']_{\mathbf{x}} T \\ 0 & 0 & [x_3']_{\mathbf{x}} R x_3 & 0 & 0 & [x_3']_{\mathbf{x}} T \\ 0 & 0 & \dots & 0 & 0 & \dots \\ 0 & 0 & 0 & [x_{(n-1)}']_{\mathbf{x}} R x_{(n-1)} & 0 & [x_{(n-1)}']_{\mathbf{x}} T \\ 0 & 0 & 0 & 0 & [x_n']_{\mathbf{x}} R x_n & [x_n']_{\mathbf{x}} T \end{bmatrix}$$

Then the equation:

$$M\bar{\lambda} = 0 \quad (4.24)$$

determines all the unknown depths up to a single universal scale γ and the linear least-square solution of $\bar{\lambda}$ is simply the eigenvector of $M^T M$ that corresponds to the smallest eigenvalue, where finishes the depth estimation.



Chapter 5

The Thin-Plate Splines for 3D Surface Modeling



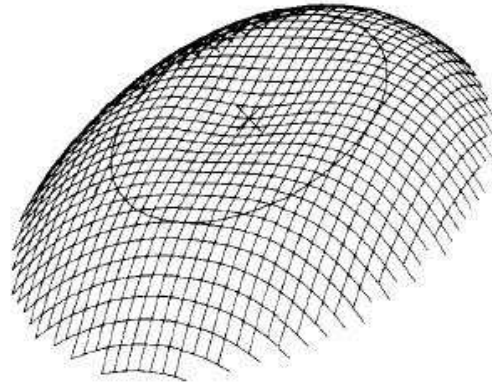


Figure 5.1: **Radio Basis Function of the Thin-Plate Splines in Two-Dimensional Space.**
 "This figure is cited from [2]"

The *Thin-Plate Splines* (TPSs) are often used for interpolating surfaces over scattered data because of its elegant algebra expressing the dependence of the physical bending energy of a thin metal plate on point constraints. For interpolation of a surface over a fixed set of sparse points in the plane, the bending energy is a quadratic form in the heights assigned to the surface. After calculating the 3D points of the image correspondences, the 3D data can be regarded as a sparse point cloud which spread over an certain area of the virtual scene. Therefore, we can use the TPS to interpolate the surface including all the reconstructed sparse 3D points. In recent years, the thin-plate splines is used for biological deformations [2] since the interpolation results of the TPSs may be suitable for analysing biological structures. In the following sections we will introduce the formation and algebra of the thin-plate splines.

5.1 The Radio Basis Function(RBFs)

The splines are all expanded by their basis functions and therefore different basis functions contribute to the variation of the splines. The thin-plate splines are expanded by the

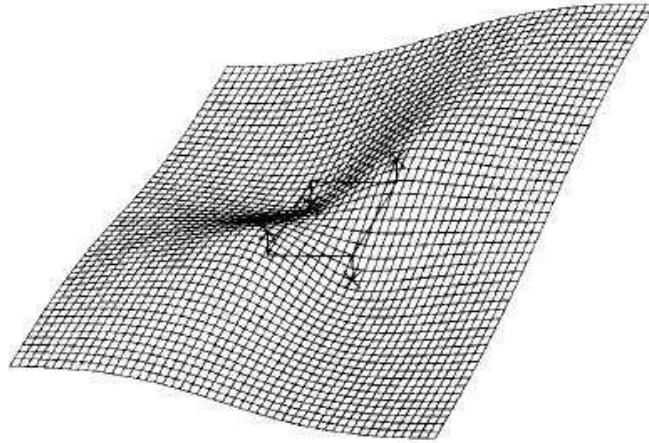


Figure 5.2: **A Mathematical Model of A Thin Steel Plate.** "This figure is cited from [2]"

radio basis functions. For instance, in two-dimensional space the radio basis function is defined as follows:

$$z(x, y) = -U(r) = -r^2 \log r^2 \quad (5.1)$$

where r is the distance $\sqrt{x^2 + y^2}$ from the Cartesian origin. The function is zero along the indicated circle in Figure 5.1, where $r = 1$. The radio basis function satisfies the following equation:

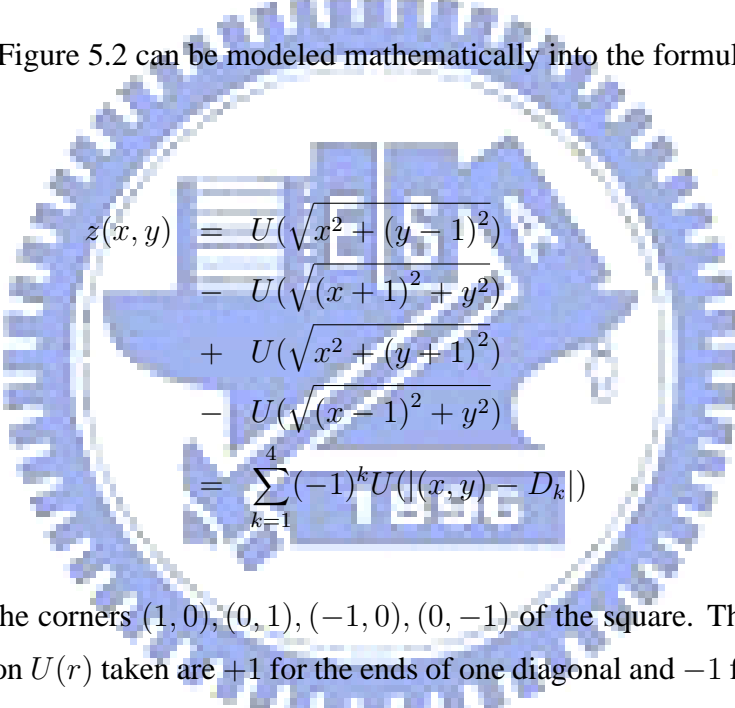
$$\Delta^2 U = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)^2 \quad (5.2)$$

In addition, U is a so-called *fundamental solution* to the *biharmonic equation* $\Delta^2 U = 0$, the equation for the shape of a thin steel plate can be shown as a function $z(x, y)$ above the (x, y) -plane.

5.2 Bounded Linear Combinations of Radio Basis Functions

Figure 5.2 illustrates a mathematical model of a thin steel plate which is in fact extending to infinite space in all directions. Passing through this plate is a rigid skeleton of square size with its side length $\sqrt{2}$, drawn as the rhombus at the center of the figure. The steel is fixed in position in some distance above two diagonally opposite corners of the square and the same distance below the other two corners of the square.

The surface in Figure 5.2 can be modeled mathematically into the formula as follows:



$$\begin{aligned}
 z(x, y) &= U(\sqrt{x^2 + (y-1)^2}) \\
 &\quad - U(\sqrt{(x+1)^2 + y^2}) \\
 &\quad + U(\sqrt{x^2 + (y+1)^2}) \\
 &\quad - U(\sqrt{(x-1)^2 + y^2}) \\
 &= \sum_{k=1}^4 (-1)^k U(|(x, y) - D_k|)
 \end{aligned}$$

where D_k are the corners $(1, 0)$, $(0, 1)$, $(-1, 0)$, $(0, -1)$ of the square. The coefficients having with function $U(r)$ taken are $+1$ for the ends of one diagonal and -1 for the ends of the other. As one travels far away from the origin, this plate is asymptotically flat and level in all directions. For instance, in Figure 5.2, the corner of the plate facing the viewer in the diagram has apparently become nearly level somewhat underneath the level of constraint at the nearest corner of the square, and the condition is similar to the other three corners.

