Practical Euclidean Reconstruction of Buildings

CHOU Yun-Sum, Bailey

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Electronic Engineering

March 2001

Acknowledgements

At this point I would like to express my gratitude towards my advisor, Prof. H. T. Tsui, who gave me the opportunity to work in his research group. He provided an exciting working environment with many opportunities to develop new ideas, work on promising applications and meet interesting people.

I would also like to thank Dr.Du Huynh, Dr. Z.Y.Hu and Dr. W.Wang for the fruitful discussions we had. I am also grateful to the other members who give the fully material support to my research.

Of course, I am also indebted to many colleagues. I would like to especially acknowledge Lau Ah Wai and Lu Le for the work we did together and also deserve a special mention for contributing to some results presented in this work. Besides this I would also like to thank all my colleagues who turned these years at CUHK into a pleasant time.

The financial support of the RGC is also gratefully acknowledged.

At last, I would like to thank my parents, my family and my friends for their patience and support. This was very important to me.

Abstract

The problem of 3D Euclidean reconstruction of a building in a city environment is tackled in this thesis. As images have to be captured by a handheld camera with variable parameters, self-calibration is needed to estimate the parameters. The following two self-calibration methods are used to get the basic model of a building and their results are compared. The basic model is then refined using a model based stereo technique [7].

The first method is based on the algorithms proposed by Faugeras et al [5] for camera self-calibration by restricting the camera to planar motions. A tripod mounted camera with variable focal length was used to capture the images of a building at different distances and angles dictated by environment for reconstruction. A new planar motion detection algorithm was proposed. The second method is based on the linear algorithm proposed by Newsam et al [6]. This self-calibration method assumes that the principal point is known, the camera has square pixels and has no skew. It allows 3D shapes to be reconstructed from two images while giving the camera the freedom to vary its focal length.

The model based stereo technique of Debevec et al [7] is extended to refine the basic model obtained by one of the above two methods. Good results on capturing small fluctuation in depths of the building surfaces are obtained.

論文摘要

本論文研究了城市建築物的歐幾裏德(Euclidean)重建問題。由於建築物圖像是利用內參數可變的手持攝像機拍攝而成，爲獲得建築物的三維資訊，必須進行攝像機自標定。因此，本論文中分別研究了兩種不同的攝像機自標定方法在城市建築物重建問題中的具體應用。

第一種方法基於Faugeras等人所提出的攝像機自標定方法，但必須限制攝像機在同一平面上移動。可變焦攝像機放在三腳架上，在不同的高度，距離和視點去拍攝建築物的不同影像，以求重建出其三維幾何模型。除此之外，我們還提出了一種新的平面移動檢測方法。

第二種方法基於Newsam等人所提出的攝像機自標定方法。本方法假設攝像機主點已知，攝像機圖像各圖元點是正方形，即攝像機的圖元傾斜因數爲零。利用本方法，可以從兩個可變焦攝像機所拍攝得到的圖像中重建出所拍攝物體的三維幾何模型。

最後，我們拓展了Debevec等人所提出的基於模型的立體視覺技術，對利用上述兩種方法所獲得的建築物三維幾何模型進行精化處理，使之視覺效果更爲逼真。

Table of contents

Chapter 3 Camera Calibration

Chapter 4 Self-calibration under Planar motions

Chapter 5 Building Reconstruction using a linear camera self-calibration technique

Chapter 6 Refine the basic model with detail depth information by a Model-Based Stereo technique

Chapter 7 Conclusions

List of Symbol

α_u	focal length in the horizontal pixel
α_v	focal length in the vertical pixel
Ω_∞	absolute conic in 3D projective space
C_o	Pinhole, Optical Center
$\mathbf{e}$	epipole in the epipolar line
$\mathbf{E}$	Essential matrix between the two images
f	focal length, distance between the focal plane and image plane.
$\mathbf{F}$	fundamental matrix between the left and right images
F_p	Focal plane
I	Image plane
$\mathbf{K}$	3×3 camera intrinsic parameter matrix
$\mathbf{L}_i$	equation of the trifocal line (i = 1,2,3)
m	image point in the image plane
$\mathbf{m}$	image point (2D) in the projective plane
M	object point in the 3D space
M_c	camera coordinate system
M_w	world coordinate system
$\mathbf{R}$	Rotation matrix
s ,λ	Arbitrary non-zero scalar
$\mathbf{t}$	translation vector
$\mathbf{T}_{ijk}$	1D trifocal tensor (i,j,k = 1,2)
ω	image of the absolute conic in the plane of infinity
(u_o, v_o)	the coordinate in center of the image
$\mathbf{u}$	image point (1D)in the projective line

Chapter 1

Introduction

1.1 The Goal : Euclidean Reconstruction

The capture of an image by a camera is equivalent to a projection from the three dimensional world onto a two dimensional image. A large amount of information is lost in the process. As a result, the reverse projection from a two-dimensional image sequence back to a three-dimensional structure is a difficult and ill posed problem. The lost information has to be recovered using two or more images.

The reasons for requiring a three-dimensional reconstruction from images are many. These includes the understanding of the environment by a robot[1], the grasping of objects by robotic arms [2], object or scene visualization, the recognition or modeling of three-dimensional objects [3] etc. The advent of virtual reality and virtual worlds dramatically increases the need for 3D scene reconstruction from recorded images for modeling in a virtual world.

A popular traditional method of computing 3D Euclidean reconstruction is by stereo vision. Camera calibration for the intrinsic and extrinsic parameters have to be performed. It is done off-line using a known 3D calibration block before any vision task is commenced. When a stereo pair is calibrated, it can only work if the target is within a certain range of the location of the calibration block. It will not work if there is any change of the camera parameters or the geometry of the stereo pair. It is very inflexible and inconvenient. Because of these, stereo vision does not work for many applications. If the input is a pre-recorded image sequence, the parameters of the camera are usually unknown. Also, the camera parameters could change during normal operations. These include the change of focal length due to zooming, undergoing significant changes in operation conditions (e.g., the temperature changes of a camera on a satellite), and misalignment due to a collision etc. In this thesis we focus on the 3D reconstruction of buildings. This problem cannot be handled by the traditional stereo vision approach as the images have to be captured at different ranges and angles. A new approach based on camera self-calibration has to be adopted

1.2 Self-calibration and 3D reconstruction using uncalibrated cameras

In 1992, Faugeras, Luong, and Maybank [4] showed that it is possible to perform camera calibration using only corresponding features in the image sequence. This is known as self-calibration.

Since then, 3D reconstruction using un-calibrated cameras has become a hot topic. Many self-calibration schemes have been proposed by various authors. Faugeras et al and Loung et al [4] proposed using the absolute conic for self calibration. Unfortunately, their methods is quite complex and the robust solution of the Kruppa equations may be a problem in many cases. Hartley [25] proposed a method of self calibration with at least three images taken from the same point in space with different orientations of the camera. The method is relatively simple in theory, but difficult to implement in reality. One has to determine the lens center of the camera by calibration and rotate the camera about this center to capture the images. Images captured by rotating a camera about its optical axis is a degenerated case which will not work. Marc Pollefeys et al [40] proposed a stratified approach to metric self calibration. Bill Triggs had also proposed a self-calibration method using absolute quadric[31]. The computation of these methods is quite complicated. A reasonably robust and easily implemented method may be a better choice in practice.

1.3 Scope of the thesis

In this research, we study the problem of 3D reconstruction of building. Two different self-calibration methods had been used. One by Faugeras et al[5] and the other by Newsam et al [6]. We compare the robustness of their methods with those of other self-calibration methods.

Further, the model based stereo technique of Debevec et al [7] is extended to refine a basic model obtained by 3D reconstruction using one of the two self calibration methods.

There are two major contributions of this thesis. First, a method for 3D reconstruction of buildings using two alternative techniques [5,6] for camera self-calibration is proposed. Second, the model-based technique of Debevec[7] is extended and applied to detect small fluctuations in depths on the surface of the reconstructed model above.

1.4 Thesis Outline

The following two chapters(2 and 3) of this thesis will introduce some of the basic concepts of 3D reconstruction and review the relevant literature. **Chapter 2** examines the general idea of reconstructing a scene from images taken by a standard camera. The geometry of the perspective camera and the traditional camera calibration method to get the camera intrinsic and extrinsic parameters will be introduced. 3D reconstruction by stereo vision will be discussed. **Chapter 3** reviews the methods of calibrating a camera. The idea of self-calibration is introduced. A number of self-calibration methods will be reviewed. Two of these methods will be adapted to our task of 3D reconstruction of buildings. **Chapter 4** introduces a self-calibration technique under restricted (planar) motions with our new planar motion detection scheme [8,5]. This method required the camera to take at least 3 images in the same plane to get the camera intrinsic parameters by the 1D trifocal tensor. The method was verified in real experiments on 3D reconstruction of a building and a flower. Texture mapping was used to improve efficiency with some loss in accuracy. An error analysis will given. Source of errors of selected methods in the literature will be discussed. **Chapter 5** introduces a linear camera self-calibration method. This method is an adaptation of the theory proposed in [6] to 3D building reconstruction in collaboration with Du from Murdoch University. **Chapter 6** proposes a method to improve the reconstruction by the model-based epipolar geometry. This is an extension of the method in [7] for our 3D reconstruction job. A conclusion is given in **Chapter** 7.

Chapter 2

An introduction to stereo vision and 3D shape reconstruction

Overview

This chapter reviews some of the basic ideas of computer vision in the area of computing a 3D Euclidean reconstruction using images of a scene taken by standard cameras. The pinhole camera model is explained, and its limitations are discussed. The fundamental of stereo matching is also introduced in this chapter.

2.1 Homogenerous Coordinates

Projective geometry [14] is used as a theoretical framework for camera calibration and the representation of structure. It is an extension of Euclidean geometry in which points, lines or planes at infinity are treated no differently from those in finite space. This results in simpler formulae, and removes the problem of exceptions resulting from infinity (i.e., two lines always intersect in projective space, even if they are parallel in Euclidean space).

In *n* dimensional projective space $\boldsymbol{P}^n$, a point may be represented by an *n+1* vector $(X_1, X_2, \ldots, X_{n+1})^T$. For 3-space $\boldsymbol{P}^3$, the homogeneous vector representing a point $\mathbf{x_p}=(X_1, X_2, X_3, X_4)^T$ is related to the corresponding point in Euclidean 3-space R^3, $\mathbf{x_e}=(X,Y,Z)^T$ by

$$X=X_1/X_4, \qquad Y= X_2/X_4, \qquad Z= X_3/X_4.$$

$\mathbf{x_p}$ is only defined up to a non-zero scaling, such that for a non-zero λ, then $\lambda\mathbf{x_p}$ defined the same points as $\mathbf{x_e}$, but conventionally it is chosen that X_4=1. Points at infinity can now be treated in the same way as finite points, except that X_4=0.

2.2 Camera Models

A pinhole camera model is assumed in this thesis. It is the most commonly used camera models in computer vision [15,16,17]. In section 2.2.1 the calibration of a pinhole camera is explained.

2.2.1 Pinhole Camera Model

Figure 2.1 show a pinhole camera model. Consider a focal plane F at a fixed distance f in front of an image plane I. The image plane is also called the retinal plane. An ideal pinhole C_o is in the focal plane F_p. Assume that an enclosure is provided so that only light emitted or reflected by a object pass through the pinhole and form an inverted image of that object on the image plane. Each point in the object, its corresponding image point and the pinhole constitute a straight line. This kind of projection from 3D space to a plane is called perspective projection.

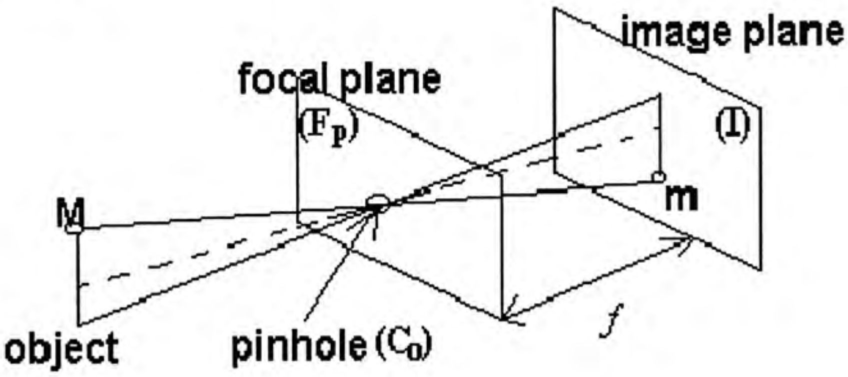


Figure 2.1 : The pinhole camera model

The geometry model of a pinhole camera thus consists of an image plane I and a point C_o on the focal plane F_P. The point C_o is called the optical center, or the focus. The plane F_P going through C_o and parallel to I is called the focal plane. The distance between the optical center and the image plane is the focal length of the optical system. The line going through the optical center C_o and perpendicular to the image plane I is called the optical axis, and its intersects I at a point C_o called the principal point. It is clear that the focal plane is also perpendicular to the optical axis. Experiences have shown that such a simple system can accurately model the geometry and optics of most of the modern Vidicon and CCD cameras.

Now let us derive the equations for the perspective projection. The coordinate system (c,x,y) for the image plane is defined such that the origin is at the point c (intersection of the optical axis with the image plane) and that the axes are determined by the camera scanning and sampling system. We choose the coordinate system (C, X, Y, Z) for the three-dimensional space as indicated in Figure 2.1, where the origin is at the optical center and the Z-axis coincides the optical axis of the camera. The X- and Y- axes are parallel, but opposite the direction, to the image x- and y-axes. The coordinates system (C, X, Y, Z) is called the standard coordinate system of the camera, or simply camera coordinate system. From the above definition of the camera and image coordinate system, it is clear that the relationship between 2D image coordinates and 3D space coordinates can be written as

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \quad \text{..........} \quad \text{..........} \quad \text{..........} \quad .(2.1)$$

If the principal point is located on the image center (x_o, y_o), then the equation (2.1) will

$$\frac{x - x_0}{X} = \frac{y - y_0}{Y} = \frac{f}{Z} \text{………………} (2.2)$$

become as equation (2.2).

where (x_o , y_o) is the coordinate in the center of the image.

It should be noted that, from the geometry viewpoint, there is no difference to replace the image plane by a virtual image plane located on the other side of the focal plane (Figure 2.2). Actually this new system is what people usually use. In the new coordinate system, an image point (x , y) has 3D coordinates (x , y ,f), if the scale of the image coordinate system is the same as that of the 3D coordinate system.

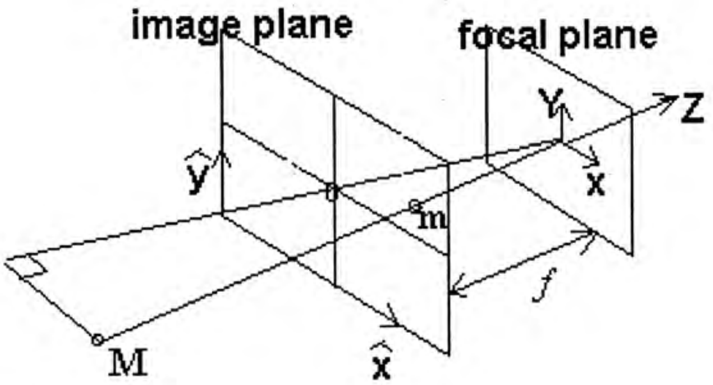


Figure 2.2 : The pinhole camera model with a virtual image plane

The ideal pinhole camera is a perspective projective from the world to the image plane, which does not model any non-linear distortion introduced by the camera. The mapping is a perspective projective from 3D projective space $\mathbf{P}^3$ to the 2D image plane $\mathbf{P}^2$ with the position of the world and image points expressed in homogeneous coordinates (see section 2.1).

The relationship between 3D coordinates and image coordinates, equation (2.1), can be rewritten linearly as

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where x=*U/S*, and y=*V/S* if *S*≠0.

Where *f* is the focal length of the camera measured in pixel units and assume the aspect ratio is unity (1). (x_o, y_o) is the principal point of the camera, which is the intersection of the optical axis and the image plane, and is measured in pixels.

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_o \\ 0 & f & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

Let **K** be the 3 × 3 matrix

Which is called the camera perspective projective matrix.

Given a 3D point **M**= $[X,Y, Z,1]^T$ and its image **m**=$[U,V, S]^T$, the equation (2.2) can be written in matrix form as

$$s\mathbf{m}=\mathbf{KM},$$

where *s*=*S* is an arbitrary nonzero scalar.

So far, we assume that 3D points are expressed in the camera coordinate system. They can also be expressed in any 3D coordinate system, such as the world coordinate system, as shown in Figure 2.3.

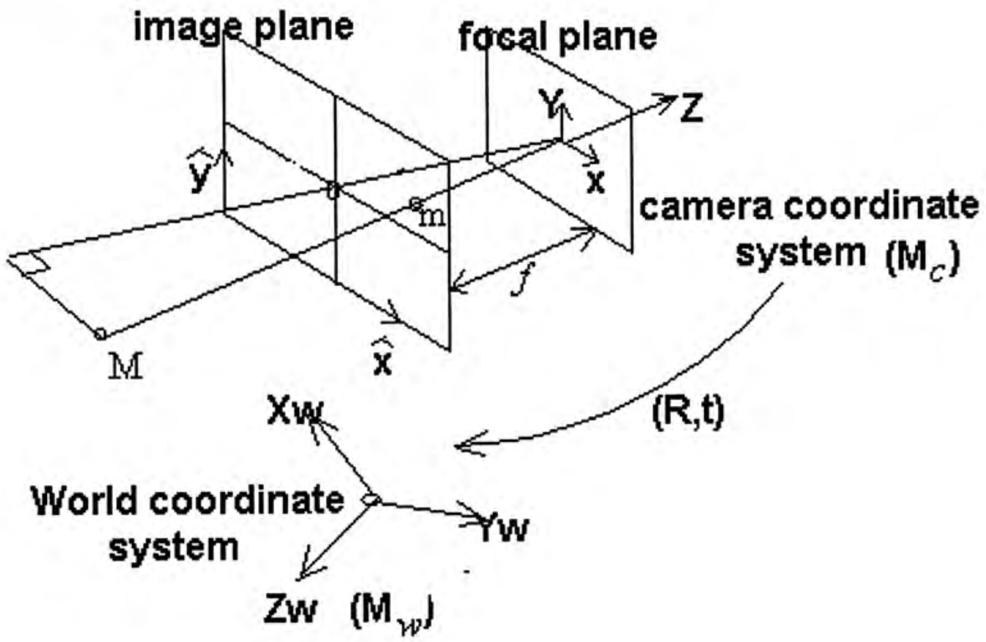


Figure 2.3 : World coordinate system and camera extrinsic parameters

We go from the old coordinate system centered at the optical center C_o (camera coordinate system) to the new coordinate system centered at point O (world coordinate system) by a rotation **R** followed by a translation **t**=C_oO. Then for a given point, its coordinates expressed in the camera coordinate system, M_c, and those expressed in the world coordinate system, M_w, are related by

$$M_c = \mathbf{R}\,M_w + \mathbf{t} \text{ ....(2.3)}$$

Or more compactly

$$M_c = \mathbf{D}M_w \text{ ....(2.4)}$$

Where **D** is a Euclidean transform of the three-dimensional space :

$$\mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

where the matrix **R** and the vector **t** describe the orientation and position of the camera with respect to the new world coordinate system. They are called the extrinsic parameters of the camera.

From equation(2.3) and equation(2.4), we have

$$M = \mathbf{K}\, M_c = \mathbf{K}\, \mathbf{D}\, M_{w},$$

The new perspective projective matrix is given by

$$\mathbf{P} = \mathbf{K}\,\mathbf{D} \text{ ....}(2.5)$$

This tell us how the perspective projection matrix **P** changes when we change coordinate system in the three-dimensional space : We simply multiply it on the right by the corresponding Euclidean transformation.

Expanding matrix **P** defined in (2.5) gives

$$\mathbf{P} = \begin{bmatrix} fR_{11} + x_0 R_{31} & fR_{12} + x_0 R_{32} & fR_{13} + x_0 R_{33} & ft_x + x_0 t_z \\ fR_{21} + y_0 R_{31} & fR_{22} + y_0 R_{32} & fR_{33} + y_0 R_{33} & ft_y + y_0 t_z \\ R_{31} & R_{31} & R_{31} & t_z \end{bmatrix}$$

Where **t**=(t_x t_y t_z) and R-ij is the ij-$^{\text{th}}$ elememt of rotation matrix **R**

The 12 elements of matrix **P** :

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

Where the **P** is determined only up to a scale. So there are in fact only 11 unknowns.

2.3 Camera Calibration

Despite all the approximation and problems with lenses, it must be emphasized that perspective projection is an extremely useful and convenient model for the geometry of image formation by a lens. We must, however, always bear in mind that that's just what perspective projection is : it is a model.

To derive three-dimensional geometric information from an image, it is necessary to determine the parameters that relate the position of a scene point to the position of its image. This determination is known as camera calibration, or, more accurate, as geometric camera calibration. Let us assume that the perspective-projection model is valid. Let us further assume a global coordinate frame for the scene, and an independent two-dimensional frame from the image. We need to relate the spatial positions and orientations of these two frames, and to determine the position of the center of projection. In addition, to account for the transformation undergone by an image between its capture on the image plane and its display, we need to determine two independent scale factors, one for each image coordinate axis.

As perspective projection and image scaling along any direction in these operations, and therefore, the complete mapping from a scene position to its image position, can be expressed as a matrix in homogeneous coordinates. Given the image positions and scene coordinates of six points, it is straightforward to derive a closed-form solution to this matrix; more points offer greater robustness. Ganapathy has shown that this matrix, in turn, provides closed-form solutions to the six extrinsic camera parameters and to the four intrinsic camera parameters. Of the six extrinsic camera parameters, three are for the position of the center of projection, and three are for the orientation of the image-plane coordinate frame. Of the four intrinsic camera parameters, two are for the position of origin of the image coordinate frame, and two are for the scale factors of the axes of this frame. Although the distance of the image plane from the center of projection cannot be modeled independently of the scale factors of the axes of the image, as indicated in our discussion of lenses, this distance is often well approximated by the focal length of the lens. On the other hand, if the scale factors of the image axes are known a priori, this distance too may be calibrated.

Typically, camera calibration is pursued using a known calibration object whose images exhibit a large number of distinct points that can be identified easily and located accurately in the image. Clearly, it is desirable that the calibration object is easy to generate and to measure accurately, and that the shape of the object be conducive to simplifying the calibration computations. One object that meets these criteria comprises either one or multiple planar rectilinear grid [10].

Camera calibration is a process to recover the 11 unknown elements of the projective matrix **P**. Depending on the type of applications, we may need to extract from matrix **P** the parameters *f*-focal length, (x_o , y_o)- principal point, the parameters from the rotation matrix-**R**, and translation vector-**t**. This traditional calibration method is need to know a number of 3D scene points (at least 6) and their image projection points as shown in the figure (2.4)

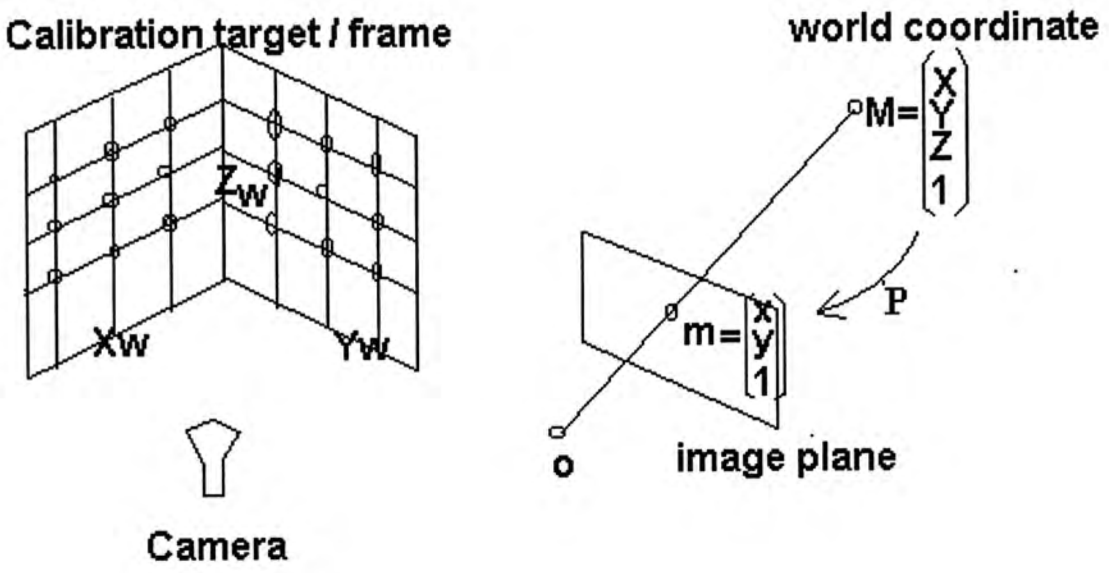


Figure 2.4 : Setup to calibrating a camera

Now, we have **PM** = *s***m**, i.e.

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Expanding the equation gives

$$\begin{aligned} P_{11}X + P_{12}Y + P_{13}Z + P_{14} &= sx \\ P_{21}X + P_{22}Y + P_{23}Z + P_{24} &= sy \\ P_{31}X + P_{32}Y + P_{33}Z + P_{34} &= s \end{aligned}$$

It become as following,

$$\begin{aligned} P_{11}X + P_{12}Y + P_{13}Z + P_{14} - (P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = 0 \\ P_{21}X + P_{22}Y + P_{23}Z + P_{24} - (P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = 0 \end{aligned} \quad \ldots\ldots\ldots\ldots(2.6)$$

This can be written as

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{bmatrix} \begin{pmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The above matrix-vector multiplication is for one scene point only. If we have n scene points then we have the following equation,

$$\begin{bmatrix} \underbrace{\begin{matrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ & & & & & & & \vdots & & & & \\ X_n & Y_{n1} & Z_n & 1 & 0 & 0 & 0 & 0 & -x_{n1}X_{n1} & -x_{n1}Y_n & -x_nZ_n & -x_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_{n1} & -y_{n1}Z_n & -y_{n1} \end{matrix}}_{2n \times 12} \end{bmatrix} \begin{pmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

or $\mathbf{B}\,\mathbf{P} = 0$

So the known vector **P** is the null-vector of the data matrix **B**. We can recover vector **P** from the singular value decomposition of **B**. However, we need to notice that the camera setting cannot be changed after the camera has been calibrated.

2.4 Geometry of a binocular system

To recover the 3D structure from 2D images, we need to use at least two images. Points **m** and **m'** are the projections of the same world point **M**; they are known as corresponding points, denoted as **m** ↔ **m'** as shown in the figure (2.5).

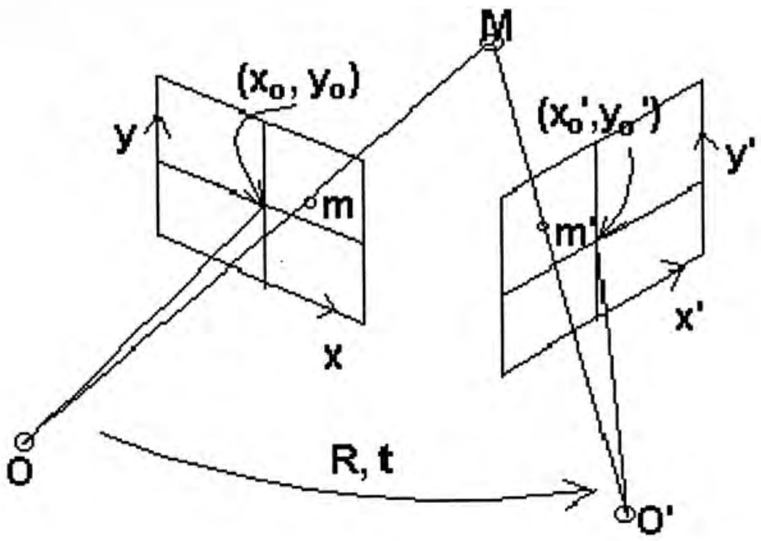


Figure (2.5) : Geometry of a binocular system

For calibrating a binocular system, we need to use the same calibration target, calibrate the left camera and then the right camera individually. Then we can get a two projective matrices **P** and **P'** and this can now be used for shape reconstruction of any object of interest. However, the camera setting for both camera cannot be changed after the binocular system has been calibrated.

2.5 Stereo matching

The establishment of correspondence requires that an entire image be searched for every point in the other image. Its meaning is given a point in one image (say, left image) , the objective of the stereo matching is to find the matching point in the other image (say, right image) as example shown in figure 2.6.

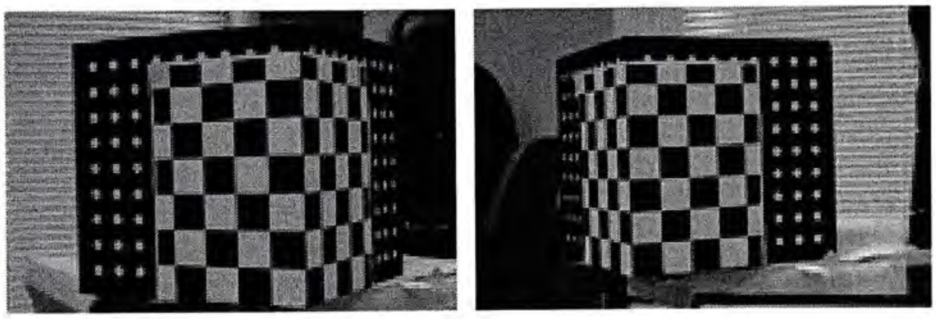


Figure 2.6 : Example pair of stereo image

Fortunately, such a two-dimensional search is not necessary because of a simple but powerful constraint: the epipolar constraint. As illustrated in figure 2.7., given an image point, its corresponding point in the other image is constrained to lie on the straight line that is the projection of the line through the given image point and its center of projection-actually, it is constrained to lie on the projection of only that portion of this line that extends outward from the given image point, rather than on the projection of the whole line. It is useful to introduce some terminology here. Consider figure 2.7. The line connecting the two centers of projection is called the baseline. A plane through the baseline is termed an epipolar plane. Any such plane will, in general, intersect the two image planes along straight line-these straight lines are called epipolar lines. Clearly, any point on an epipolar line has its corresponding image, if any, on the corresponding epipolar line-this restriction is called the epipolar constraint.

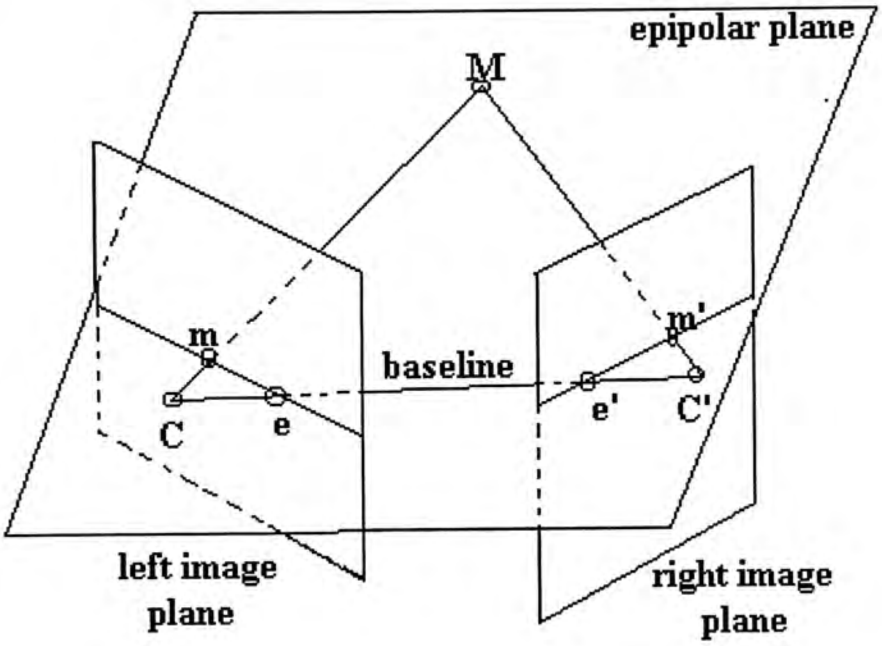


Figure 2.7 : epipolar geometry (e-epipole,C,C'-optical center, m-image point,M-Object point)

In general, the epipolar lines in each image converge toward the intersection of the image plane with the baseline; such an intersection is called an epipole. For computational convenience, the two image planes are often chosen to be coplanar and parallel to their baseline. Such an arrangement of image planes can be accomplished either physically, or, more conveniently, through analytic transformations. At any rate, when the stereo images are, in effect, coplanar and parallel to their baseline, the images are said to be rectified.

2.5.1 Accuracy of corresponding point

The accuracy of correspond point was affected by the distance between the two centers of projection. If the distance is increased, the angle between the corresponding projection rays through any given object point will also increase. For this reason, geometric stereo in which the distance between the two centers of projection (known as the baseline) is large is called wide-angle stereo, and geometric stereo in which the distance between the two centers of projection is small is called narrow-angle stereo. From Figure 2.8a, any errors in the image can produce a large error in the reconstruction when the baseline is small. Wide-angle stereo provides more precise estimates for the three dimensional position of scene points than does narrow-angle stereo. However, wide-angle stereo is disadvantageous with respect to narrow-angle stereo in at least two respects :

(1) It is more difficult to establish correspondence between image points when wide-angle stereo is used. (figure 2.8b)

(2) In wide-angle stereo, there is likely to be less overlap between the two fields of view.

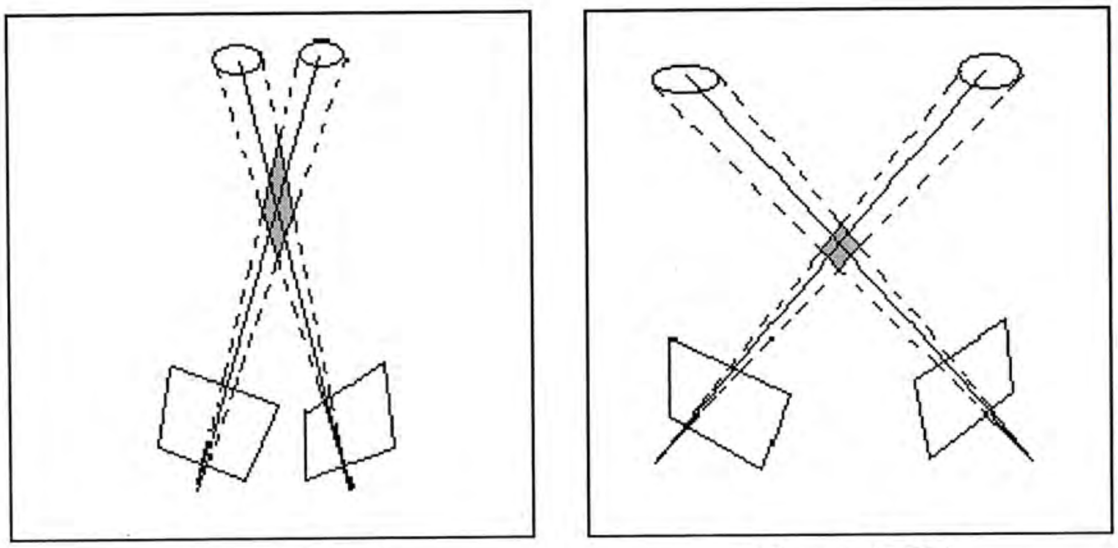


Figure (2.8a) **Figure (2.8b)**

Figure 2.8 : The effect of baseline on scene reconstruction

2.5.2 The stereo matching approach

We established in last section that correspondence search can be restricted to conjugate epipolar lines. Let us now consider how search along such lines may proceed. One set of techniques is based on matching image intensities, and the other is based on matching the features in the image.