The displacement of the thin plate in Figure 5.2 lies in a direction orthogonal to the plate itself. We can imagine that the displacements $z(x, y)$ is applied directly to one or both of the coordinates of x or y -axis of the plate with which we started. Thus we may interpret the scheme of Figure 5.2 as the *interpolation function*. Thus we can formulate the mapping function of the interpolation as follows:

$$(x, y) \rightarrow (x', y') = (x, y + z(x, y)) \quad (5.3)$$

In this manner the thin-plate spline we have been examining can be used to solve a two-dimensional interpolation problem as the computation of a map $\mathbf{R}^2 \rightarrow \mathbf{R}^2$ from sparse arbitrary data. Likewise, we can interpolate 3D data as well using the thin-plate splines.

5.3 Algebra of the Thin-Plate Splines

In order the model the surface over sparse 3D reconstructed feature points using the thin-plate splines, the following text of this section focus on the overview of the algebraic form of the thin-plate spline method. Let $P_1 = (x_1, y_1, z_1), P_2 = (x_2, y_2, z_2), \dots, P_n = (x_n, y_n, z_n)$ be n points in the Euclidean coordinate. Define $r_{ij} = |P_i - P_j|$ as the distance between points i and j and the following matrices:

$$K = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \cdots & \cdots & \cdots & \cdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix}, n \times n$$

$$P = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & y_n & z_n \end{bmatrix}, n \times 4$$

$$L = \begin{bmatrix} K & P \\ P^T & O \end{bmatrix}, (n+4) \times (n+4)$$

where O is a 4×4 matrix consists of zeros.

Let $V = (v_1, v_2, \dots, v_n)$ be any arbitrary n -vector and have $Y = (V|0\ 0\ 0\ 0)^T$, which is a column vector of length $n+4$. Define the vector $W = (w_1, \dots, w_n)$ and the coefficients a_1, a_x, a_y, a_z by the equation:

$$L^{-1}Y = (W|a_1, a_x, a_y, a_z)^T \quad (5.4)$$

Use the elements of $L^{-1}Y$ to define a function $f(x, y, z)$ everywhere in the Euclidean 3D space:

$$f(x, y, z) = a_1 + a_x + a_y + a_z + \sum_{i=1}^n w_i U(|P_i - (x, y, z)|) \quad (5.5)$$

We take the points (x_i, y_i, z_i) to be our source landmarks and V to be the $n \times 3$ target matrix:

$$V = \begin{bmatrix} x_1' & x_2' & \cdots & x_n' \\ y_1' & y_2' & \cdots & y_n' \\ z_1' & z_2' & \cdots & z_n' \end{bmatrix}$$

where every (x_i', y_i', z_i') is the landmark homologous to (x_i, y_i, z_i) mapped by the thin-plate spline function in the Euclidean 3D space \mathbf{R}^3 . The resulting function $f(x, y, z) = [f_x(x, y, z), f_y(x, y, z), f_z(x, y, z)]$ is vector-valued: it maps each 3D point (x, y, z) to its corresponding result (x', y', z') . These vector-valued functions $f(x, y, z)$ are the *thin-plate spline mappings*.

5.4 The Wendland Radial Basis Function

The radial basis function of Wendland is shown below:

$$\psi_{d,k}(r) = I^k (1 - r)_+^{\lfloor \frac{d}{2} \rfloor + k + 1}(r) \quad (5.6)$$

Therefore, we can use the functionality of the Wendland radial basis function after building point correspondences in the 3D virtual space to interpolate the surface among the computed discrete and sparse 3D points.

Chapter 6

Our Image-Based 3D Environment

Reconstruction Procedure and Results



In this section we focus on the main steps of our 3D environment reconstruction procedure. First of all, we compute the intrinsic parameters of the two perspective pan-tilt-zoom (PTZ) cameras by using Zhang's method [23]. After the cameras are calibrated, we capture the photos with the two cameras in different point of view and specify the image correspondences manually. With the image correspondences we can compute the fundamental matrix and in addition the essential matrix by combining camera intrinsic parameters and the fundamental matrix. Then we can extract the relative pose of the two cameras from the essential matrix. Positive depth constraint is used for choosing the correct solution from the extraction of the essential matrix. After the relative pose is obtained, we can calculate the 3D coordinates of the image pairs, which is a sparse distributed point cloud in the virtual scenes.

In order to make the computed 3D points into a surface, we use the algebra of the thin-plate splines by creating some source-and-target mapping, we merge the reconstructed distinct scene surfaces with the aid of calibrated pan-tilt informations. Finally, after the procedure mentioned previously is done, with texture mapping we can browse the virtual 3D model of the environment from any arbitrary point of view.

6.1 Perspective Camera Calibration with a 2D Plane

In this section we will describe how we calibrate our two perspective cameras by using Zhang's method. The method is really a milestone when proposed in the year 2000 since 2D planar objects is proven to be able to be used for camera calibration. The calibration procedure consists of a closed-form solution, followed by an nonlinear refinement of maximum likelihood criterion.

The calibration procedure recommended by Zhang [23] is as follows:

1. Print a pattern and attach the pattern to a planar surface.
2. Take some images of the planar pattern under different orientations and positions by moving either the plane or the camera.

<i>Computational Part</i>		
<i>CPU</i>	AMD 2200+ 1.8GHz	
<i>RAM</i>	1.0 GB	
<i>OS</i>	Windows XP	
Programming Tool	Microsoft Visual Studio 6.0	
<i>Image Acquiring Part</i>		
PanoServer 3000		
Video Input	4 channels, 120 fps, NTSC	
Video Output	VGA: 640 x 480 at 60Hz, D-SUB 15 pin	
Size	427(W) x 88.5(H) x 366.6(D) m.m.	
Pan-tilt-zoom Camera		
Signal System	NTSC	PAL
Effective Pixels(HxV)	768 x 494	752 x 582
Imaging Area	4.9mm x 3.7mm	
Image Pickup Device	1/4"-type SuperHAD CCD	
Mechanism		
Panning Range	360 degrees endless	
Tilting Range	92 degrees	
Pan-tilt Accuracy	Cumulative error less than 0.6 degree per 10000 rotations	
Panning Speed Manual	0.1-90 degree/sec	
Panning Speed Preset	300 degree/sec	
Tilting Speed Manual	0.1-45 degree/sec	
Tilting Speed Preset	200 degree/sec	

Table 6.1: **The Specification of Our 3D Environment Reconstruction System.** "Cited from http://www.messo.com/product/Products_Model_Spec.aspx?ModelId=51."

3. Detect the features in the images(usually done with image processing techniques).
4. Estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution.
5. Estimate the coefficients of the radial distortion.
6. Refine all parameters by minimizing the reprojection error.

Since any camera usually exhibits lens distortion, we assume k_1, k_2 to be the *radio distortion coefficients* and p_1, p_2 to be the *tangential distortion coefficients* and model this two kinds of distortion using the following formula:

$$\begin{aligned}\tilde{x} &= x + x(k_1r^2 + k_2r^4) + (2p_1xy + p_2(r^2 + 2x^2)) \\ \tilde{y} &= y + y(k_1r^2 + k_2r^4) + (2p_2xy + p_1(r^2 + 2y^2))\end{aligned}$$

where $r^2 = x^2 + y^2$, (x, y) the ideal(distortion-free) and (\tilde{x}, \tilde{y}) the real(distorted) image physical coordinates. The formula shown above can be transformed into pixel coordinates via camera intrinsic parameters, assume that u, v is the image point corresponding the x, y in the pixel coordinate, Since by definition, $u = xf_x + c_x$ and $v = yf_y + c_y$, the above formula can be rewritten as follows:

$$\begin{aligned}\tilde{u} &= u + (u - c_x)(k_1r^2 + k_2r^4 + 2p_1y + p_2(\frac{r^2}{x} + 2x)) \\ \tilde{v} &= v + (v - c_y)(k_1r^2 + k_2r^4 + 2p_2x + p_1(\frac{r^2}{y} + 2y))\end{aligned}$$

Our camera calibration method differs from the Zhang's method in only the image processing and the feature extraction part. We use a planar pattern with circles printed on it and find the centers of the circles to avoid errors occurred by corner detection due to the noise in images. We compute the coordinates by extracting the center of the circles printed on our planar calibration pattern with the following steps:

1. Transform the color image into intensity(gray scale) image.
2. Binarize the tranformed intensity image with the threshold calculated by Otsu's method [11].
3. Specify the blobs by finding connected components and determine whether the blob is a circle or not, if the blob is a circle then preserve it as a grid point.
4. Use ransac grid alignment to recover the grid points of the calibration pattern.
5. Define the coordinates of the grid points and perform Zhang's camera calibration method.