2.5.2.1 Intensity-based stereo matching

A straightforward approach to establishing correspondence along conjugate epipolar lines is to match points on the basis of their image intensities. As several points along each epipolar line may have closely matching intensities, establishing correspondence by matching intensities on a point-by-point basis is clearly not feasible. We must instead minimize some measure of similarity between the intensity patterns exhibited by image regions. These regions could be small image windows, whole epipolar lines, or even complete images. Two conceivable measures of similarity are the sum of squared differences and the cross-correlation. If we represent the two images by $I_L(m,n)$ and $I_R(m,n)$, then the sum of squared differences between the two images over a region R can be defined as

$$SSD(\Delta m, \Delta n) \cong \sum_{i,j}\sum_{\in R} [I_L(i,j) - I_R(i-\Delta m, j-\Delta n)]^2,$$

where (Δm, Δn) is disparity between the image locations of the two matched regions. The cross-correlation can be defined as

$$CC(\Delta m, \Delta n) \cong \sum_{i,j}\sum_{\in R} [I_L(i,j) - I_R(i-\Delta m, j-\Delta n)]$$

Variations of these definitions include computing weighted sums rather than just plain sums, and, in the case of the cross-correlation, normalizing the sum by the product of the root-mean-square intensities of each of the two matched regions. The advantage of the intensity based stereo matching method is that it attempts to match all pixels in the image and output result is more dense than other method.

2.5.2.2 Feature-based stereo matching

Another stereo matching approach is detect prominent image features, such as corners, edges in the images. To establishing correspondence between image points by matching image-intensity patterns along conjugate epipolar lines is first to detect edges or corners, and then to seek matches between these edges' or corners' intersections with conjugate epipolar lines. This approach is, of course, not useful in image regions without features. It is also ineffective in the interior of features that lie along epipolar lines, Hence, feature based methods for correspondence establishment are often used in conjunction with intensity-based method.

Due to occlusion, the corresponding feature may not exist in the right (or left) image. The disadvantage of the feature based stereo matching method is the output matching result is sparse.

2.5.3 Matching Constraints

(1) Uniqueness constraint [18,19]

- As show in figure (2.9), for each point *m* in the left image, there is at most one point *m'* in the right image that matches *m*.
- For each point *m'* in the right image, there at most one point *m* that matches *m'* in the left image.

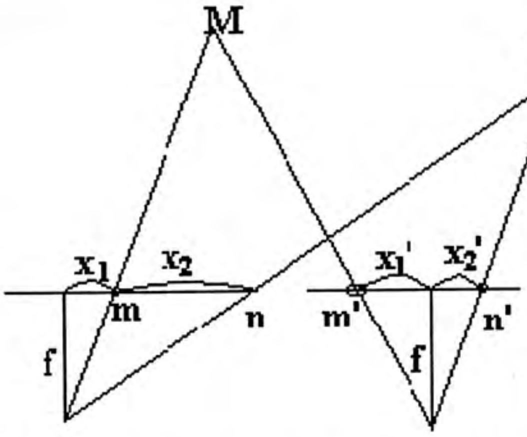


Figure 2.9 : Uniqueness constraint and Continuity constraint

(2) Continuity constraint [18,19]

The disparity changes smoothly across the image, as shown in figure (2.9), the disparity (d) = x-x'.

(3) Compatibility constraint [20]

The gray level of the dots in the corresponding image must match, where back dot match with black dots and whites dots match with the white dots.

(4) Ordering constraint

The ordering of matching features s preserved. That is if $x_p < x_q$ (ie, x_p is on the left of x_q) then $x_p' < x_p'$ (ie. x_p' should also be on the left of x_q')

(5) Epipolar constraint

Given a point *m* in the left image, the corresponding point *m'* in the right image must lie on the epipolar line of *m*. If given a point *m'* in the right image, the corresponding point p in the left image must lie on the epipolar line of *m'*.

2.6 3D reconstruction

Once the corresponding point has been established, we want to estimate the 3D coordinates of P relative to the global coordinate frame on the calibration target. Recall equation (2.6), that the left camera is

$$
\begin{aligned}
P_{11}X + P_{12}Y + P_{13}Z + P_{14} - (P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = 0 \\
P_{21}X + P_{22}Y + P_{23}Z + P_{24} - (P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = 0
\end{aligned}
$$

which can be rewritten as

$$
\begin{bmatrix} P_{11} - P_{31}x & P_{12} - P_{32}x & P_{13} - P_{33}x \\ P_{21} - P_{31}y & P_{22} - P_{32}y & P_{23} - P_{33}y \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} P_{34}x - P_{14} \\ P_{34}y - P_{24} \end{pmatrix} \cdots\cdots\cdots\cdots (2.7)
$$

Similarly, for the right camera we have

$$
\begin{bmatrix} P'_{11} - P'_{31}\,x & P'_{12} - P'_{32}\,x & P'_{13} - P'_{33}\,x \\ P'_{21} - P'_{31}\,y & P'_{22} - P'_{32}\,y & P'_{23} - P'_{33}\,y \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} P'_{34}\,x - P'_{14} \\ P'_{34}\,y - P'_{24} \end{pmatrix} \cdots\cdots\cdots\cdots (2.8)
$$

Combine (2.7) and (2.8), we have

$$
\begin{bmatrix} P_{11} - P_{31}x & P_{12} - P_{32}x & P_{13} - P_{33}x \\ P_{21} - P_{31}y & P_{22} - P_{32}y & P_{23} - P_{33}y \\ P'_{11} - P'_{31}\,x & P'_{12} - P'_{32}\,x & P'_{13} - P'_{33}\,x \\ P'_{21} - P'_{31}\,y & P'_{22} - P'_{32}\,y & P'_{23} - P'_{33}\,y \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} P_{34}x - P_{14} \\ P_{34}y - P_{24} \\ P'_{34}\,x - P'_{14} \\ P'_{34}\,y - P'_{24} \end{pmatrix} \cdots\cdots\cdots\cdots (2.9)
$$

From equation (2.9), we can solve for X, Y, Z using least-squares method. In the presence of image noise, need nonlinear optimization to avoid biased estimation. The example of 3D box reconstruction as shown in Figure 2.10.

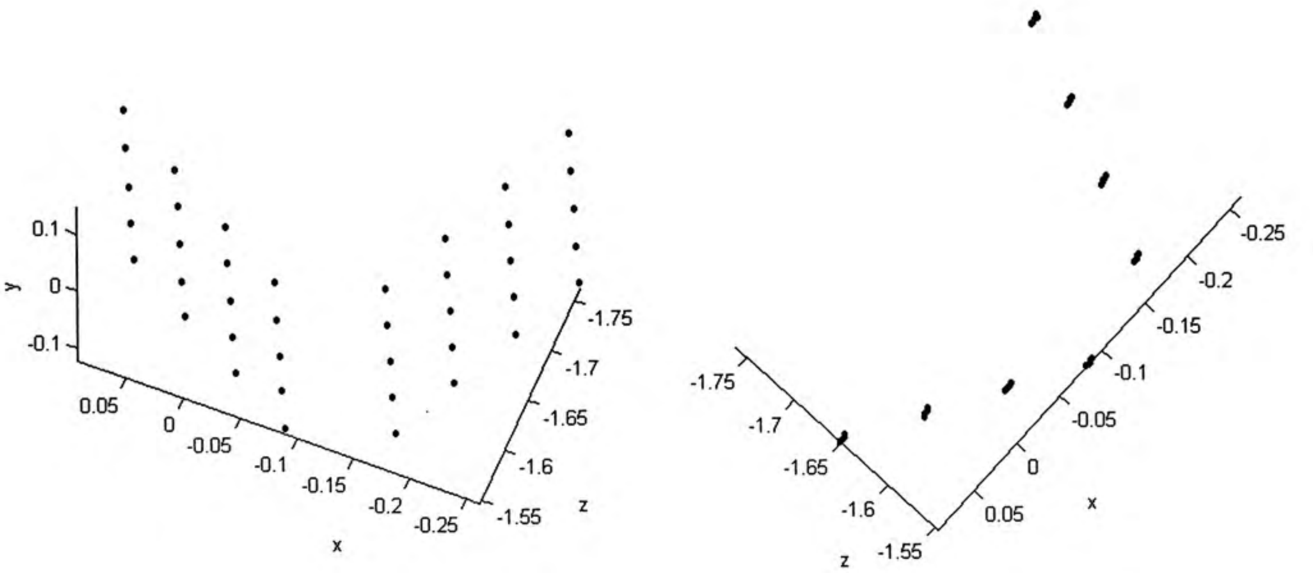


Figure 2.10 : Example of 3D reconstruction (up-to scale)

2.7 Recent development on Self calibration

The disadvantage of the standard calibration method [10] is always to use a calibration objects with regular pattern to get the camera parameters. This tedious calibration steps must redo if the camera focal length is changed. For this reason, many different researchers concentrate on the calibration without using standard calibration block [3,4,5,6,11,16,21,26,29]. Where, the Euclidean reconstruction of the scene can be computed without knowledge of camera parameters, nor the scene coordinate. Only information about point and line matches and angular relations have been used. As the path to use priori information about the scene in figure 2.11. Once the mapping which bring points from projective to affine and Euclidean space have been computed, the projective matrices are updated so that image point correspondences are directly reconstructed in the Euclidean space.

Moreover, Euclidean reconstruction can obtain without using prior information about the scene. As the path to use the priori information about the camera in figure 2.11.

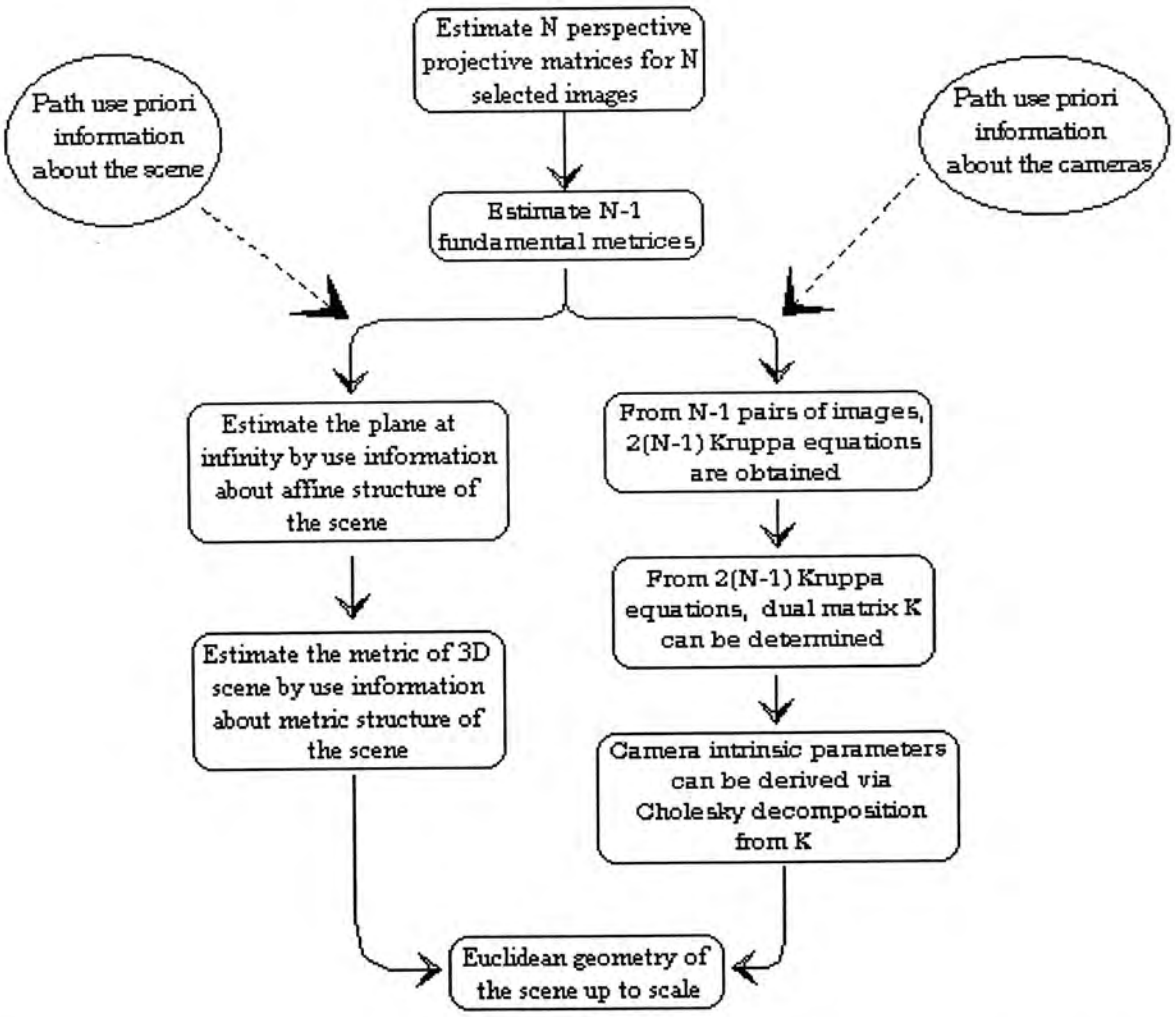

Estimate N perspective projective matrices for N selected images
Path use priori information about the scene
Path use priori information about the cameras
Estimate N-1 fundamental metrices
Estimate the plane at infinity by use information about affine structure of the scene
From N-1 pairs of images, 2(N-1) Kruppa equations are obtained
From 2(N-1) Kruppa equations, dual matrix K can be determined
Estimate the metric of 3D scene by use information about metric structure of the scene
Camera intrinsic parameters can be derived via Cholesky decomposition from K
Euclidean geometry of the scene up to scale

Figure 2.11: Recovering the Euclidean structure of the scene from images

2.8 Summary of the Chapter

The chapter introduced and reviewed several areas of computer vision concerned with the 3D Euclidean reconstruction of a scene using images taken by a camera. It concluded that to obtained a Euclidean reconstruction of a scene using a image sequence, the camera calibration needs to be known. In this chapter, the traditional camera calibration method to get the camera parameters is introduced. Also, stereo matching is important step to get the corresponding point between the images. The brief introduction of recently development without using standard calibration and disadvantage of the standard calibration has described in the end of this chapter. For simplify the camera calibration step, the camera self-calibration methods can be obtained in several ways, and is the topic of the next chapter.

Chapter 3

Camera Self-Calibration

3.1 Introduction

Traditionally, the camera calibration was obtained off-line using images of a special calibration object. High accuracy is obtainable. However, the method cannot cope with the cases when the parameters of the camera changes during the normal operation (i.e., zooming in or out), or when trying to reconstruct a scene from a pre-recorded image sequence where the camera calibration are not known a prior. A major disadvantage is that a calibrated system can only work when the target object is within a limited range. For large objects like buildings, the traditional stereo vision would not work.

Faugeras et al. [4,21] introduced the idea of self-calibration, where the camera calibration can be obtained from the image sequence itself, without requiring knowledge of the scene. This has allowed the possibility of reconstructing a scene from pre-recorded image sequences, or performing the camera calibration during the normal vision tasks. A lot of work has been done in this area, and section 3.3 review the self-calibration methods under general camera motion. Section 3.4 review the methods under specially designed motion sequences.

The main advantage of camera self-calibration is it can provide an approach for 3D reconstruction that is flexible and convenient. Also the price is cheaper when compare with the traditional camera calibration. However it is a difficult problem. During image capture for 3D reconstruction, the focal length of the camera has to vary to keep the object in focus. Even if the camera's intrinsic parameters are kept unchanged, due to the high non-linearity of the problem, the self-calibration is not an easy job. The major drawback the existing techniques of self-calibration are poor in robustness.

3.2 Camera Self-calibration

Faugeras, Luong, and Maybank [4,21] introduced the idea that a camera could be calibrated using only point matches between images, and termed the method self-calibration. This avoided the use of a calibration object (or known scene), or any knowledge of the camera motion. Since then several algorithms have been suggested, which differ in the permissible camera motions, and in the actual methods for finding the camera calibration. Some methods self-calibrate directly in one step, while others use a stratified approach are reviewed below and each method is explained in more details in the following sub-sections.

3.3 Self-calibration under general camera motion

The meaning of general camera motion is the camera involved rotation and translation between the corresponding images. Some preliminaries for the self-calibration is all self-calibration techniques assume that the image correspondence was established beforehand, e.g., by point tracking. And the fundamental matrix F was assuming to be known, e.g., by 8-point algorithm. Some specific methods which take advantage of general motion are discussed in the following paragraphs.

3.3.1 The absolute Conic Based Techniques

The definition of the absolute conic in 3D projective space is :

$$x^2+y^2+z^2=0,\ t=0, \qquad \text{or} \qquad \mathbf{a}=[x\ y\ z]^T,\ \mathbf{a}^T\mathbf{a}=0$$

The absolute conic (Ω_∞) is invariant to rigid motion. However, more interestingly for self-calibration, the image of the absolute conic (ω) is also invariant for rigid motions, and is determined by and determines the camera calibration. Also, it is possible to find ω and thence the camera calibration.

Lemma 4.1 The image of the absolute conic (ω) is invariant to rigid motions of the camera, determines, and is determined by the internal parameters of the camera.

The image of the absolute conic also is a conic. This conic keeps unchanged when the camera undergoes a rigid motion. It depends only on camera's intrinsic parameters, more explicitly,

$$\mathbf{u}^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{u}=0$$

Where, **K** is the camera's intrinsic parameter matrix, $\mathbf{u}$ = $(u, v, k)^T$ is the homogenous coordinates of an image point. This is because :

$$\mathbf{u_o}=\mathbf{K}\ [I\ |\ 0]\ [\mathbf{a}\ \ \mathbf{t}]^T,$$

for the reference frame, and

$$\mathbf{u_i}=\mathbf{K}\ [\mathbf{R}\ |\ 0]\ [\mathbf{a}\ \ \mathbf{t}]^T,$$

for the arbitrary frame, where **R** is a rotation matrix and **t** is a translation vector.

For a point lying on the absolute conic, t = 0, then

$$\mathbf{u_i}=\mathbf{K\ R\ a}$$
$$\mathbf{a}=\mathbf{R^TK^{-1}u_i},$$

For $\mathbf{a}^T\mathbf{a} = 0$ (by definition), we have :

$$\mathbf{u_i}^T\mathbf{K}^{-T}\mathbf{RR}^T\mathbf{K}^{-1}\mathbf{u_i} = \mathbf{u_i}^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{u_i} = 0 = \mathbf{u_i}^T\mathbf{C}^{-1}\mathbf{u_i}$$

Using the matrix equation for (Ω_∞) gives the definition of a conic in the image planes, which is independent of the rigid displacement (**R**, **t**) and only dependent on the camera calibration (**K**). This conic ($\mathbf{C}^{-1}$) is (ω), the image of the absolute conic, and the dual of the conic $\mathbf{C} = \mathbf{KK}^T$. Hence, ($\omega$) determines and is determined by the camera calibration.

Once the image of the absolute conic $\mathbf{C}^{-1}$ has been found, It is trivial to determine the camera calibration (**K**) by Choleski decomposition [22] of **C**. If there is significant noise on the image, it is possible that **C** will not be positive definite, which means that Choleski decomposition will give complex values for the calibration.

Kruppa's equations

The camera matrix **K** can be determine via the Kruppa equations. The original method by Faugeras et al [4] involved the computation of the fundamental matrix **F**, which encodes epipolar geometry between two images [15,16,23]. Each fundamental matrix generates two quadratic constraints involving only the five elements of **C** (and not the 3D structure or camera motion). From three views a system of polynomial equations is constructed called

Kruppa's equations. Originally [4], homotopy continuation was used to solve the set of polynomial equations, but the method is computationally, expensive and requires extreme accuracy of computation. Addition views increase the complexity. Since then, Q-T Luong has used an iterative search technique to solve the set of polynomial equations, but results were limited by the choice of initial values and the complexity of the equations.

Kruppa's equations are used are based on the relationship between the image of the absolute conic (ω) and the epipolar transformation. If an epipolar line (l) is tangent to ω, then the corresponding epipolar line (l') is also tangent to ω (see [24] for proof].

Lemma 4.2 From a pair of images it is possible to obtain a set of polynomial equations, quadratic in elements of *C*, called Kruppa's equations.

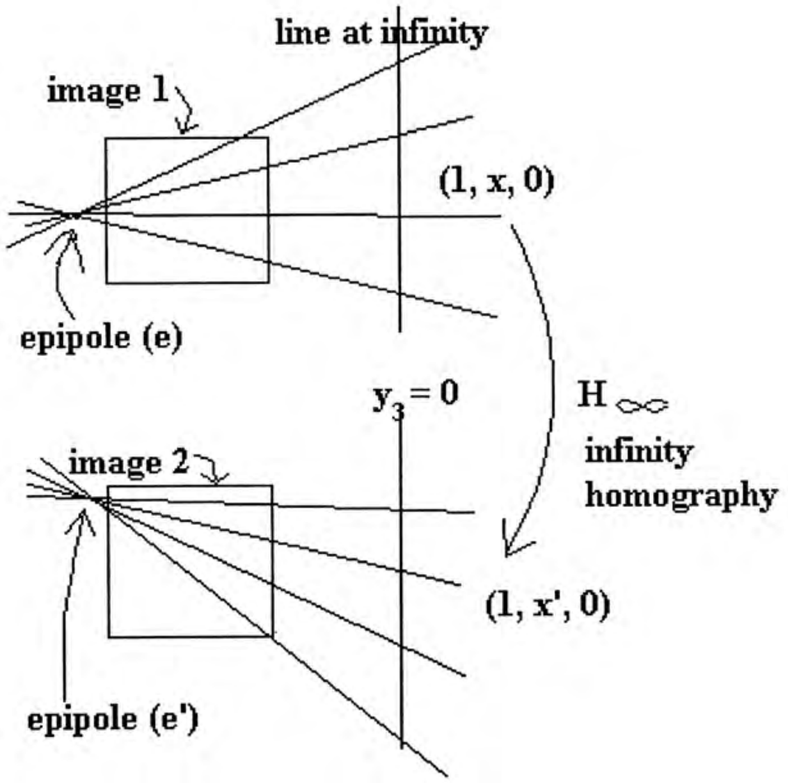


Figure 4.1 : Parameterization of the epipolar transformation.

Proof : If the epipole is not at the infinity line, any epipolar line can be determined by a point (y) lying on the line at infinity and the epipole $\mathbf{e} = (e_1,e_2,1)^T$. $\boldsymbol{y}$ can be expressed as $\boldsymbol{y} = (1, x, 0)^T$. This is because

$$y=\{(e_1,e_2,1) \wedge (q_1,q_2,1)\} \wedge (0\ 0\ 1) = (1, x, 0)^T$$

($\wedge$ denotes cross product. $\mathbf{q}=(q_1,q_2,1)$ is an arbitrary point not lying on the line at infinity.) That is to say, $\boldsymbol{y}$ is the intersection point between line e $\wedge$q and line at infinity (0,0,1), see figure (4.1).

A 3D conic two tangent planes containing the baseline are projected on corresponding epipolar lines tangent to the image conic. If an epipolar line (e $\wedge$ y) in the left image is tangent to the image of the absolute conic, then (e $\wedge$ y) must go through the dual of the image of the absolute conic, then we have :

$$(e \wedge y)^T \mathbf{C}\, (e \wedge y) = 0 \text{ ....(4.1)}$$

The corresponding epipolar line in the right image of the epipolar line (e $\wedge$ y) in the left image will be $\mathbf{F}\,\boldsymbol{y}$ ($\mathbf{F}$ is the fundamental matrix between left and right images), then in the right image, we have :

$$(\mathbf{F}\, y)^T \boldsymbol{K}\, (\mathbf{F}\, y) = 0 \text{ ....(4.2)}$$

Where (4.1) and (4.2) are quadratic polynomials in x. More specifically, from (4.1), we have :

$$a_1x^2 + a_2x + a_3 = 0 \text{............................(4.3)}$$

from (4.2), we have :

$$a'_1x^2 + a'_2x + a'_3 = 0 \text{.........................(4.4)}$$

a_i a'_i (i=1,2,3), depend on $\mathbf{F}$, $\mathbf{e}$ and $\mathbf{C}$. From (4.3) and (4.4), we obtain the following 3 Kruppa equations from a pair of view :

$$a_1/a'_1 = a_2/a'_2 = a_3/a'_3 \text{ ....(4.5)}$$

Where **C** has 6 unknowns, From (4.5), Each pair of views gives two quadratic equations containing the elements of **C**, and given three camera displacements (four independent pairs of views), they form an over-determined set of simultaneous polynomial equations. When there are only two displacements there are only four equations and five unknowns, and another constraint is required to solve for **C**.

Summary of Algorithm

- Control camera to undergo a general motion, and take 3 images;
- Determine the 3 fundamental matrices of these 3 pairs of images;
- For each pair of image, 2 independent Kruppa equations are obtained. From 3 pairs of images, 6 Kruppa equations are obtained;
- From these 6 equations, the dual matrix **C** can be determined;
- Camera intrinsic parameter matrix **K** can be derived via Cholesky decomposition from **C**.

Unfortunately, their methods are quite complex and the robust solution of the Kruppa equations may be a problem in some cases. In addition, the Cholesky decomposition can solve positive definite matrices only. Otherwise the method needs to repeat all the necessary steps again to get a positive definite matrices. Other hand, this algorithms can provide a fundamental of the self-calibration. Because the above disadvantage, so we choose another self-calibration that is more easy to implement and to get a more robust result that is described in Chapter 4 and Chapter 5.

3.3.2 A stratified approach for self-calibration by Pollefeys

Its basic principle is using modulus constraint to determine the homographies of the plane at infinity $\mathbf{H}_\infty\mathbf{S}$, then determine the camera's intrinsic parameters via $\mathbf{H}_\infty\mathbf{S}$.

Its advantage is hierarchical approach. At first determine $\mathbf{H}_\infty\mathbf{S}$, then $\mathbf{K}$, At each stage, the number of the parameters to be determined is less than that of traditional bulk one.

Modulus constraint

If the projection matrix is $\mathbf{P}=(\mathbf{P}_{3x3}, \mathbf{P}_o)$, since $\mathbf{P}_{3x3} = \mathbf{C}(\mathbf{KRK}^{-1})$, then the modulus of the 3 eigenvalues of $\mathbf{P}_{3x3}$ are equal, i.e., $|\lambda_1| = |\lambda_2| = |\lambda_3|$.

Summary of the Algorithm

1. Determination of a consistent set of projection matrices;
2. Using the modulus constraint, determine the homographies $\mathbf{H}_\infty\mathbf{S}$. of the plane at infinity;
3. Similarly as Hartley's work, using the homographies of the plane at infinity to determine ($\mathbf{KK}^T$) (replacing Hij by $\mathbf{H}_\infty$).
4. Using Cholesky decomposition method to find out $\mathbf{K}$.

The advantage of the method can let the camera to move freely, however, the calculation is too complex to solve many problem, then it cause the accuracy is limited. Additional, the points on the accuracy of homographies of the plane at infinity is difficult to find and Cholesky decomposition can solve the positive definite matrices. So we didn't choose to use this method to implement our practical experiment but use another method that is described in section 3.3.4 and section 3.4.3.

3.3.3 Pollefeys self-calibration with Absolute Quadric

In Euclidean space two entities are invariant –setwise, not pointwise– under rigid transformations. The first one is the plane at infinity Π_∞ which allows to obtain affine measurements. The second entity is the absolute conic Ω which is embedded in the plane at infinity. If besides the plane at infinity Π_∞ the absolute conic Ω has also been localized, metric measurements are possible.

When looking at a static scene from different viewpoints the relative position of the camera towards Π_∞ and Ω is invariant. If the motion is general enough, only one conic in one specific plane will satisfy this condition. The absolute conic can therefore be used as a virtual calibration pattern which is always present in the scene.

They [29] introduce a practical way to encode both the absolute conic and the plane at infinity is through the use of the absolute dual quadric Ω^* [30] with the reference in computer vision [31].

Although the camera can be moved flexibly, however, the calculation is too complex to solve many non-linearity problem, then it cause the speed is slow and the accuracy is limited. Additional, the points on the image of the absolute conic is difficult to find. So we didn't choose to use this method to implement our practical experiment in this thesis.

3.3.4 Newsam's et al self-calibration with linear algorithm

Newsam et al [6] introduced the idea of self-calibration method assumes that the principal point is known, the camera has square pixels and has no skew. It allows 3D shape to be reconstructed from two images while allowing the camera to vary its focal length.

The method is assuming that the principal point is known (so the origin of the image coordinate system can be set at (u_0, v_0)) and the camera contains square pixels (so $f_u = f_v \equiv f$), the camera matrices **K** and **K'** can be simplified to **K** = diag(*f*; *f*; 1) and **K'**= diag(f' ; *f*; 1) where *f* and f' are the unknown focal lengths for the two images in consideration. This diagonal form of **K** and **K'** allows the extrinsic parameters to be eliminated nicely from the 3x3 matrix $\mathbf{FF}^T$ and leads to a linear self-calibration method for recovering two focal lengths.

The advantage of the method is to get the focal length of the camera by the linear algorithm robustly. Although the principal point is difficult to fix when the focal length is change. However, we discover the accuracy of the principal point is less affect the accuracy of 3D reconstruction by experiment. Additional, it is not complex in calculation. So we choose to use this method with application that is described in chapter 5 in detail.

3.4 Camera Self-calibration under specially designed motion sequence

Restricted motions can result in simpler algorithms; but, on the other hand, it is not always possible to retrieve all the calibration parameters from these motions. Some specific methods which take advantage of specially design motion are discussed in the following paragraphs.

3.4.1 Hartley's self-calibration by pure rotations

Hartley [25] introduced the idea of self-calibration using a rotating camera. When there is no translation of the camera between views, there is an image-to-image projective mapping which can be calculated using point matches. This projective mapping gives linear constraints on **C**, the dual of ω. Given three or more images, these constraints define **C** and hence camera calibration.

If the camera is purely rotated about is optical center, the transformation between two images becomes a pure projective transformation. This method is : given a set of matched points $\mathbf{x_i}$, compute the 2D projective transformation as equation (4.6).

$$\mathbf{x_i} = \mathbf{H_{ij}}\, \mathbf{x_0} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.6)$$

Each projective transformation gives a constraint on *C* of the form

$$\mathbf{C}\,\mathbf{H_{ij}}^{-T} = \mathbf{H_{ij}}\,\mathbf{C} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.7)$$

Two or more projective transformations give sufficient constraint to solve for **C**, and hence the calibration **K**.

3.4.1.1 Summary of the Algorithm

1. Rotating the camera about its optical center;
2. Determine the projective transformation matrices, **Hij**, where i,j=12,3...
3. Since the image of the absolute conic does not change when the camera is translated and/or rotated, it depends only on the camera's intrinsic parameters, then we can determine ($\mathbf{KK}^T$) as below :

$$\mathbf{H}_{\mathbf{ij}}^{\mathbf{T}}\left(\mathbf{KK}^T\right)^{-1}\mathbf{H}_{\mathbf{ij}} = \left(\mathbf{KK}^T\right)^{-1}$$

$$\left(\mathbf{KK}^T\right)^{-1}\mathbf{H}_{\mathbf{ij}}^{\mathbf{T}} = \mathbf{H}_{\mathbf{ij}}^{-1}\left(\mathbf{KK}^T\right)^{-1}$$

4. Using Cholesky decomposition method to find out **K**.

Its disadvantage is impractical since the camera's optical center is not known in practice. The condition to get a unique solution from these two pairs of images is not provided. This method also cannot guarantee the obtained matrix **C** is a positive definite one, which is a prerequisite for Cholesky decomposition.

3.4.2 Pollefeys self-calibration with variant focal length

Pollefeys et al. have shown than even when the focal length changes it is still possible perform self-calibration. Several different algorithms have been suggested, including self-calibration of a stereo head [26], and self-calibration from a monocular image sequence [27,28].

The method uses an adaptation of the self-calibration from affine structure which can deal with a varying focal length. However, the adaptation requires that the position of the principal point is known. Hence, the calibration is found sequentially, with the principal point found first by zooming with a stationary camera, and then the varying focal length is found by zooming with a rotating camera. These deliberate motions can be achieved easily by taking an image sequence with a video camera.

Summary of the Algorithm

Assuming that the principal point is known and keeps unchanged.

1. From the first two translational images, an affine reconstruction is done.
   If the camera undergoes only a translation, the epipole moves along a line going through the principal point. Based on this fact, an affine reconstruction is possible.

2. Take a third image with a different orientation of the camera, the variant focal length can be obtained based on the modulus constraint.

3. Eliminate the effect of variant focal length, then determine the camera's intrinsic parameters.

Its main problem is it required the first two images are translational ones. In practical, it is difficult to ensure the achieve it. Additional, the modulus constraint is complex in calculation and its robust solution may be a problem in some cases. On the other hand, its computation speed is slow. So we choose another faster algorithm to implement our practical experiment as describe in Chapter 4 and Chapter 5.

3.4.3 Faugeras's et al self-calibration of a 1D Projective Camera

Faugeras et al [5] have introduced the concept of self-calibration of a 1D projective camera from point correspondences. This method for uniquely determining the two internal parameters of a 1D camera based on the trifocal tensor of three 1D images. It requires the estimation of the trifocal tensor which can be achieved linearly with no approximation unlike the trifocal tensor of 2D images, and solving for the roots of a cubic polynomial in one variable.

Summary of the Algorithm

1. Setup a camera to take horizontal planar motions. Capture a subsequence of 3 images from 3 view points in general position on a plane. This is repeated for three horizontal planes.

2. Estimate the fundamental matrices of the three images in the same subsequence.

3. Verify whether the camera motion is planar for each subsequence.

4. Transform the homogeneous 2D image points to the homogeneous 1D image points.

5. Estimate the linear 1D trifocal tensor and get the three intrinsic parameters by self-calibration.

Although its method is restricted to take a image under planar motion is its disadvantage. However, it can get the intrinsic parameter robust and it is not complex in calculation is its advantage. For the application, if the target object is far away from the camera, then the planar motion is easily to achieve correspondingly. For this advantage, we choose and extend this self-calibration method with application that description in Chapter 4 in detail.

3.5 Summary of the Chapter

This chapter has reviewed the different methods of camera calibration. Section 3.3 and 3.4 introduced the idea of self-calibration, where the camera calibration can be using only information contained in the images themselves, and explained the actual knowledge required for self-calibration. It reviewed the many methods that have been suggested for self-calibration, derived many of the basic results. A problem for many of the methods is the algebraic and numerical complexity of self-calibration, and that the methods are slow and require extreme accuracy of computation. Also, increasing the number of views used greatly increases the complexity of the computation. This is not advantageou as increasing the number of images used in the computation should improve the accuracy obtainable.

In the next two chapter, the detail to study and extend the two self-calibration method (one is proposed by Faugeras et al's. and another is proposed by Newman et al's) with experiment result analysis will be given.