6.2 Extraction of Relative Pose Between Cameras from Images

After calibrating the two pan-tilt-zoom cameras, the intrinsic parameters of both the cameras are all known. Therefore, we can compute the fundamental matrix and then combine the two intrinsic matrices in order to obtain the essential matrix. After the essential matrix is calculated, eight solutions can be extracted as described in Section 4.2.4 and positive depth constraint and straightward visualization can be used to eliminate the incorrect relative poses extracted from the essential matrix which was introduced in Section 4.2.5.

6.3 Sparse 3D Points Reconstruction

With the rigid-body transformation of the cameras (R, T) and the intrinsic camera parameters known, we can calculate 3D coordinates of the paired image points in terms of the image coordinates and depths λ, λ' , let x_i, x_i' be the i th image pair points in homogeneous image coordinate system, the relation between depths and the transformations between two camera coordinates can be formulated as follows:

$$\lambda'_i x'_i = \lambda_i R x_i + \gamma T \quad (6.1)$$

notice that because (R, T) are known, the depths λ 's and the scale of translation γ in Equation 4.21 form a linear system of equations and thus they can be easily solved. For each 3D point, λ and λ' denote its depths with respect to the first and second camera frames, respectively. One of the depths λ or λ' is therefore redundant, for instance, if λ is known, λ' is simply a function of (R, T) . Hence we can eliminate one of the depths, say, λ' by multiplying both sides of the Equation 4.21 by $[x'_i]_{\mathbf{x}}$:

$$\lambda [x'_i]_{\mathbf{x}} R x_i + \gamma [x'_i]_{\mathbf{x}} T = 0 \quad (6.2)$$

this is equivalent to solving the following linear equation:

$$M_i \bar{\lambda}_i = \begin{bmatrix} [x'_i]_{\mathbf{x}} R x_i & [x'_i]_{\mathbf{x}} T \end{bmatrix} \begin{bmatrix} \lambda_i \\ \gamma \end{bmatrix} = 0 \quad (6.3)$$

where $M_i = \begin{bmatrix} [x'_i]_{\mathbf{x}} R x_i & [x'_i]_{\mathbf{x}} T \end{bmatrix} \in \mathbf{R}^{3 \times 2}$ and $\bar{\lambda}_i = \begin{bmatrix} \lambda_i \\ \gamma \end{bmatrix} \in \mathbf{R}^2$, for $i = 1, 2, \dots, n$.

Notice that all the n equations above shares the same coefficient γ , we define a vector $\bar{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^T \in \mathbf{R}^{n+1}$ and a matrix $M \in \mathbf{R}^{3n \times (n+1)}$ as

$$M = \begin{bmatrix} [x'_1]_{\mathbf{x}} R x_1 & 0 & 0 & 0 & 0 & [x'_1]_{\mathbf{x}} T \\ 0 & [x'_2]_{\mathbf{x}} R x_2 & 0 & 0 & 0 & [x'_2]_{\mathbf{x}} T \\ 0 & 0 & [x'_3]_{\mathbf{x}} R x_3 & 0 & 0 & [x'_3]_{\mathbf{x}} T \\ 0 & 0 & \dots & 0 & 0 & \dots \\ 0 & 0 & 0 & [x'_{(n-1)}]_{\mathbf{x}} R x_{(n-1)} & 0 & [x'_{(n-1)}]_{\mathbf{x}} T \\ 0 & 0 & 0 & 0 & [x'_n]_{\mathbf{x}} R x_n & [x'_n]_{\mathbf{x}} T \end{bmatrix}$$

Then the equation:

$$M \bar{\lambda} = 0 \quad (6.4)$$

determines all the unknown depths up to a single universal scale γ and the linear least-square solution of $\bar{\lambda}$ is simply the eigenvector of $M^T M$ that corresponds to the smallest eigenvalue, where finishes the depth estimation. After the depths are computed for every image pair, we can use depth information from one of the camera frames, say, λ_1 , for calculating their corresponding 3D coordinates, which completes the sparse 3D reconstruction.

6.4 3D Environment Surface Modeling and Texture Mapping Using Thin-Plate Splines

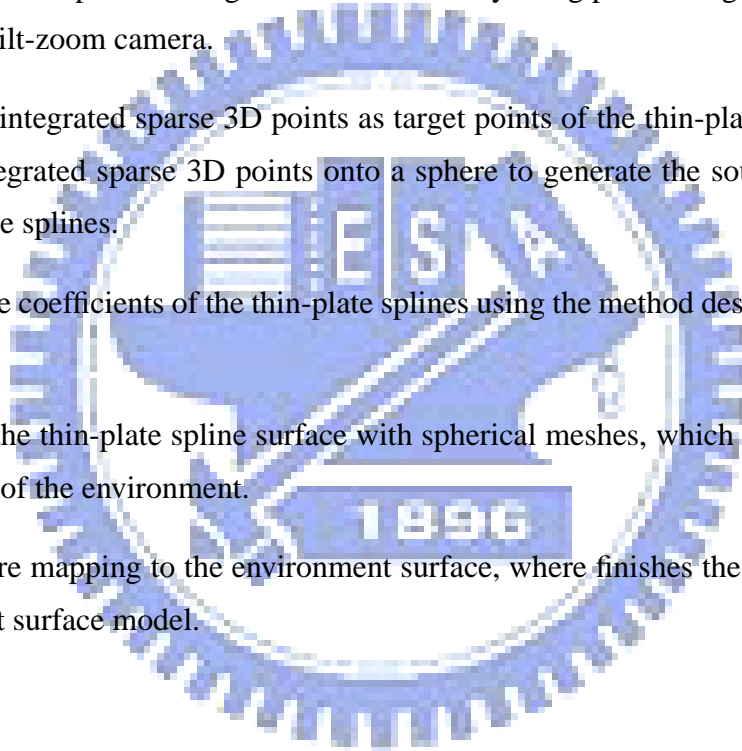
After the sparse 3D point cloud is reconstructed, we can interpolate a surface in 3D space by using the thin-plate splines. Since the coefficients of the thin-plate splines can be calculated from the corresponding source and target points in the 3D coordinate system, we specify the reconstructed 3D points as targets and generate the source points by mapping them onto a sphere surface. Once the source points and the target points are determined, the coefficient of the thin-plate splines in 3D space can be calculated and then we can interpolate a nonlinear surface over the sparse 3D points which all located exactly on the interpolated thin-plate spline surface.

In order to reconstruct the whole environment, we combine the eight-point algorithm for reconstructing particular scenes and the calibrated pan-tilt angles of the two pan-tilt-zoom cameras. Therefore, our reconstruction procedure is based on the following assumptions:

1. The mechanic model parameters (e.g. pan angle, tilt angle) of the two pan-tilt-zoom cameras are calibrated accurately.
2. The pan axis and tilt axis of the pan-tilt-zoom cameras are orthogonal.
3. The transformation of different views from the same pan-tilt-zoom camera with various positions and orientations consists of only pure rotation. In other words, the focal length is the radius of the pan-tilt sphere and the lense rotates with its center exactly on the sphere center.

With the assumptions listed above, we can use the pan and tilt angles of a pan-tilt-zoom camera to calculate relative poses of the individually reconstructed scenes and then combine them altogether using the thin-plate splines with source points mapped onto a sphere. Our reconstruction procedure is listed as follows:

1. Reconstruct point clouds of reconstructable scenes in the virtual 3D space individually with two intrinsically calibrated cameras.
2. Calculate relative poses among individual scenes by using pan-tilt angles of the calibrated pan-tilt-zoom camera.
3. Specify the integrated sparse 3D points as target points of the thin-plate splines and map the integrated sparse 3D points onto a sphere to generate the source points of the thin-plate splines.
4. Calculate the coefficients of the thin-plate splines using the method described in Section 5.3.
5. Interpolate the thin-plate spline surface with spherical meshes, which forms the virtual surface of the environment.
6. Apply texture mapping to the environment surface, where finishes the final textured environment surface model.



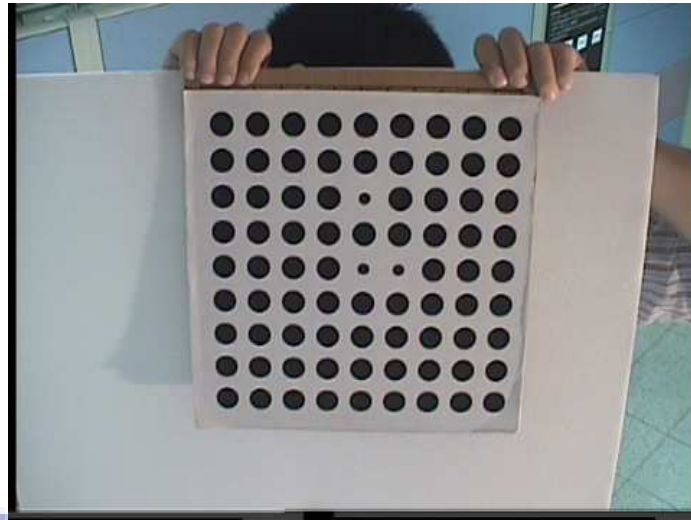


Figure 6.1: **Our 2D Planar Camera Calibration Pattern.** "The pattern consists of circles in order to reduce image processing errors due to the noise by finding the centers of the circles."

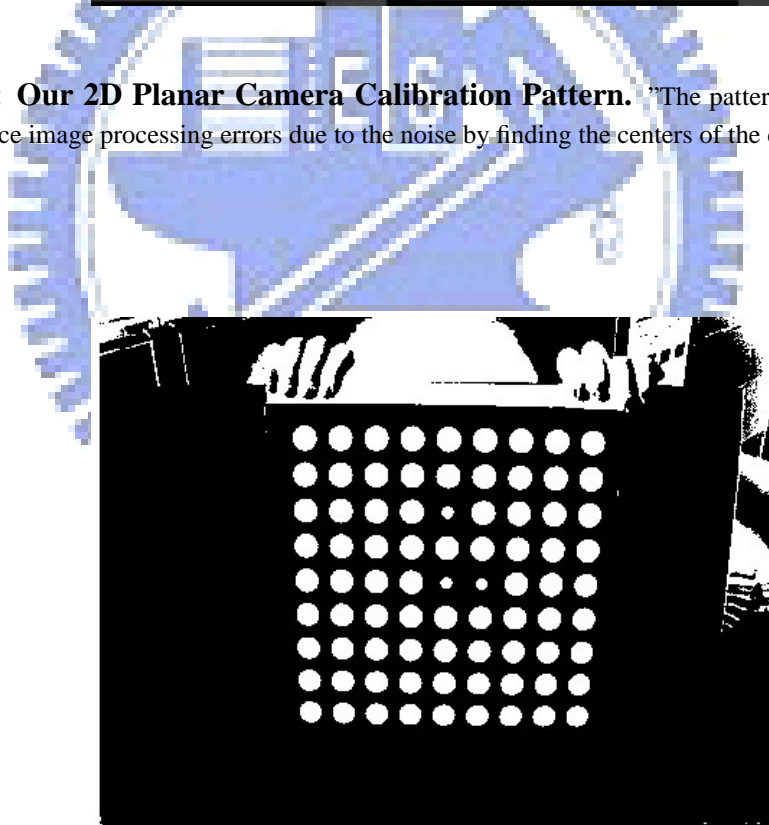


Figure 6.2: **Result Image After Binarization.**

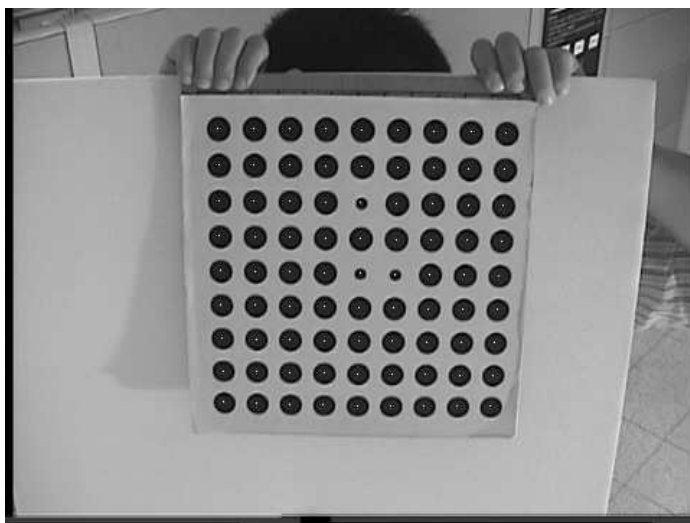


Figure 6.3: Result Image After Grid Alignment.

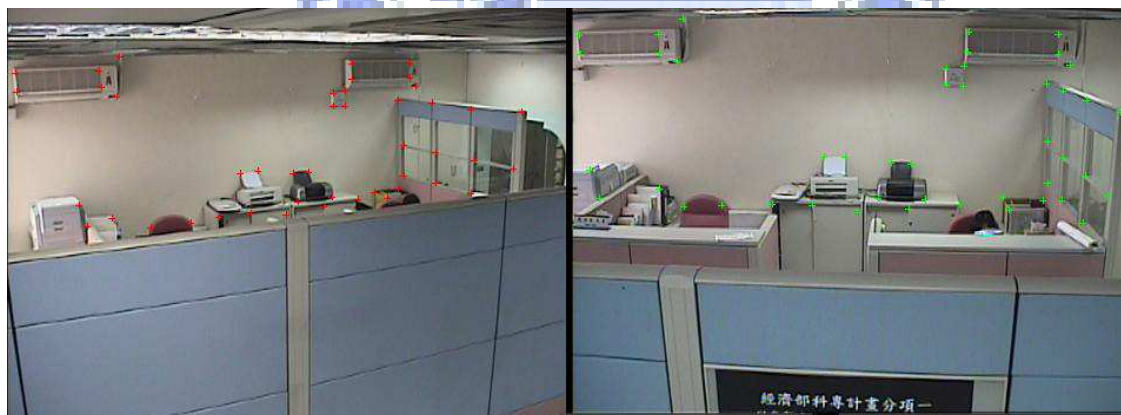


Figure 6.4: Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 1.