Chapter 4

3D reconstruction based on self-calibration under planar motions

4.1 Introduction

The 1D trifocal tensor required can be estimated accurately using a linear method. If the camera motions are planar and horizontal, the above three parameters can be estimated robustly directly from the circular points and the trifocal line. On the other hand, if the camera moves in three different planes, 5 intrinsic parameters of the camera can be estimated. By fitting an complex ellipse to the images of six circular points, two from each planar motion, one can obtain the image of the absolute conic. The intrinsic parameter matrix can be obtained from the Cholesky decomposition of the **C** matrix representing the image of the absolute conic. In using these algorithms, we had proposed a new simple method for detecting planar motions.

Once the self- calibration is done, we can do a partial 3D reconstruction of a building from at least 3 images on a subsequence from viewing angles which have a large overlap of the visible scene. The intrinsic parameters are fixed within a subsequence, but the focal length and thus the principal points may vary between subsequences covering very different viewing angles. We then scale and then transform all the partial 3D model into one reference frame and merge them into a single model. Finally, we perform texture mapping from the images to the 3D model to complete the job.

We had verified our methods by simulation. We have done a 3D reconstruction of a building and obtained some good results. The reconstructed pairs of orthogonal walls are at an angle of about 88 degrees. The average ratio of equal length lines is about 0.98. We had also done a 3D reconstruction of a real paper flower with satisfactory results.

4.2 1D Projective Camera self-calibration

Faugeras et al proposed two self-calibration schemes using two different planar motion alternatives: camera moving in horizontal planes and camera moving in three different planes. A planar motion consists of a planar translation and a rotation about an axis perpendicular to the plane. For a full perspective 2D camera, self-calibration must be performed on at least three different planar motions to solve for the 5 intrinsic parameters. From each plane, two imaginary circular points can be obtained. All six of these distinct imaginary points from three planes must lie on the image w of the absolute conic. So we can fit an imaginary ellipse to these six points to obtain the conic matrix **C** of ω. The resulting intrinsic parameter matrix **K** can be obtained by the Cholesky decomposition of **C**. However, The fitting problem of the absolute conic is very hard to be solved because the matrix **C** has to be positive definite for Cholesky decomposition. This is very hard to achieve. Further, the converting of a 2D image points to a 1D image points for non-horizontal planar motion may bring some large errors on the estimation of the 1D trifocal tensor. This will be described in details in section 4.2.2.

If the skew of the camera can be ignored, and the sizes of the camera cells are known, then only three intrinsic parameters: coordinates of the principal point and the focal length are required to be obtained by self-calibration. In this case, horizontal camera motions are sufficient for the self-calibration. Fitting of an ellipse to complex points is not required and there is no need for Cholesky decomposition. Intrinsic parameters can be obtained directly from the imaginary points and the location of the trifocal line directly. This simpler method may lead to less computational errors. On the other hand, the horizontal motions may provide effective constraints for accurate estimation. In this paper, we shall compare the performance of self calibrations using only horizontal planar motion to that using three different planar motions.

A tripod fitted with a wooden plane was used to mount a camera. The plane can be tilted at different angles so that different planar motions can be obtained by moving the tripod in a horizontal plane and rotating the camera on the wooden plane. This set up is very convenient for taking a sequences of images under horizontal planar motion (by setting the tilt angle to zero) and other planar motions (by setting the tilt angle) as shown in figure 4.1 . A gradient meter was used to ensure the camera moving plane is parallel to the ground and the tripod kept the height between camera moving plane and the ground constant.

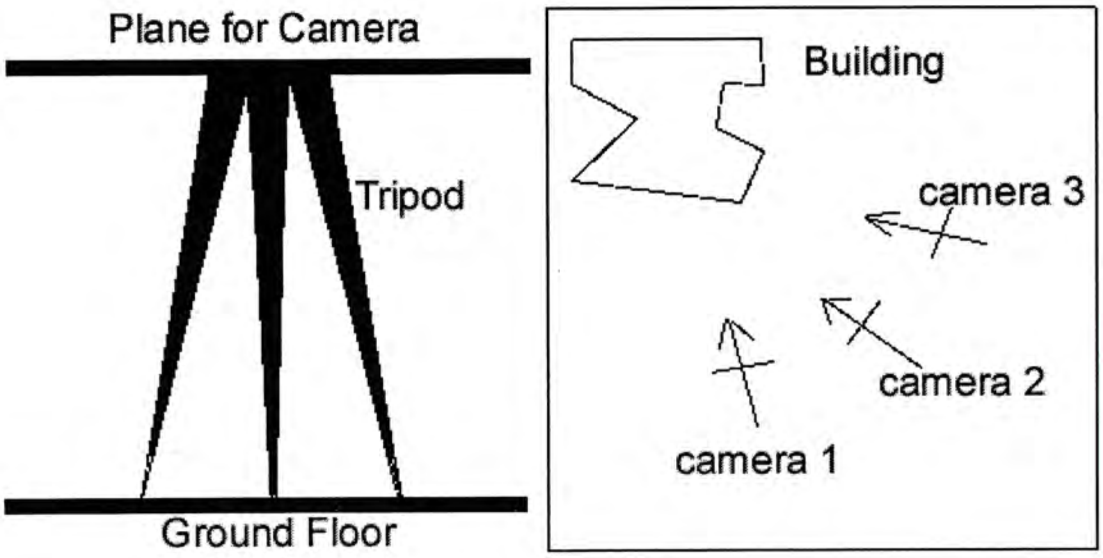


Figure 4.1 : Setup and the camera position in horizontal plane (Top view)

Here we use several image subsequences captured at different view angles for 3D recovery of a building. The focal length is constant in the same subsequence. However, it can be changed in different subsequence which are captured at different distances and angles. The height of the moving plane of the camera can be change for different subsequences too.

4.2.1 1D camera model

This 1D model is due to Faugeras [5]. Consider a 2D projective camera. It projects a 3D object point $\mathbf{M}=[X,Y, Z,1]^T$ in the projective space to a 2D image point $\mathbf{m}=[u,v,1]^T$ in the retina, the projection can be described by a 3×4 matrix $\mathbf{P}_{3\times4}$, the projection Equation (4.1) can be written in matrix form as :

$$s\mathbf{m}=\mathbf{P}_{3\times4}\mathbf{M}\ldots\ldots\ldots\ldots\ldots(4.1)$$

where s is an arbitrary nonzero scalar.

Consider a 1D projective camera. It projects a 2D point $\mathbf{m}=[u,v,1]^T$ in the projective plane to a 1D point $\mathbf{u}=[x,1]^T$ in the projective line. The projection can be described by a 2 × 3 homogeneous matrix $\mathbf{P}_{2\times3}$ the projection equation (4.2) can be written in matrix form as :

$$\lambda\mathbf{u}=\mathbf{P}_{2x3}\mathbf{m}\ldots\ldots\ldots\ldots(4.2)$$

where λ is an arbitrary nonzero scalar.

Assume a 3D coordinate system with x-axis and z-axis in the horizontal plane and the y-axis in the vertical plane. The corresponding points in 2D triplet of images are described as $\mathbf{m} \leftrightarrow \mathbf{m}' \leftrightarrow \mathbf{m}''$, then the corresponding points of 1D triplet of images are described as $\mathbf{u} \leftrightarrow \mathbf{u}' \leftrightarrow \mathbf{u}''$. The definition of the 1D trifocal tensor $\mathbf{T}_{ijk}$ is given in equation(4.3).

$$\mathbf{T}_{ijk}\,\mathbf{u}^i\,\mathbf{u}'^j\,\mathbf{u}''^k=0\ldots\ldots\ldots\ldots(4.3)$$

We can solve the tensor components linearly with at least 7 correspondence points. The complete 1D projective camera model is shown in figure (4.2a). The 2D image point (**m**) is project to 1D image point (**u**) in the projective line (trifocal line) perpendicular. All of the trifocal line, 1D image points (**u**, **u**′ and **u**″) and optical center (C, C′ and C″) is project on the same plane as shown in figure (4.2b).

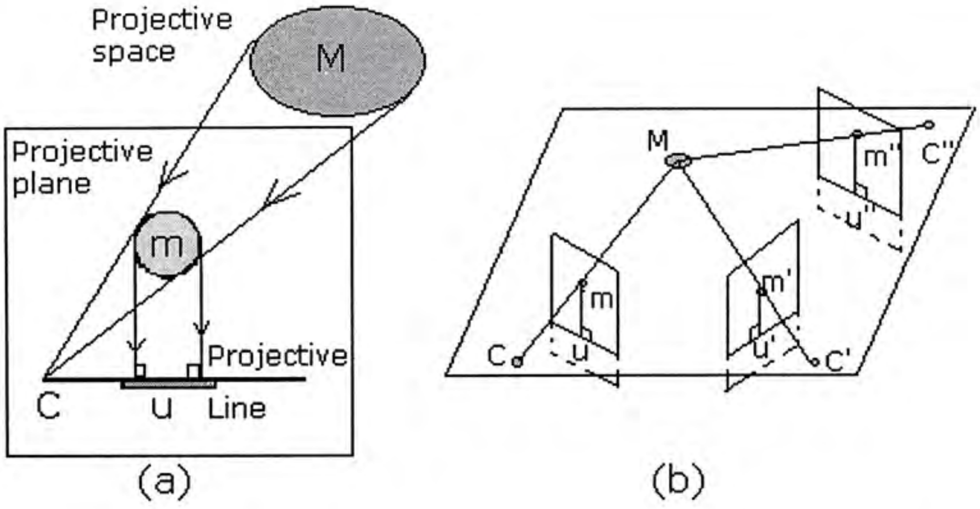


Figure 4.2 : (a) Complete 1D projective camera model (b) Creating a 1D images from a 2D images in the same plane

4.2.2 1D Projective Camera Self-calibration Algorithm

The algorithm for self-calibration using three different planar motions is due to Faugeras et al [5]. He also proposed a simplified algorithm using only horizontal planar motions. A short review of this is given in section 4.2.4. The former algorithm is based on the fact that the determination of the image ω of the absolute conic is equivalent to the determination of the 5 intrinsic parameters of a 2D projective camera. The object space of a 1D camera is a projective plane and any rigid motion will leave a pair of complex conjugate points, called circular points, at the plane of infinity invariant. The image of the circular points will also be invariant to camera motion too. Further, they are imaginary points on ω.

Since the camera intrinsic parameters are constant in the same subsequence, 1D coordinates of the circular points are invariant in these three images. Let us consider a circular point, say ***i***. This circular point is projected into **i**, **i′** and **i″** in the three views, we have

$$\lambda \mathbf{i} = \lambda' \mathbf{i}' = \lambda'' \mathbf{i}'' = \mathbf{u}$$

λ, λ' and λ'' is arbitrary scalar and $u=(u^1,u^2)^T$.

Because the triplet of corresponding points i=i′=i″ satisfies the trilinear constraints all corresponding points do, therefore,

$$T_{ijk}\, i^i\, i'^j\, i''^k = 0 \ldots\ldots\ldots\ldots (4.4)$$

i.e. equation (4.3).

This yield the following cubic equation in the unknown $x = u^1/u^2$:

$$T_{111}x^3 + (T_{211} + T_{112} + T_{121})x^2 + (T_{212} + T_{221} + T_{122})x + T_{222}=0 \ldots\ldots\ldots (4.5)$$

where (T_{ijk} , i,j,k,=1,2) is 1D trifocal tensor.

The solutions of (4.5) will give a pair of complex points (circular points) and a real point. This is repeated for 3 different planar motions. Then we can fit a complex ellipse to the 6 complex points obtained to get the image of the absolute conic using equation (4.6) below.

$$\mathbf{x}^T \mathbf{C}\, \mathbf{x} = \mathbf{0} \ \ldots\ldots\ldots\ldots (4.6)$$

where the conic matrix **C** is written as

$$\begin{bmatrix} a & 0 & d \\ 0 & b & e \\ d & e & c \end{bmatrix}$$

when the skew is equal to zero.

From Faugeras et al[5], we obtained the summary of this algorithm as following :

1. Take three images of the scene from 3 viewing points in general positions on a plane.
2. Estimate the three fundamental matrices between the 3 pairs of images.
3. Verify that the motion is planar (see section 4.2.3 below). If it is not planar, stop.
4. Project the point correspondences in the retinal plane using either one of the two methods proposed by Faugeras et al[5].
5. Estimate linearly the trifocal tensor of the 3 corresponding 1D images. See [5] for details.
6. Solve for the 3 roots of equation (4.5). Two of the roots are complex conjugate number. They are 2 points on the image of the absolute conic.
7. If the number of complex points on the absolute conic is less than the number of internal parameters, go to step 1.
8. Fit a complex ellipse to the complex points to obtain **C** of equation(4.6).

For camera under horizontal planar motion, the last two steps can be neglected.

4.2.3 Planar motion detection

Faugeras et al[5] has proposed a method to verify the planar motion, but it is a little complex and not easy to understand. Here we propose an alternative method which is more effective and simple.

Once three images of the scene have been taken with the camera setup as the description in section 4.2.1, we can estimate the fundamental matrices and epipole for each pair of images. Then:

1. Get the equation of the trifocal line by the cross product of the two epipoles on each image.
2. The equations of the trifocal lines in the same image subsequence must be equal up to scale if they are planar motions, as shown in Equation(4.7). Where $L_1 = \lambda L_2 = sL_3$ are equations of trifocal line and λ and s are arbitrary scale factors.

$$L_1 = \lambda L_2 = sL_3 \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.7)$$

The equality can be tested by whether the cross product of any two trifocal line vectors is equal to zero or not as shown in Equation (4.8).

$$L_1 \times L_2 = 0 \ , \ L_1 \times L_3 = 0 \ , \ L_2 \times L_3 = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.8)$$

The above is easier to understand and implement than Faugeras' method [5]. Once the images have been verified as being planar, one can convert the 2D image into a 1D image using 1D projective camera model [5].

4.2.4 Self-calibration under horizontal planar motions

Since the camera is moving in horizontal planes, all the 2D image points in the same subsequence can be projected onto the same trifocal line which is parallel to the u-axis of the image, given by $v = v_o$, i.e.

$$(u_i^n, v_i^n, 1)^T \rightarrow (u_i^n, 1)^T \cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.9)$$

The equation(4.9) shows the image projection from 2D image points to 1D image points on the trifocal line. This projection is shown in the CASE 1 of figure(4.3). For horizontal planar motion, this projection is very easy to be calculated.

Then more than 7 homogeneous 1D image points are used to estimate the 1D trifocal tensor (**T**$_{ijk}$, i,j,k,=1,2) linearly by equation(4.3). As the intersection of the absolute conic and the trifocal line, the images of the circular points are given by $u_o \pm i\alpha_u$ from the solutions of equation (4.5). It turns out that the focal length is α_u in horizontal pixels and u_o is horizontal location of the principal point [35 , 5]. For horizontal planar motion, the vertical location of the trifocal line determined the vertical position of the principal point v_o. The three intrinsic parameter f, u_o and v_o of the camera are estimated. Here we assumed that the skew is equal to zero and the ratio of pixel cells is known.

4.2.5 Self -calibration under three different planar motions

However, for a non-horizontal planar motion of the camera, the calculation of 1D image points needs more work and is sensitive to 2D image noise. In the following, two important remarks are made on the process of converting 2D image points into 1D image points for a non-horizontal planar motion.

Remark (1):

Given two image points (**m₁** and **m₂**) and a line (line 1) obtained by their cross product, the intersection $(v_1,v_2,1)$ between line 1 and the trifocal line shown in figure(4.3) does not directly give the coordinate of the corresponding 1D image point $(x_1,1)$ for the 1D camera case. These intersection points $(v_{1i},v_{2i},1)$ i=1,2,...n must be represented in 1D homogenous coordinates $(x_1,1)$ as described in Remark (2).

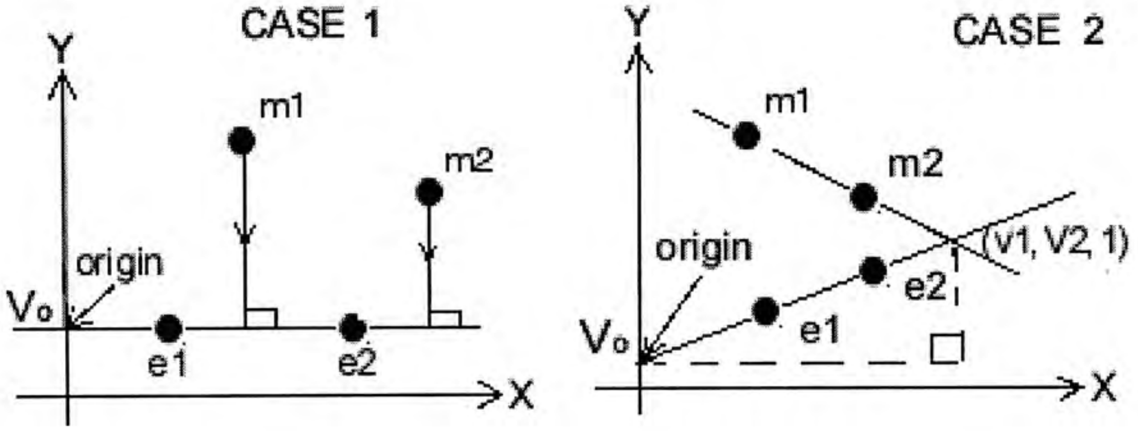


Figure (4.3) : The projection from 2D image points to 1D image points for the horizontal planar motion (CASE 1) and non-horizontal planar motion (CASE 2)

Remark(2) :

To convert the intersection points $(v_{1i},v_{2i},1)$ into 1D homogeneous coordinates $(x_1,1)$, the origin of the 1D camera image coordinate must be defined. Since the trifocal lines in the same subsequence are the same image line, here we can define any point on the trifocal line as the origin so long as all the origins have the same 2D image coordinates. In our experiments we took the intersection point $(0, v_o,1)$ between the trifocal line (i.e. $t_1u + t_2v + t_3 = 0$)and the y-axis of the 2D image as the 1D origin (0,1) in each image. Then the system can convert these intersection points $(v_{1i},v_{2i},1)$ into 1D homogeneous coordinates $(x_1,1)$ by the following relations :

Relation (1) :

$$x_i = \sqrt{v_{1i}^2 + (v_{2i} - v_0)^2}$$

if $V_{1i} > 0$

Relation (2) :

$$x_i = -\sqrt{v_{1i}^2 + (v_{2i} - v_0)^2}$$

if $V_{1I} \leq 0$

For non-horizontal planar motion, the 2D image noise can influence this projection's direction (as line 1) and cause large error in the 1D coordinates of these points. Thus the noise of the 1D image coordinate is magnified by the 2D image noise.