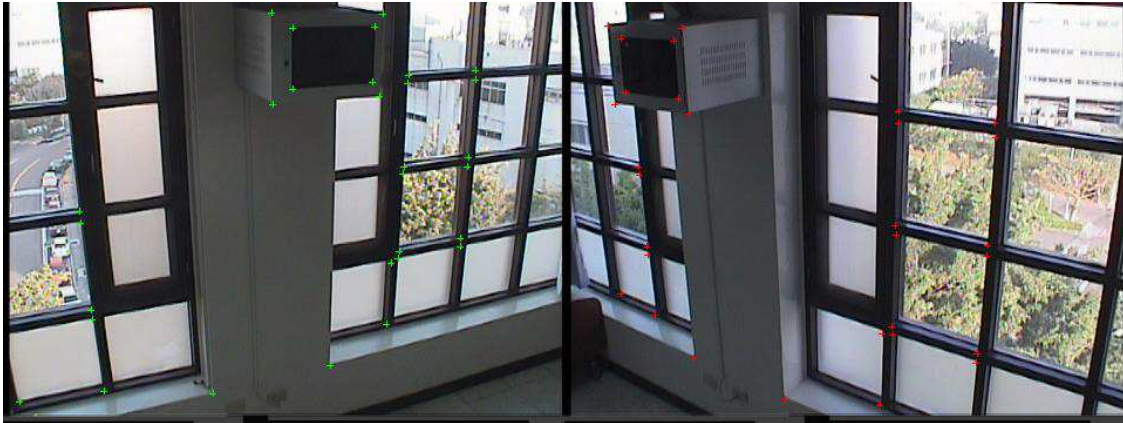


Figure 6.5: Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 2.



Figure 6.6: Multi-view Photos For 3D Environment Reconstruction With Corresponding Image Points Specified - Scene 3.

Camera Intrinsic Parameters	Camera A	Camera B
α	556.480999	559.695405
β	565.233869	569.268765
γ	0.0	0.0
u_0	239.572583	232.915437
v_0	179.785209	177.405891
Distortion Coefficients	Camera A	Camera B
k_1	-0.261327	-0.271668
k_2	0.290120	0.324966
p_1	-0.001173	-0.000919
p_2	-0.001431	0.000440
Average Reprojection Error	0.149034	0.164551

Table 6.2: The Calibrated Pan-Tilt-Zoom Camera Intrinsic Parameters Using Our Calibration Method.



Figure 6.7: Scene 1 - Epipolar Lines Calculated With The Fundamental Matrix.

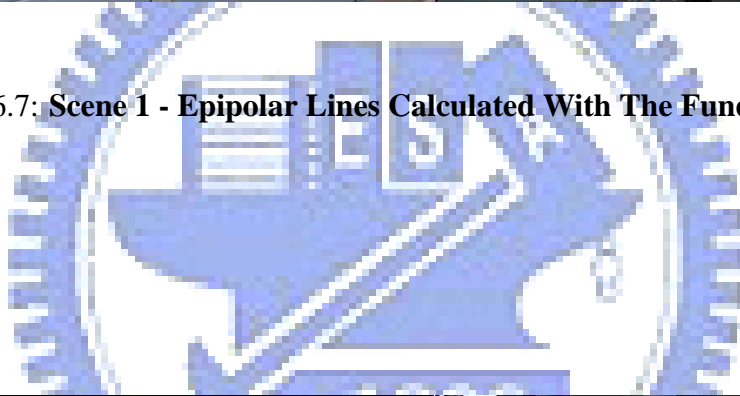


Figure 6.8: Scene 2 - Epipolar Lines Calculated With The Fundamental Matrix.

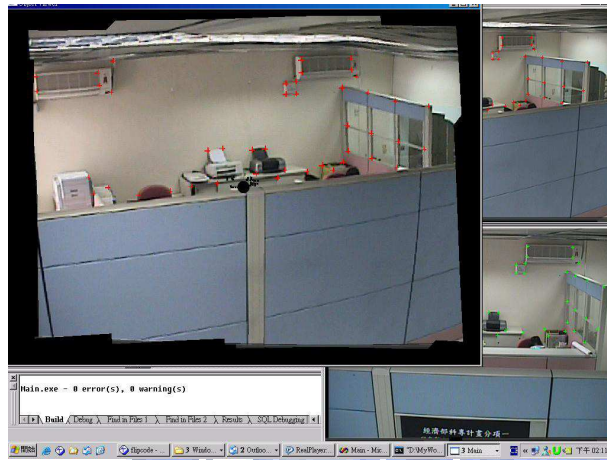


Figure 6.9: Scene 1 Viewpoint 1 - Individually Reconstructed Surface Model.

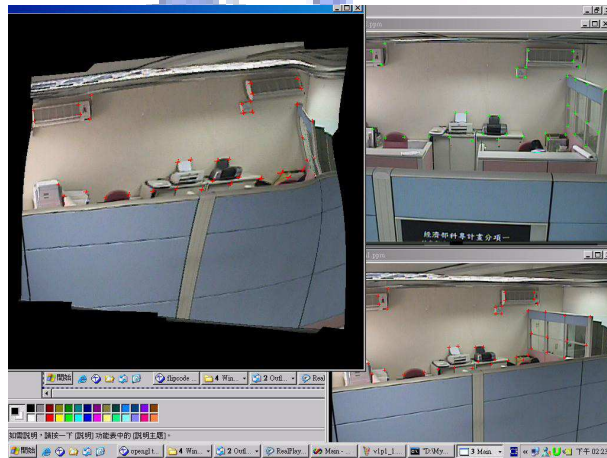


Figure 6.10: Scene 1 Viewpoint 2 - Individually Reconstructed Surface Model.

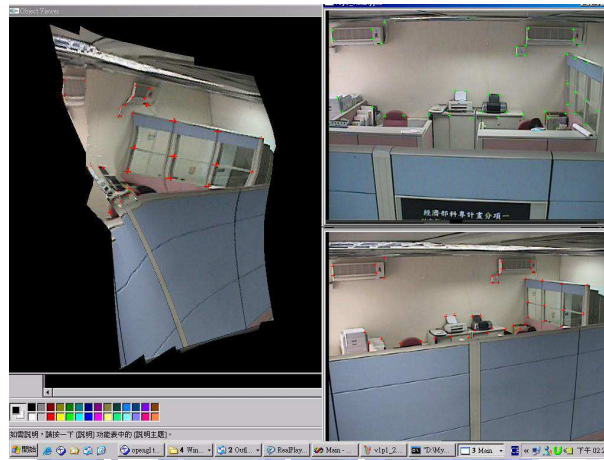


Figure 6.11: Scene 1 Viewpoint 3 - Individually Reconstructed Surface Model.

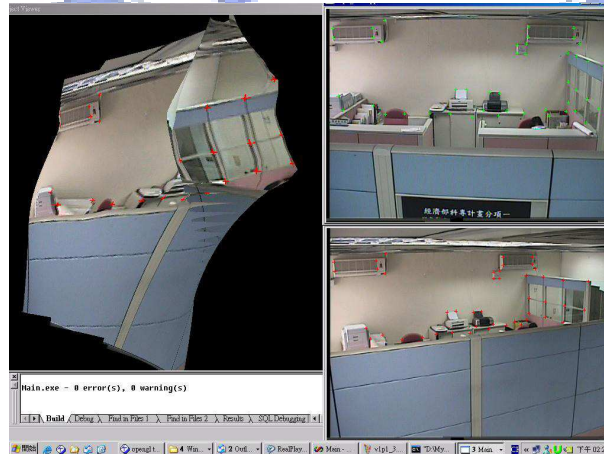


Figure 6.12: Scene 1 Viewpoint 4 - Individually Reconstructed Surface Model.

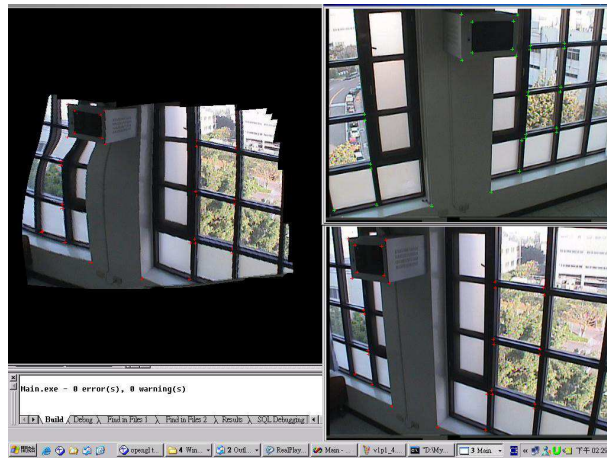


Figure 6.13: Scene 2 Viewpoint 1 - Individually Reconstructed Surface Model.

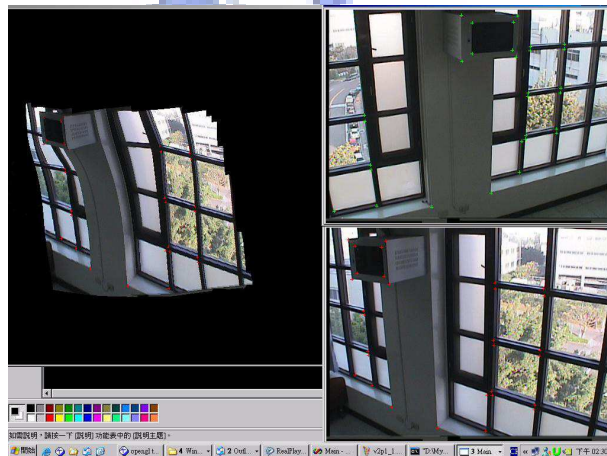


Figure 6.14: Scene 2 Viewpoint 2 - Individually Reconstructed Surface Model.

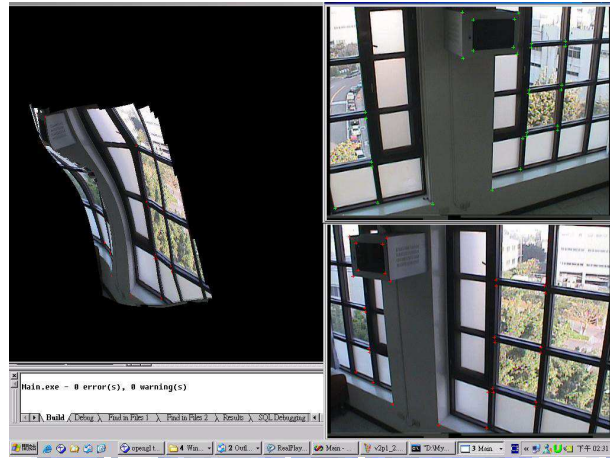


Figure 6.15: Scene 2 Viewpoint 3 - Individually Reconstructed Surface Model.

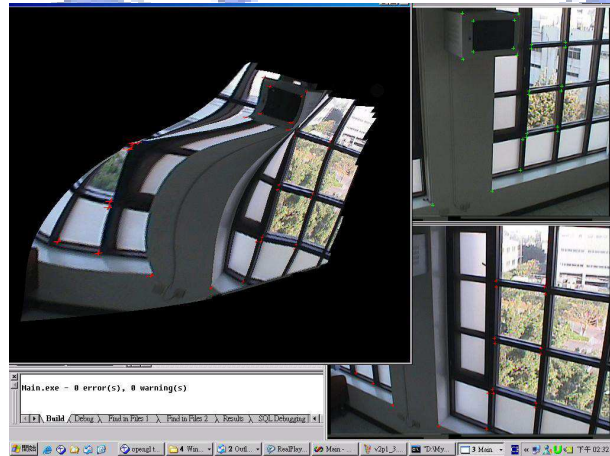


Figure 6.16: Scene 2 Viewpoint 4 - Individually Reconstructed Surface Model.



Figure 6.17: Scene 3 Viewpoint 1 - Individually Reconstructed Surface Model.



Figure 6.18: Scene 3 Viewpoint 2 - Individually Reconstructed Surface Model.

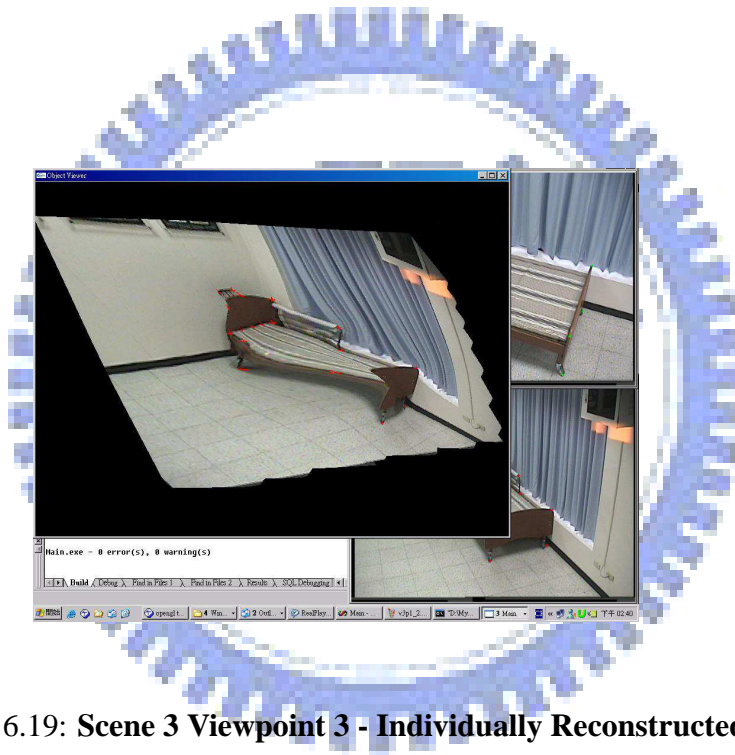


Figure 6.19: Scene 3 Viewpoint 3 - Individually Reconstructed Surface Model.