Then we can calculated their 1D trifocal tensors and find the circular points on the trifocal lines by equation (4.5). When the image subsequences of the three different planar motion has been deal as above, then we can obtain **C** from 6 circular points as described in section 4.2.2.

4.2.6 Result analysis on self-calibration Experiments

Assume the intrinsic parameter matrix as

$$\begin{bmatrix} 1000 & 0 & 20 \\ 0 & 1100 & 30 \\ 0 & 0 & 1 \end{bmatrix}$$

The results of self-calibration using 3 different planar motions (T : not including any horizontal planar motion, T_h : including one horizontal planar motion) or using only horizontal planar motions (H) are shown in table 4.1. It can been seen that the results of T_h are better than T. However, the results by using only horizontal planar motions are the best. The estimation of f_u is very robust to noise for the cases H and T_h.

	Noise = 0 pixel			Noise = 0.1 pixel			Noise = 0.3 pixel		
	T	T_h	H	T	T_h	H	T	T_h	H
f_u	1000	1000	1000	992.7	1000.9	1001	1038	998.5	1003.6
f_v	1000	1100		1104.9	1098.8		1095.1	1120.6	
u_o	20	20	20	24.4	20.1	17.6	27.7	22.9	22.1
V_o	30	30	30	45.7	31.8	28.55	34.7	42.9	25.976

	Noise = 0.5 pixel			Noise = 0.7 pixel			Noise = 1.0 pixel		
	T	T_h	H	T	T_h	H	T	T_h	H
f_u	974.7	1009.4	1005.6	1046	984.4	1008.5	927.3	989.9	991.4
f_v	1085.5	1108.4		1032.3	1085.2		1068.5	1161.9	
u_o	44.7	7.5	30.1	-407.9	37.2	31.2	168.2	10.98	29.2
v_o	112.5	-13.7	33.7	-185.1	57.4	29.453	148.7	91.3	38.7

	Noise = 1.2 pixel			Noise = 1.5 pixel			Noise = 1.8 pixel		
	T	T_h	H	T	T_h	H	T	T_h	H
f_u	1385	984.2	996.94	1125.5	1025.9	1016.6	755.7	1055.4	1015.6
f_v	1514.6	1187		1262.8	1061.8		1283.1	1355.3	
u_o	61.7	24.3	25.5	-306.1	33.4	11.27	116.3	60.1	18.8
v_o	-353.9	117	44.881	-7.9	-7.0	53.8	464.6	-14.9	15.964

Table 4.1 : Relationship between intrinsic parameters and noise. (unit in pixels)

(T : experiment not including any horizontal planar motion, T_h : experiment including one horizontal planar motion, H : experiment using only horizontal planar motions)

(f_u,f_v) : focal length in horizontal and vertical direction,

(u_o, v_o) : coordinate of the principal point.

4.3 Essential matrix and Triangulation

Having recovered the focal lengths, the essential matrix **E** can be estimated easily using the formula $\mathbf{E}=\mathbf{K}'^{-T}\mathbf{F}\mathbf{K}^{-1}$. Our current version of triangulation for 3D reconstruction still has room for optimization and is part of the on-going work of our project. Triangulation in the presence of image noise has been discussion by Weng et al [43] and recently by Hartley and Strum [34,16]. We will conduct further investigation on this issue.

4.4 Merge of Partial 3D Model

Once different partial models have been built by the triangulation method [34], we can merge them together to form a complete 3D model as shown in figure (4.4).

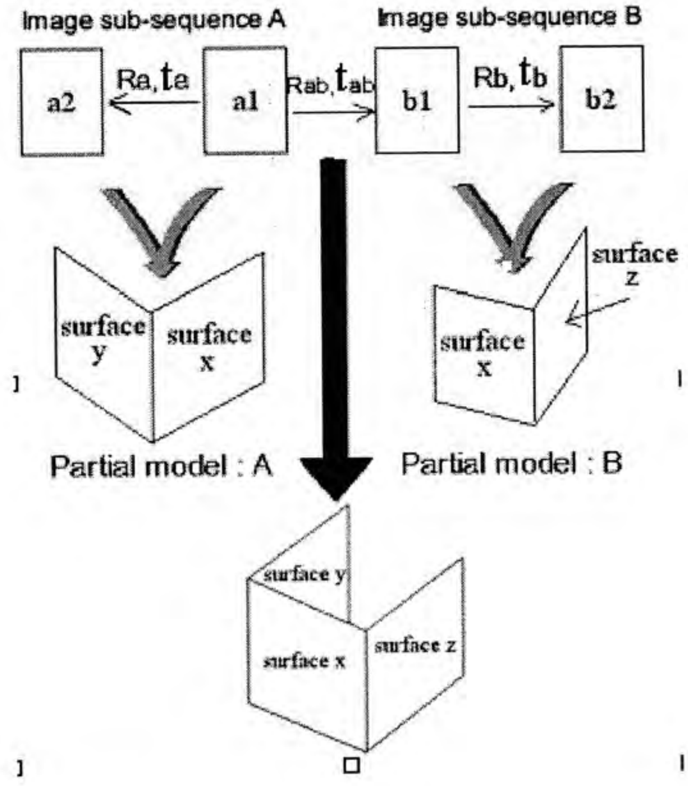


Figure 4.4 : Partial models generate from 2 difference sequence of images

Hartley's method [16] is used to get the poses of the camera relative to a reference coordinate frame. However, different subsequences have different reference coordinate systems. Figure (4.4) shows that the relative rotation and translation between two images in subsequence A are $\mathbf{R}_a$ and $\mathbf{t}_a$. Frame a_1 is taken as the reference frame to get the partial model A. In a similar way, the relative rotation and translation between two images in subsequence B are Figure

(4.4) shows that the relative rotation and translation between two images in subsequence B are $\mathbf{R_b}$ and $\mathbf{t_b}$ and frame b_1 is taken the reference frame to get the partial model B.

To merge model A and model B, we must estimate the pose between the image a_1 and image b_1 as $\mathbf{R_{ab}}$ and $\mathbf{t_{ab}}$ first using Hartley's method. Let frame a_1 be the reference frame of the whole sequence. Thus model B is transformed by $\mathbf{R_{ab}}$ and $\mathbf{t_{ab}}$ to frame a_1. Finally, two partial models are merged together as a complete 3D model as shown in figure (4.4)

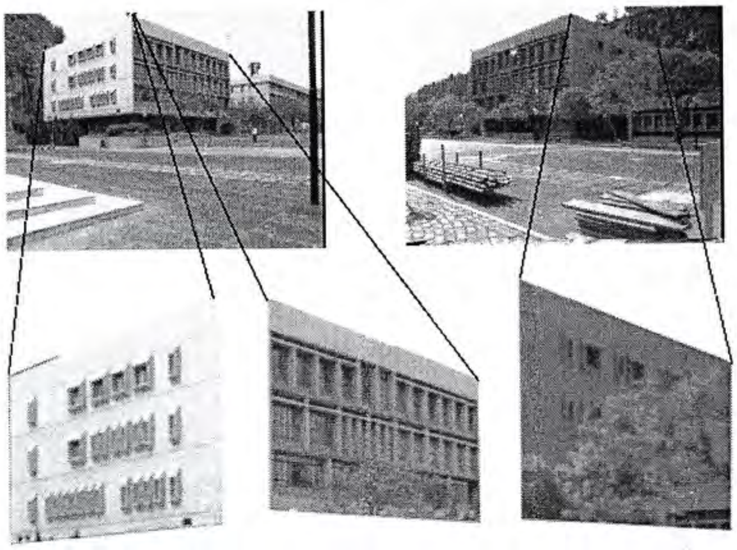


Figure 4.5 : Define a texture

Once the complete wire frame of the reconstructed model has been built, we can use the standard texture mapping method to warp an area of texture to the corresponding planar areas. At the beginning, we extract the necessary texture from the image manually as shown in figure (4.5) . The 3D location of the texture area is calculated. The defined texture is then mapped onto the corresponding 3D location.

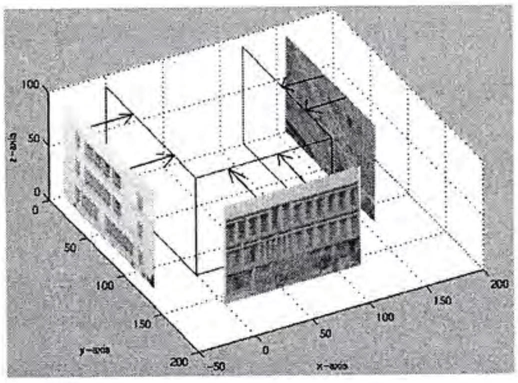


Figure 4.6 : Texture mapping

4.5 Summary of the Reconstruction Algorithms

The summary of the algorithm for 3D reconstruction system from camera setup to texture mapping is given as follows.

1. Setup a camera to take horizontal planar motions. Capture a subsequence of 3 images from 3 viewing points in general position on a plane. This is repeated for three horizontal planes.
2. Estimate the fundamental matrices of the three images in the same subsequence.
3. Verify whether the camera motion is planar for each subsequence by our proposed method.
4. Transform the homogeneous 2D image points to the homogeneous 1D image points.
5. Estimate the linear 1D trifocal tensor and the get the three intrinsic parameters by self-calibration[5].
6. Transform the image points by the inverse of intrinsic parameter matrices to get the essential matrices.
7. Estimate a rotational matrix **R** and translation vector **t** by the method that is proposed by Hartley[16].
8. Calculate the 3D coordinates from corresponding image points by the linear-eigen triangulation method[34]. If the 3D points are not in front of both cameras, go to step 7.
9. Scale the 3D model by the ratio of the corresponding features in the real object and the reconstructed model.
10. Verify the orthogonality to ensure the model is correct.
11. Merge the partial 3D model reconstructed from each subsequence into a complete 3D model.
12. Extract the texture from the images manually .
13. Warp the extracted texture to the corresponding area in the reconstructed 3D model.

4.6 Experimental results

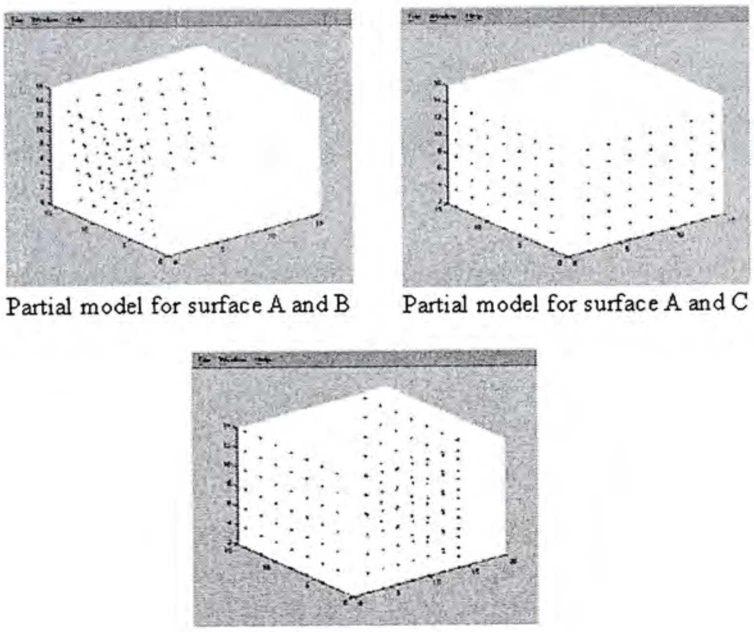
Experiments were performed using simulation data and real images to verify our method. Error analyses are also done to assess the robustness of the scheme.

4.6.1 Experiment 1 : A simulated box

We simulated a 3D box with three surfaces and each surface is represented with 49 regular points. Two subsequence of images are captured by a ideal pin hole camera under horizontal planar motion from the left side to the right side of the box respectively. The intrinsic parameter matrix (**K**) is given as :

$$\begin{bmatrix} 250 & 0 & 256 \\ 0 & 260 & 256 \\ 0 & 0 & 1 \end{bmatrix}$$

The reconstructed model is shown in figure (4.7). In this case, we did not add any noise to images. The 3D relative error and 2D RMS back projection error of this simulated box are extremely small. This verify the correctness of our scheme.



Partial model for surface A and B

Partial model for surface A and C

Complete 3D model

Figure (4.7) : A reconstructed box model (simulation)

The accuracy of the reconstructed model is highly affected by the noise level in the 2D images. Two experiments were repeated on the simulated box with uniform noise added to observe the relationship between the reconstruction error and the noise level. With the reconstruction model, we calculate the average 3D relative error between the ideal model and the reconstructed model. The result is shown in figure(4.8). When the noise level is one pixel, the 3D error is 2.43%. When the noise level is two pixels, the 3D relative error increases to 8.01%.

In the reconstructed 3D model, the angle between the orthogonal planes is 87.16°. We also calculate the RMS 2D back projection error between the ideal model and the reconstructed model. The result is shown in figure(4.9). When the noise level is one pixel, the error is 2.32 pixels in x-direction and 2.18 pixels in y-direction. When the noise level is two pixels, the error is 8.32 pixels in x-direction and 6.89 pixels in y-direction.

It can be seen from the figure 4.8 and 4.9 that the 3D reconstruction error and the 2D back projection error will increase almost linearly with the noise level.

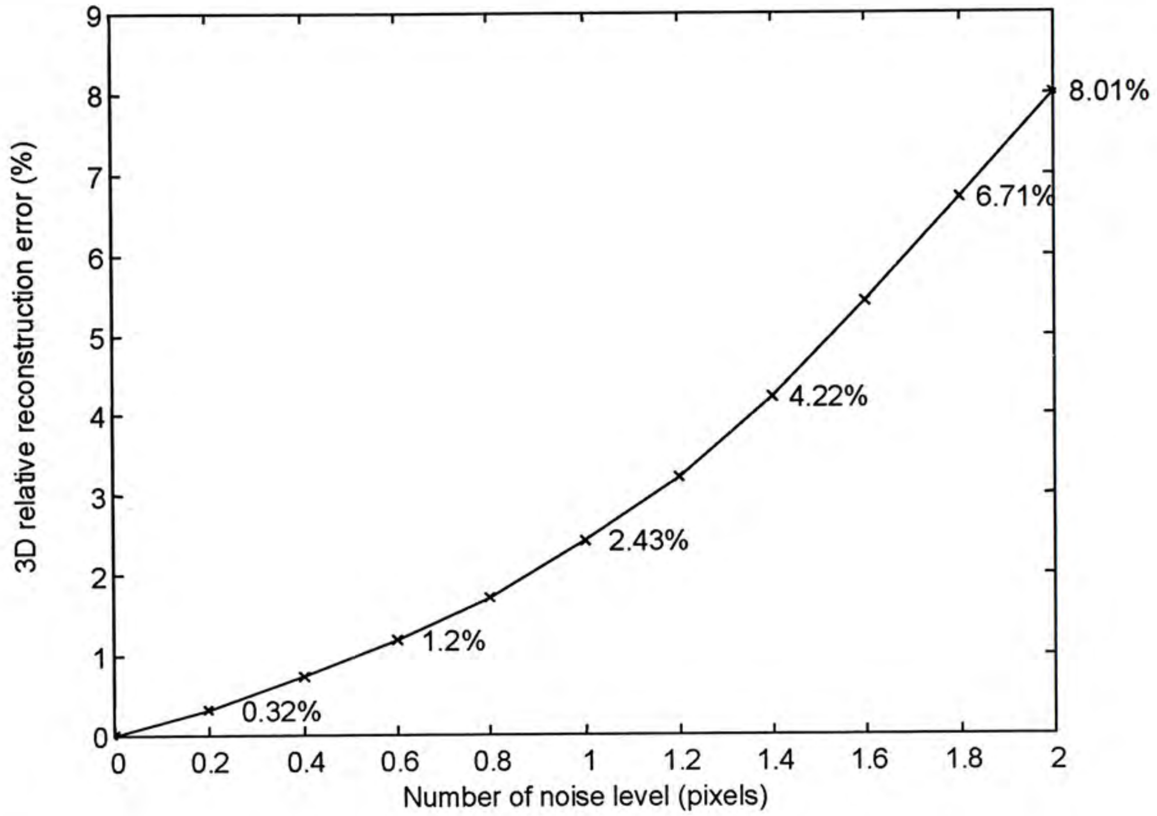


Figure 4.8 : 3D relative reconstruction error of a simulated box

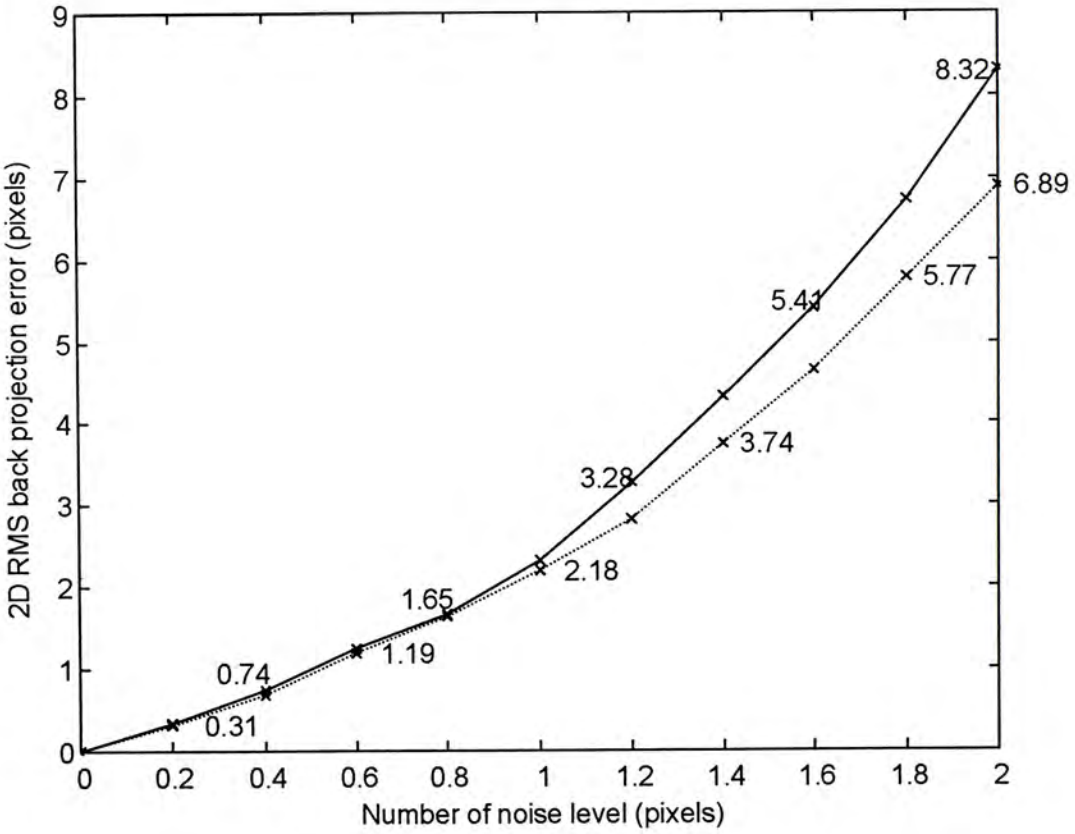
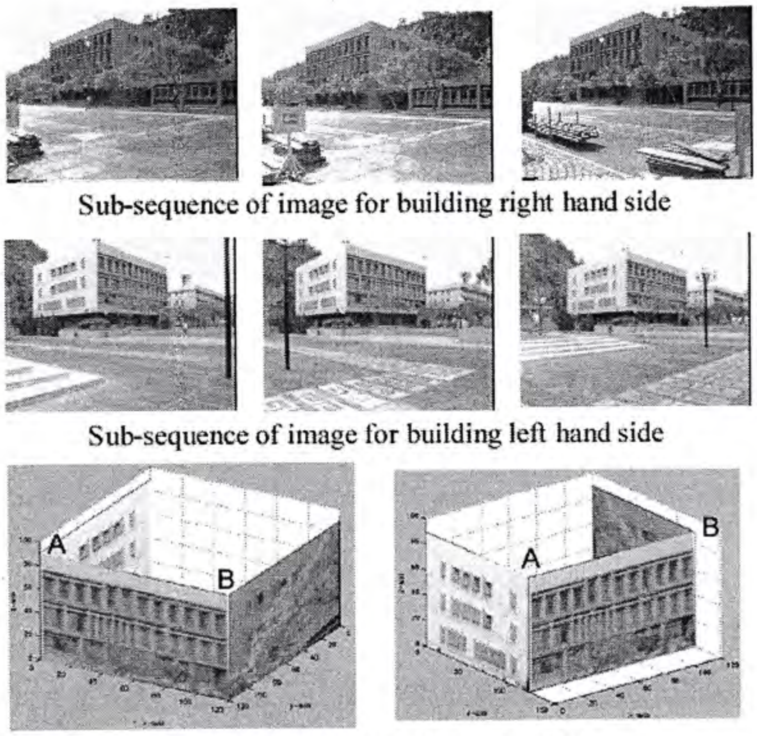


Figure 4.9 : 2D RMS back projection error of a simulated box

[Solid line – pixel error in x-axis direction] [Dash line – pixel error in y-axis direction]

4.6.2 Experiment 2 : A real building

Two image subsequences of a building captured by a camera under different horizontal planar motions is shown in figure(4.10). A For-Tech CCD camera is used. The focal length is 6mm, the image size is 500(H) x 582(V), and the cell size is $12.7μm(H) × 8.3μm(V). The reconstruction results with texture mapping is shown in figuer(4.10). The RMS 2D back projection error is 2.19 pixels in x-direction and 1.89 pixels in y-direction. The orthogonality of corner A is 88.12° and corner B is 87.93° . The average ratio of equal-length lines is 0.982.



Sub-sequence of image for building right hand side

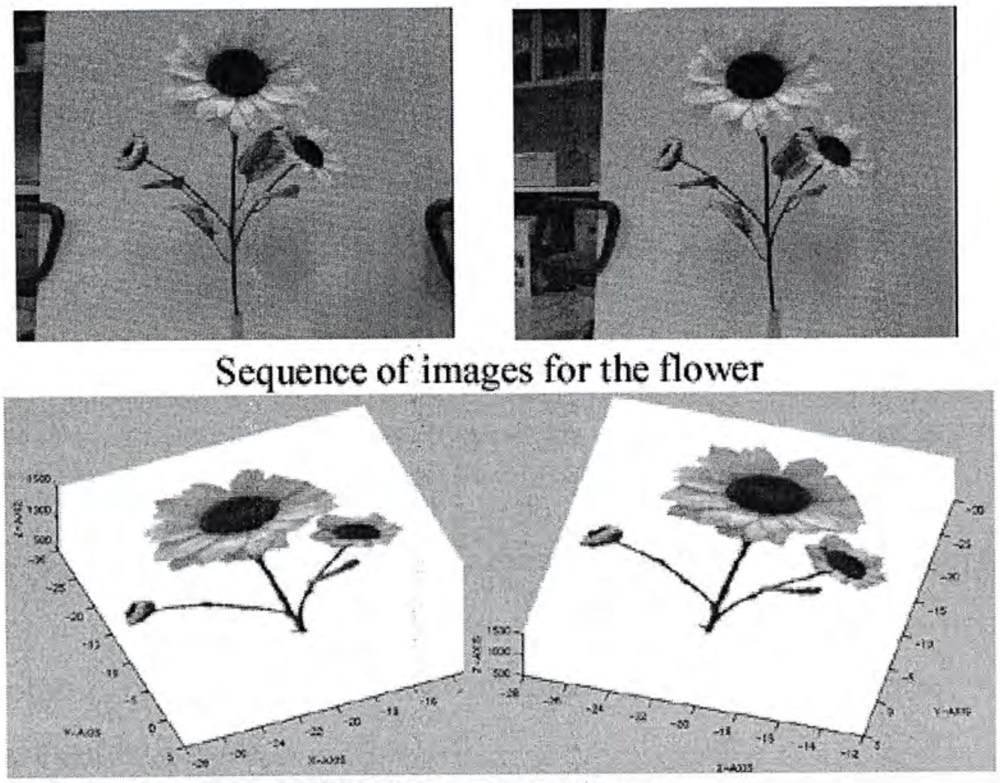
Sub-sequence of image for building left hand side

Complete 3D reconstruction model

Figure 4.10 : 3D Building reconstruction

<u>*4.6.3 Experiment 3 : A sun flower*</u>

We use the same CCD camera to take two subsequence of images of a sun flower with the camera is under horizontal planar motion. In figure(4.11), the reconstructed 3D model of the sun flower is shown. The 2D RMS back projection error is 3.35 pixels in x-direction and 2.93 pixels in y-direction.



Sequence of images for the flower

3D Reconstruction

<u>Figure 4.11 : 3D sun flower reconstruction</u>

4.7 Conclusion

A new simpler algorithm for detecting planar motions was proposed. We have also compared the accuracy and robustness to noise for the two self-calibration methods: (1) camera moves only in horizontal planes and (2) camera moves in three different non-parallel planes. According to our experiments, the former method is much simpler and more robust to noise. Further, when the noise level is very low, the accuracy of the first method is the same as that of the second method.

To obtain a 3D model of a building in city environment, we capture a subsequence of images from viewing points having large overlapping of the visible scene. Different subsequences cover quite different viewing angles. Focal length and other camera parameters are assumed fixed for each subsequence, but the focal length and thus the principal point can vary significantly between different subsequences to cater for the environmental restriction. A partial 3D model is constructed for each subsequence. These partial models are transformed to the same reference frame to form a single complete 3D model of the building. Finally, texture mapping from the real images to the 3D model is then performed to complete the job. Real experiments show that the method is quite efficient and the accuracy is quite good for visualization purpose. We also verify our method in a real experiment on the 3D reconstruction of a paper flower.

Chapter 5

Building Reconstruction using a linear camera self-calibration technique

Overview

Algorithms for camera self-calibration vary depending on the number of images used, the camera model assumed, and the number of intrinsic parameters that need to be recovered. In this chapter, we investigate the linear self-calibration method proposed by Newsam et al [6] for our project on 3D reconstruction of architectural buildings. This self-calibration method assumes that the principal point is known, the camera has square pixels and has no skew. It allows 3D shape to be reconstructed from two images while giving the camera the freedom to vary its focal length. Since the paper by Newsam et al reports only the theoretical work on camera self-calibration, in this chapter, we evaluate the focal lengths obtained from their method with those computed from Tsai's calibration method. Our experimental results show that the focal lengths from the two methods differed by less than 5% and the reconstructed 3D shape was very good in that angles were well preserved. Our future research will focus on the further improvement of optimal 3D reconstruction in the presence of image noise and further development of this method into a package for 3D re-construction of buildings to be used by a layman.

5.1 Introduction

It is now widely known that given a number of corresponding points $\mathbf{q}_i \leftrightarrow \mathbf{q}_i'$,for $1 \leq i \leq n$, the fundamental matrix **F** satisfies the epipolar equation: $\mathbf{q}_i'^T \mathbf{F} \mathbf{q}_i=0$, and can be recovered from corresponding points alone. The 7 degrees of freedom property of **F** allows only 7 camera parameters to be retrieved. Out of these 7 camera parameters, 5 of them are the parameters that describe the relative orientation between the two images (3 rotation angles and 2 components of the translation vector). This leaves us with only 2 unknown intrinsic parameters to be recovered from camera self-calibration. One approach to take from here is to consider more images (of a static scene) taken by a camera that undergoes motion and to restrict the camera from changing its intrinsic parameter setting (i.e. fixed focal length, etc), such as the self-calibration problem tackled by Faugeras et al [4]. An alternative approach is to assume a more simple camera model and various assumptions on certain intrinsic parameters of the camera(s), For example, the principal point is known and the camera has square pixels and has no skew [16, 6, 29]. Thus, the two unknown camera parameters to be recovered are the focal lengths. This latter approach can be taken to be camera self-calibration for a partially calibrated camera whose focal length is variable or camera self-calibration for two distinct partially-calibrated cameras.

The primary aim of our project is to reconstruct architectural buildings from partially calibrated images. The system to be built will be semi-automatic in that prominent image features will be automatically detected by a feature detector but a human operator will be involved to do some manual editing to the image correspondences, if necessary. More image feature correspondences will be automatically established, after the epipolar geometry is recovered, to achieve a dense reconstruction.

From a pair of images taken by a partially calibrated camera to the final metric reconstruction, a number of steps are involved :

(i) Partially calibrate the camera to estimate the principal point,

(ii) Estimate the epipolar geometry by optimally computing the fundamental matrix,

(iii) Retrieve the two unknown focal lengths of the images involved from the fundamental matrix,

(iv) Compute the extrinsic parameters or relative orientation between the two images for triangulation,

(v) Recover the 3D information of each pair of image corresponding points.

To ensure that the final reconstruction, is optimal, the computation in all the precedent step must be optimal. In this chapter, we will present our preliminary results on the study of some of the aforementioned steps. In particular, we will use the linear self-calibration method proposed by Newsam et al [6] and will focus more on step *(iii)* above. We chose to work on this method because the original paper [6] is a theoretical paper without experimental evaluation. More importantly, their method has a number of advantages as described below. First, it allows general camera motion which makes it possible for using a hand-held camera for 3D reconstruction; second, as it is an essentially linear algorithm, and is computationally efficient; third, it allows the focal length to vary so the camera can freely zoom in and out of the scene and has no restriction on its viewing distance and angle to the object(s) of interest. We hope to further develop this method into a package for 3D reconstruction of buildings to be used by a layman. We will present our initial 3D reconstruction in the form of sparse 3D points at this stage. Development of a hybrid intensity-based and partial model-based stereo matching system is currently underway for dense 3D reconstruction.

5.2 Metric reconstruction from partially calibrated images

5.2.1 Partially calibrated camera

To get an optimal estimate of the principal point is difficult. Most Vision researchers therefore simply use the image center (i.e. the center of the image) as the location of the principal point, e.g. [43]. In addition to the problem being ill-posed, some poorly manufactured CCD cameras can have their principal points some distance away from the image center. In order to verify whether the image center is a reasonable approximation of the principal point of our camera, we conducted a number of experiments using Tsai's method [10] to calibrate the principal point that is required by Newsam et al's method [6]. Fig. 5.1 shows the principal point coordinates estimated by Tsai's calibration method for our digital camera whose image buffer is 1800 × 1200 pixels. Discarding the two principal points (872.58 , 604.06) and (916.33 , 698.64) that are slightly off the image center, the average principal point was computed to be (896.33 , 598.64), which is very close to the center of the image buffer, (900 , 600). In these experiments, the focal lengths vary from 995.72 to 4681.52 pixels. We would like to note that the principal point can move slightly when the focal length varies and when the camera undergoes motion. Also, the accuracy of the principal point may not be too important for 3D reconstruction.

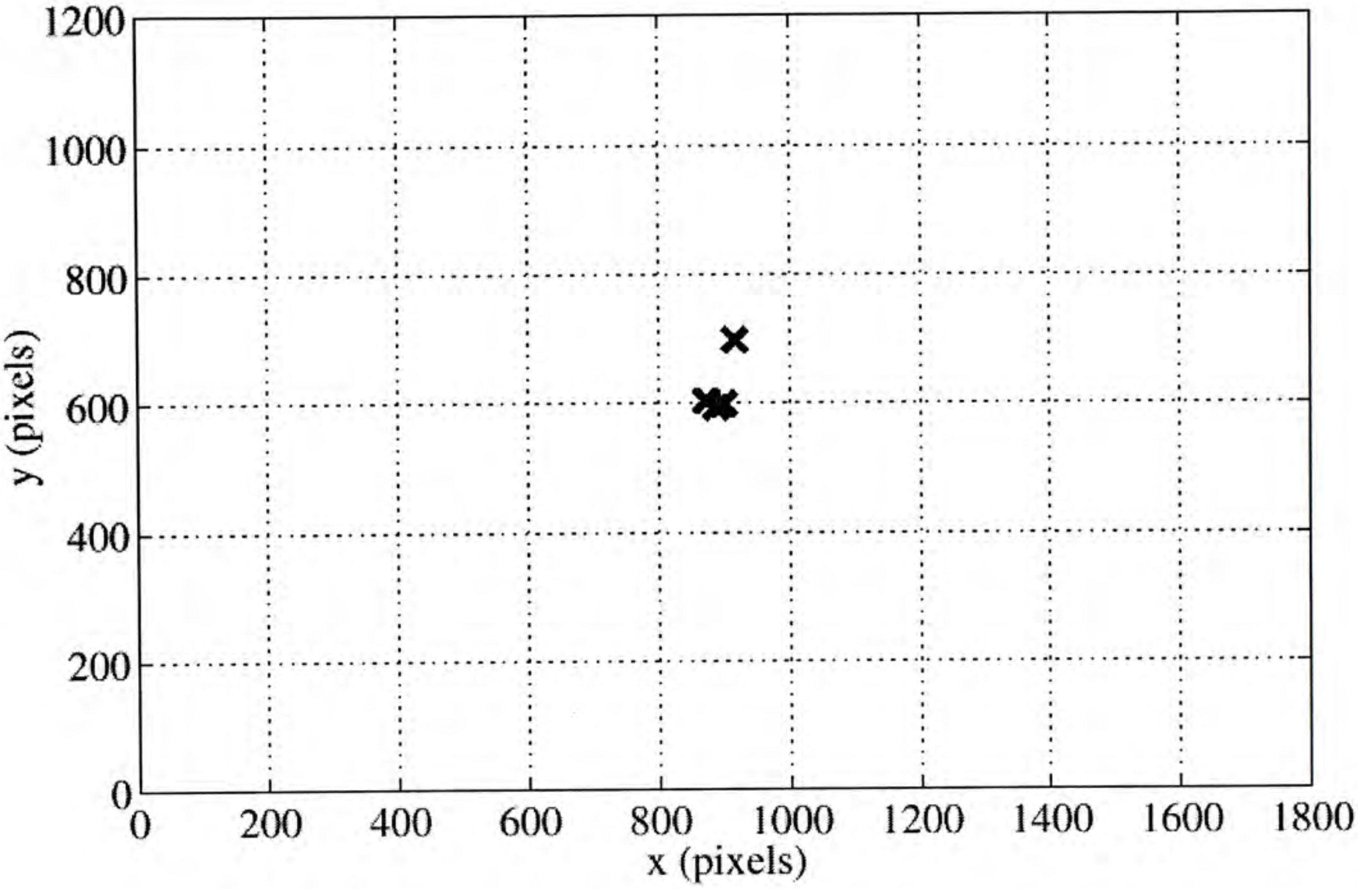


Figure 5.1. The estimated principal point coordinate from Tsai's calibration method.