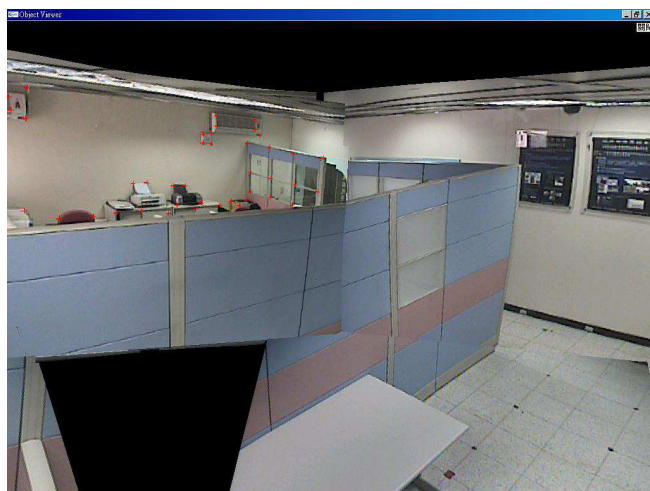


Figure 6.20: **Viewpoint 1 - Reconstructed Environment Using Our Method.**

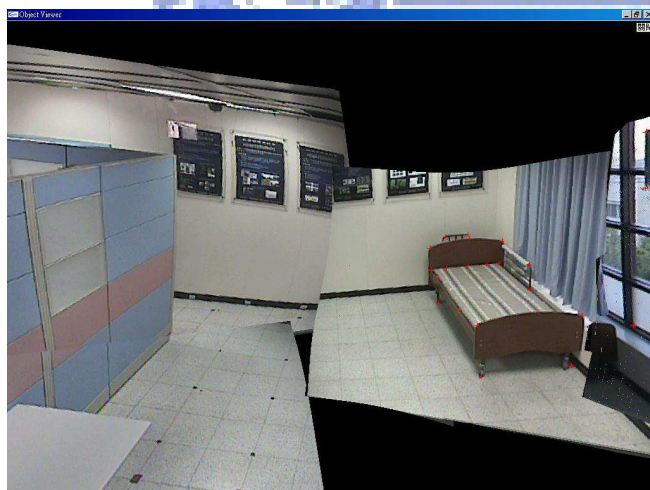


Figure 6.21: **Viewpoint 2 - Reconstructed Environment Using Our Method.**

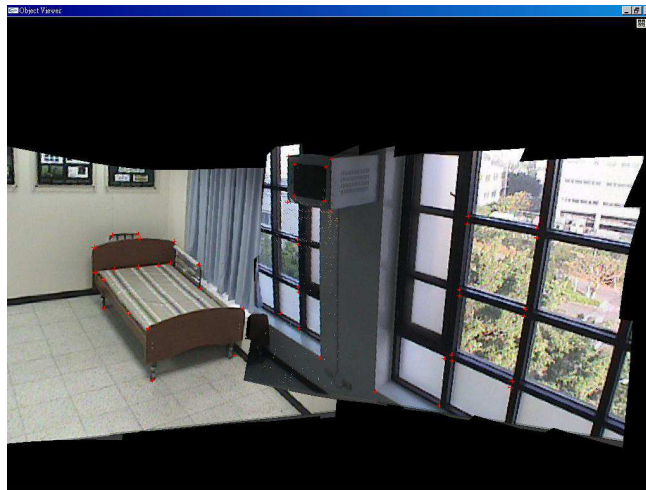


Figure 6.22: Viewpoint 3 - Reconstructed Environment Using Our Method.

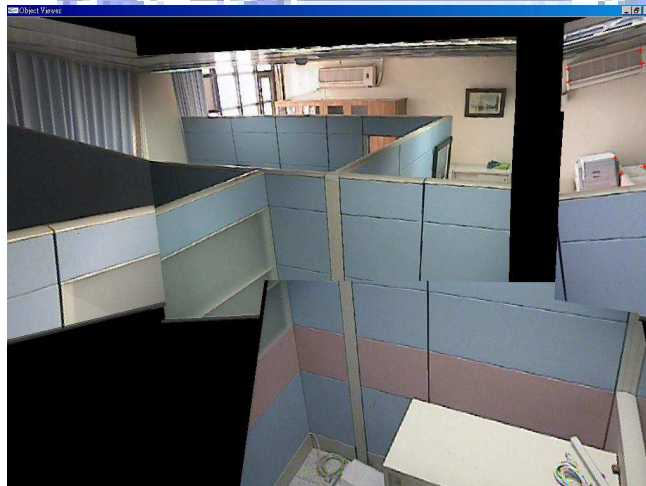


Figure 6.23: Viewpoint 4 - Reconstructed Environment Using Our Method.



Figure 6.24: **Viewpoint 5 - Reconstructed Environment Using Our Method.**

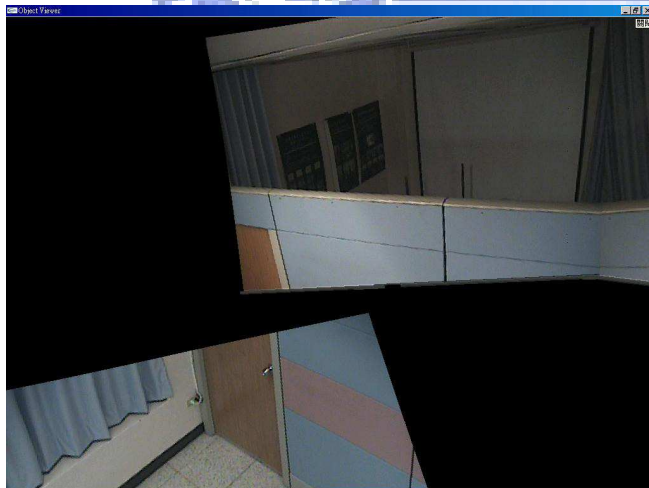


Figure 6.25: **Viewpoint 6 - Reconstructed Environment Using Our Method.**

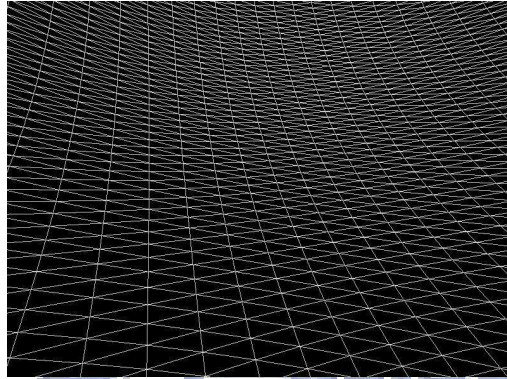


Figure 6.26: Wireframe - Projecting all the synthesized views onto a sphere.

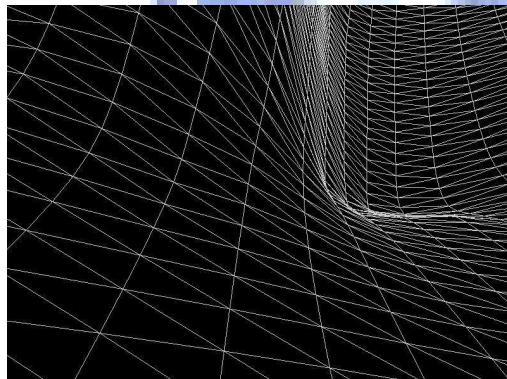


Figure 6.27: Wireframe - Reconstructed part of all the synthesized views.



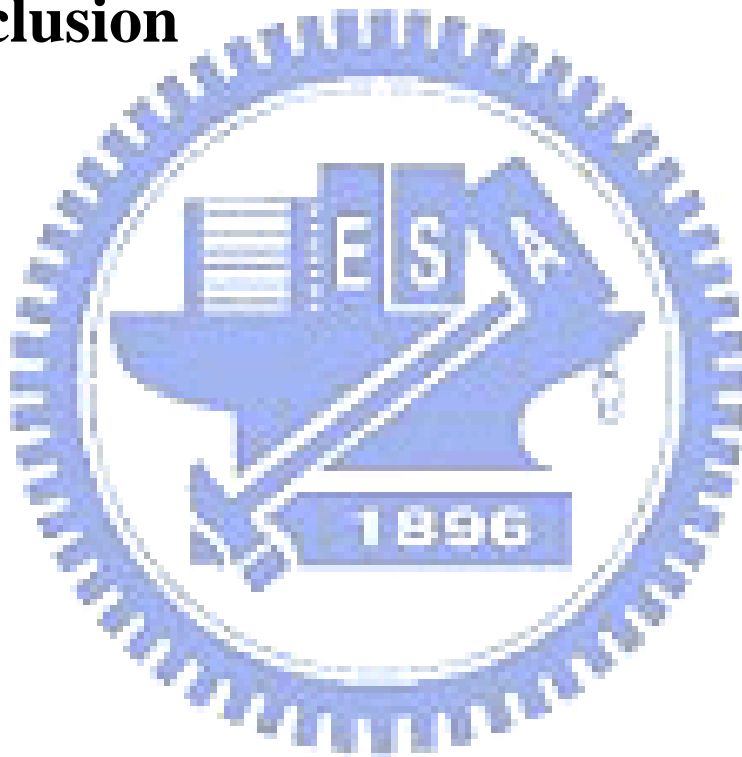
Figure 6.28: **Textured model - Projecting all the synthesized views onto a sphere.**



Figure 6.29: **Textured model - Reconstructed part of all the synthesized views.**

Chapter 7

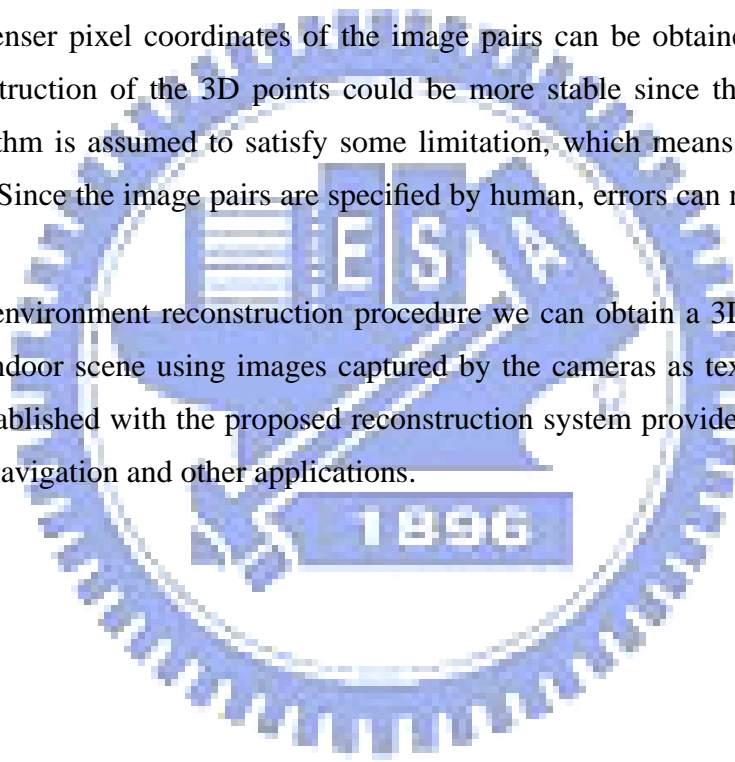
Conclusion



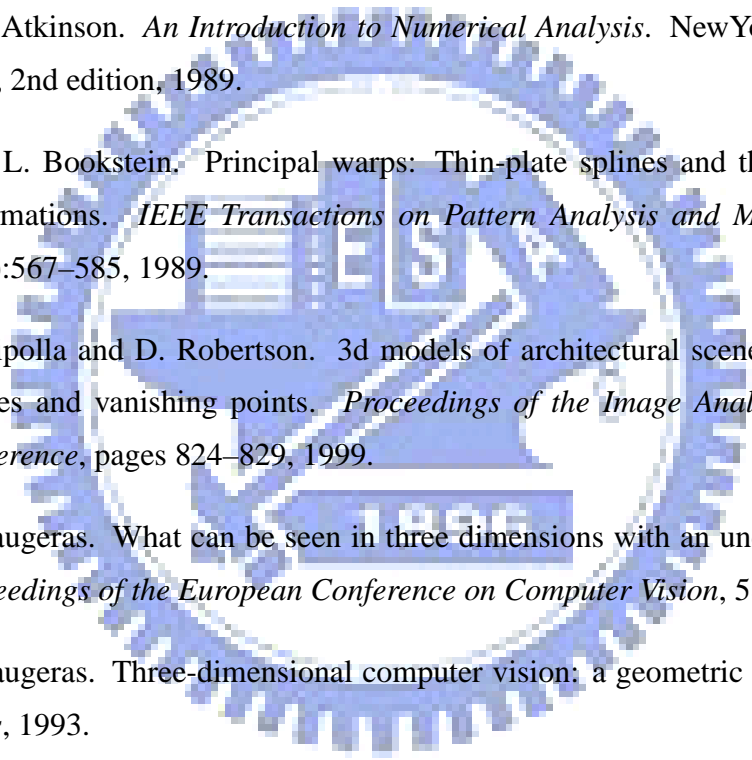
In this thesis a 3D indoor environment reconstruction system is implemented by using Wendland to interpolate the surface of the reconstructed 3D point cloud and then perform texture mapping. Relative pose between cameras are extracted in our 3D reconstruction steps which may assist robot navigation greatly. Since robotic errors are often accumulated while the robot is moving, re-coordinating the robot by calculating relative pose between cameras with images results in less error if the robot moves further. By integrating the robot system and the computer vision system may control the robot more precisely.

Extraction and matching of the image correspondences mainly influence 3D reconstruction results. If denser pixel coordinates of the image pairs can be obtained more accurately, our reconstruction of the 3D points could be more stable since the input of the eight-point algorithm is assumed to satisfy some limitation, which means to be random sampled enough. Since the image pairs are specified by human, errors can not be avoided in this case.

With our 3D environment reconstruction procedure we can obtain a 3D environment model of a real indoor scene using images captured by the cameras as texture. The 3D surface model established with the proposed reconstruction system provides useful information for robot navigation and other applications.



Bibliography

- 
- [1] K.E. Atkinson. *An Introduction to Numerical Analysis*. New York: John Wiley and Sons, 2nd edition, 1989.
- [2] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.
- [3] R. Cipolla and D. Robertson. 3d models of architectural scenes from uncalibrated images and vanishing points. *Proceedings of the Image Analysis and Processing Conference*, pages 824–829, 1999.
- [4] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. *Proceedings of the European Conference on Computer Vision*, 588:563–578, 1992.
- [5] O. Faugeras. Three-dimensional computer vision: a geometric viewpoint. *The MIT Press*, 1993.
- [6] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16:35–56, Sep 1995.
- [7] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. *Proceedings of the European Conference on Computer Vision*, 588:579–587, 1992.
- [8] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *The Cambridge University Press*, 2000.

- [9] Q.T. Luong. Matrice fondamentale et autocalibration en vision par ordinateur. *PhD Thesis*, 1992.
- [10] Q.T. Luong. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.
- [11] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, (1):62–66, January 1979.
- [12] P. Rander P. Narayanan and T. Kanade. Constructing virtual worlds using dense stereo. *Proceedings of the International Conference on Computer Vision*, 1998.
- [13] M. Pollefeys and L. Van Gool. From images to 3d models. *Communications of the ACM*, 45(7):50–55, 2002.
- [14] Marc Pollefeys. Visual 3d modeling from images. *Vision, Modeling and Visualization*, 2004.
- [15] O. Schreer. Stereo vision-based navigation in unknown indoor environments. *Proceedings of the European Conference on Computer Vision*, June 1998.
- [16] P. Torr and A. Zisserman. Robust vision. *Proceedings of the British Machine Vision Conference*, 1994.
- [17] P. Torr and A. Zisserman. Motion segmentation and outlier detection. *PhD Thesis*, 1995.
- [18] P. Torr and A. Zisserman. Robust computation and parameterization of multiple view relations. *Proceedings of the International Conference on Computer Vision*, pages 727–732, 1998.
- [19] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 13–27, 1984.

- [20] Wei Wang and Hung-Tat Tsui. An svd decomposition of essential matrix with eight solutions for the relative positions of two perspective cameras. *Proceedings of the International Conference on Pattern Recognition*, pages 1362–1365, 2000.
- [21] S.A. Teukolsky W.H. Press, B.P. Flannery and W.T. Vetterling. Numerical recipes in c: The art of scientific computing. *The Cambridge University Press*, 1988.
- [22] Jana Kosecka Yi Ma, Stefano Soatto and Shankar Sastry. *An invitation to 3D vision*. Springer, 1st edition, 2004.
- [23] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