5.2.2 Optimal computation of F

The essential element of a good 3D reconstruction is an optimally computed fundamental matrix for the recovery of the epipolar geometry. Hartley [36] reports estimating the fundamental matrix using SVD with the image coordinates normalized. Since this linear method only minimizes the algebraic error which has no meaningful geometric interpretation, nonlinear minimization with a proper objective function must be sought. Luong and Faugeras [23] examine two minimization criteria for the nonlinear method: *(i)* minimizing the sum of the distances of all the corresponding points to the epipolar lines, and (ii) minimizing the quantity :

$$\sum_{i=1}^{n} w_i C_i^2$$

where $C_i \equiv \mathbf{q}'^{\mathrm{T}}_{\mathrm{i}} \mathbf{F}\, \mathbf{q}_{\mathrm{i}}$ and w_i is the weighting factor defined as the inverse of the first order approximation of the variance of C_i. We use the fundamental matrix computation software with different minimization criteria provided by Zhang [11] on the web. We were able to examine the estimated fundamental matrices and the subsequent focal length estimation against those obtained from calibrated images via Tsai's method [10]. Our experimental results show that the criterion for minimizing the image re-projection error (i.e. criterion *(i)* of Luong and Faugeras above) gives the best focal length estimation. Experimental results of focal length estimation will be given in Section 5.3.

5.2.3 Linearly recovering two focal lengths from F

Research in camera self-calibration is first investigated by Faugeras et al [4]. They demonstrated that the key to camera self-calibration is to recover the image of the absolute conic (IAC) which is known to be invariant under rigid transformation and contains only the cameras' intrinsic parameters. Thus, recovering the IAC is equivalent to recovering the camera's intrinsic parameters that are essential for metric reconstruction. Faugeras et al's formulation to this self calibration problem assumes that the camera's intrinsic parameters are fixed in the stereo pairs of images and the number of unknown intrinsic parameters is 5: f_u, f_v, s, u_O ,and v_O where f_u and f_v are focal length in horizontal and vertical pixel unit, s is the skew parameter, and (u_o, v_o) is the principal point.

By assuming that the principal point is known (so the origin of the image coordinate system can be set at (u_o, v_o)) and the camera contains square pixels (so $f_u = f_v \equiv f$), the camera matrices **K** and **K'** can be simplified to **K** = diag(f; f; 1) and **K'**= diag(f';f';1) where f and f' are the unknown focal lengths for the two images under consideration. This diagonal form of **K** and **K'** allows the extrinsic parameters to be eliminated nicely from the 3×3 matrix $\mathbf{FF}^{\mathrm{T}}$ and leads to a linear self-calibration method for recovering two focal lengths. The full algorithm of this linear method and the proof of two classes of degenerate stereo configurations for self-calibration are reported in [6]. For the completeness of this manuscript, we summarize the algorithm below.

Given that a rank-2 fundamental matrix F has been obtained, two focal lengths *f* and *f'* can be retrieved as follows:

$$\sum_{k=1}^{3} \sigma_k u_k v_k^T$$

1. Apply the Singular Value Decomposition (SVD) to F .That is, **F** = **USV**^{T}

   where **U** and **V** are orthonormal matrices and **S** = diag(σ1, σ2, 0) with σ1 ≠ σ2 .

2. Let f_i, u_i, v_i be the *i*-th column of **F** , **U** ,and **V** respectively. i_3 is the unit vector $(0,0,1)^T$. Construct the following linear system of equations:

$$\begin{aligned} &\sigma_1^2=(u_1^T f_3)^2 \varpi_1+((u_1^T i_3)^2+(u_3^T i_3)^2)\varpi_2+\varpi_3 \\ &0=(u_2^T f_3)(u_1 f_3)\varpi_1+((u_1^T i_3)(u_2^T i_3)^2)\varpi_2 \cdots\cdots\cdots\cdots\cdots (5.1) \\ &\sigma_2^2=(u_2^T f_3)^2 \varpi_1+((u_2^T i_3)^2+(u_3^T i_3)^2)\varpi_2+\varpi_3 \end{aligned}$$

Here, ω_i's are unknown, intermediate variables that contain the unknown focal lengths as given below:

$$\omega_1= -(f^{-2}-1), \qquad \omega_2=-(f'^{2}-1), \qquad \omega_3=\lambda$$

where λ is an unknown scalar.

3. Let **w** = $(\omega_1; \omega_2; \omega_3)^T$ and , **s**=$(\sigma_1^2, 0, \sigma_2^2)^T$ and **Q** be the 3 × 3 data matrix on the right hand side of (5.1). Then (5.1) can be written as **s** = **Q w**. The 3-vector **w** can be recovered linearly from the above equation and, consequently, the two unknown focal lengths *f* and *f '* can be deduced.

4. Two classes of degenerate stereo configurations that are discovered in [6] are the cases when **Q** is singular:

Class 1: when the optical axes and the baseline are coplanar;

Class 2: when the plane containing one optical axis and the baseline is orthogonal to the plane containing the other optical axis and the baseline.

Experiments reported in this chapter focus on focal length recovery using the linear self-calibration method described above. We carefully set up the experiments such that the degenerate stereo configurations (especially for class 1) mentioned above are avoided (e.g. by enforcing a (small) tilt angle between the two camera orientation).

5.2.4 Essential matrix and triangulation

Having recovered the focal lengths, the essential matrix **E** can be estimated easily using the formula $\mathbf{E} = \mathbf{K}'^{-T}\,\mathbf{F}\,\mathbf{K}^{-1}$. Our current version of triangulation for 3D reconstruction still has room for optimization and is part of the on-going work of our project. Triangulation in the presence of image noise has been discussed by Weng et al [43] and recently by Hartley and Sturm [34]. We will conduct further investigation on this issue.

5.3 Experiments and discussions

Experiments on real images of indoor and outdoor scenes were conducted. Images of indoor scenes were fully calibrated with a calibration target and the application of Tsai's method [10]. The idea was to compare the estimated focal lengths from Newsam et al [6] with those from Tsai [10] where true 3D data were available.

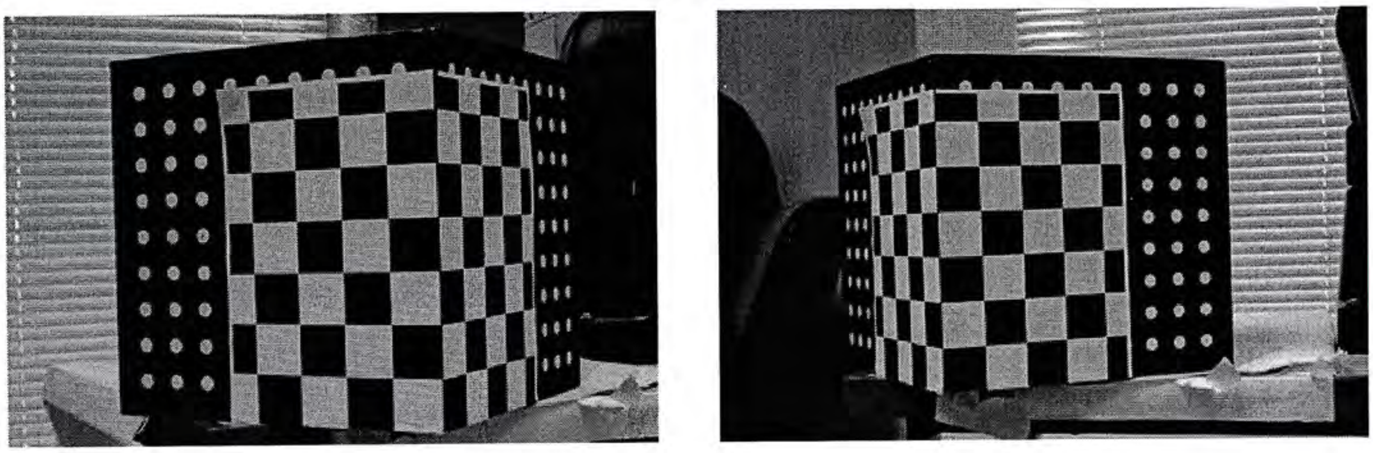


Figure 5.2. A pair of images of a calibration target.

Fig. 5.2 shows a pair of images of a calibration target, with a number of corresponding points superimposed, in one of our indoor experiments. The calibration target has two orthogonal surfaces. The image on the left is frame 1 and the image on the right is frame 30 from an image sequence. Feature points were detected and tracked by a corner detector with some manual editing as a post-process. Using the mean value of the estimated principal points reported in Section 5.2.1 as the principal point of the cameras for the linear algorithm [6], the estimated focal lengths for the two methods for nine different experiments are plotted in Fig. 5.3.

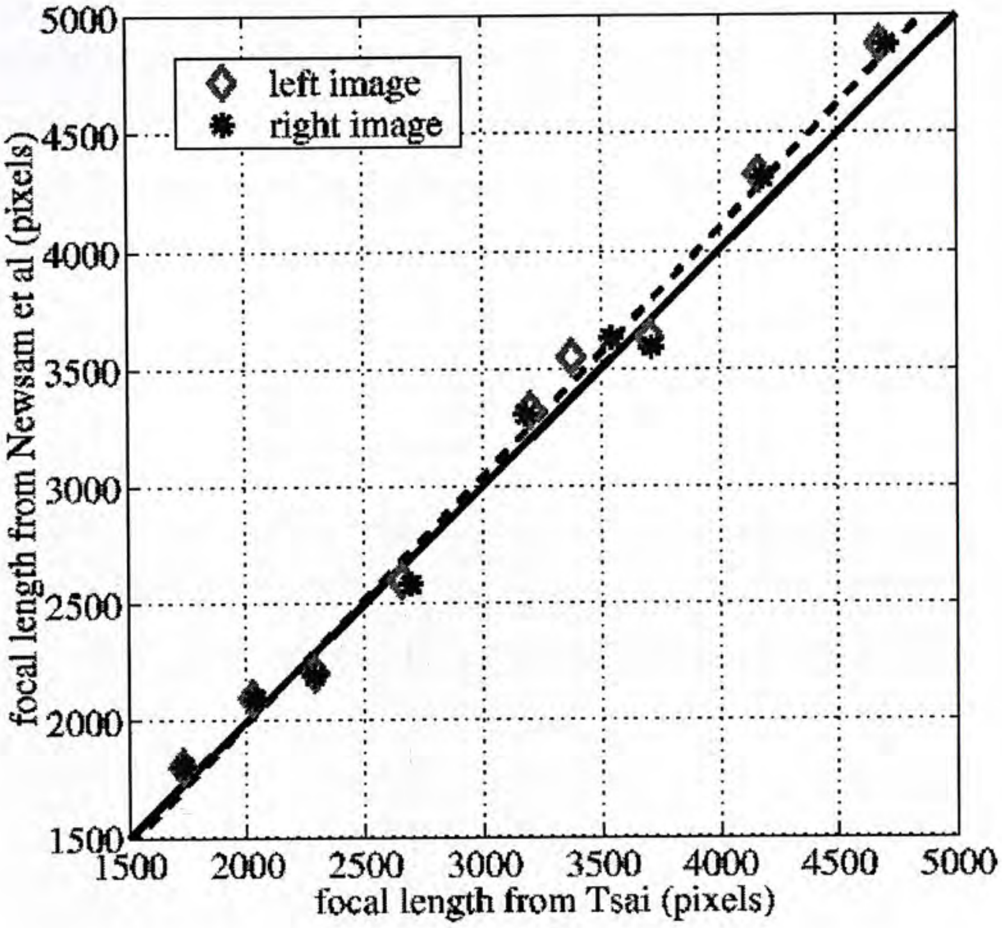


Figure 5.3. The estimated focal lengths from Newsam et al's self-calibration method versus those from Tsai's calibration method for 9 image pairs.

The best fitted line to the computed focal lengths is shown as a dashed line. Its slope was computed to be 0.95, which corresponds to an angle of inclination of 43.55°. The percentage error of angle of inclination from the 45° line (solid diagonal line) is 3.23%. The vertical intercept of the fitted line is −113.33 pixels. One may argue that as we move outside the focal length interval [1500, 5000] the two diagonal lines will be further apart (see Fig. 5.3). However, it is unlikely that the camera will have focal length significantly below 1500 or above 5000 pixels as neither can the focal length of a camera vanish nor can it, for a perspective camera model, be infinite. Moreover, it is simply meaningless to extrapolate the errors in an error analysis this way. The results shown in Fig. 5.3 demonstrate that the linear algorithm[6] performs well in comparison with the calibrated results from Tsai's method for a wide range of focal lengths. Of course, getting a good estimation of each focal length at this stage depends on all the previous stages of camera self-calibration: a reasonable estimate of the principal point and a robust method for computing the fundamental matrix. The focal lengths estimated from the two methods for the nine experiments are tabulated in Table 1, in which the columns for the percentage error were computed as (f_N - f_T)/f_T , where f_T and f_N are the focal lengths from Tsai and Newsam et al respectively. The reconstruction of the sparse 3D points on the calibration target is shown in Fig 5.4. The angle between the two surfaces of the calibration target was estimated to be 88:10° , corresponding to an error of 2.11%.

Fig. 5.5 shows a pair of images of a building which has a large curved surface whose shape is a section of a cylinder. Using the self-calibration method, the focal lengths of the left and right images were computed to be 1804.30 and 1841.90 pixels. The 3D reconstruction of a number of prominent corresponding points is displayed in Fig. 5.6. A good conic fitting program will be required to assess the reconstructed 3D shape in this experiment.

Left image			Right image		
Tsai	Newsam et al	% error	Tsai	Newsam et al	% error
4679.38	4877.61	4.24	4708.14	4876.17	3.57
4168.55	4325.65	3.77	4188.58	4310.44	2.91
3706.14	3632.60	-1.98	3712.18	3593.14	-3.21
3375.03	3549.70	5.18	3544.41	3628.50	2.37
3201.23	3323.73	3.83	3184.26	3307.75	3.88
2657.89	2598.74	-2.23	2697.37	2581.35	-4.30
2290.41	2201.54	-3.88	2287.37	2192.26	-4.16
2025.87	2097.49	3.54	2038.49	2093.15	2.68
1730.10	1803.67	4.25	1732.42	1802.12	4.02

Table 5.1 : The computed focal lengths (in pixels) for the 9 image pairs

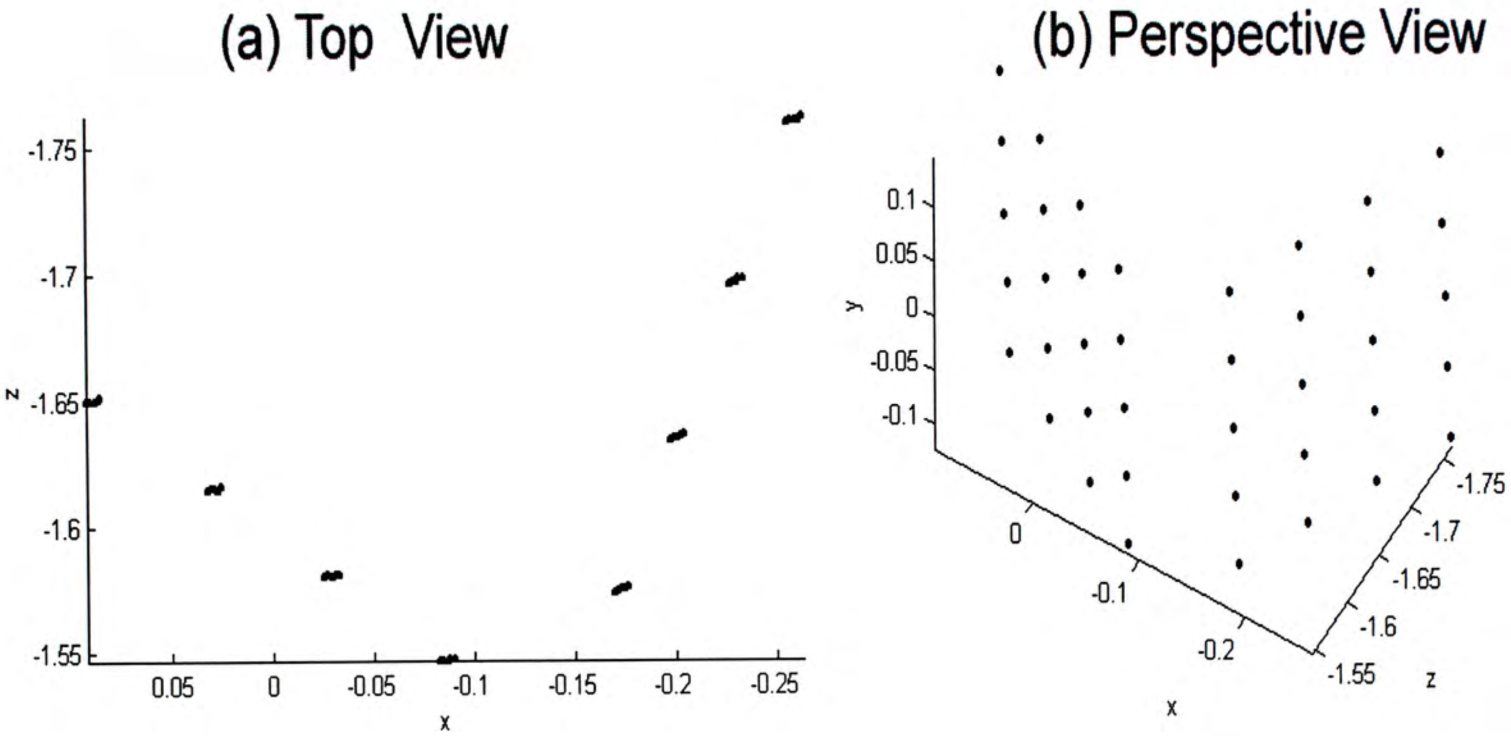


Figure 5.4. Metric reconstruction of the calibration target.

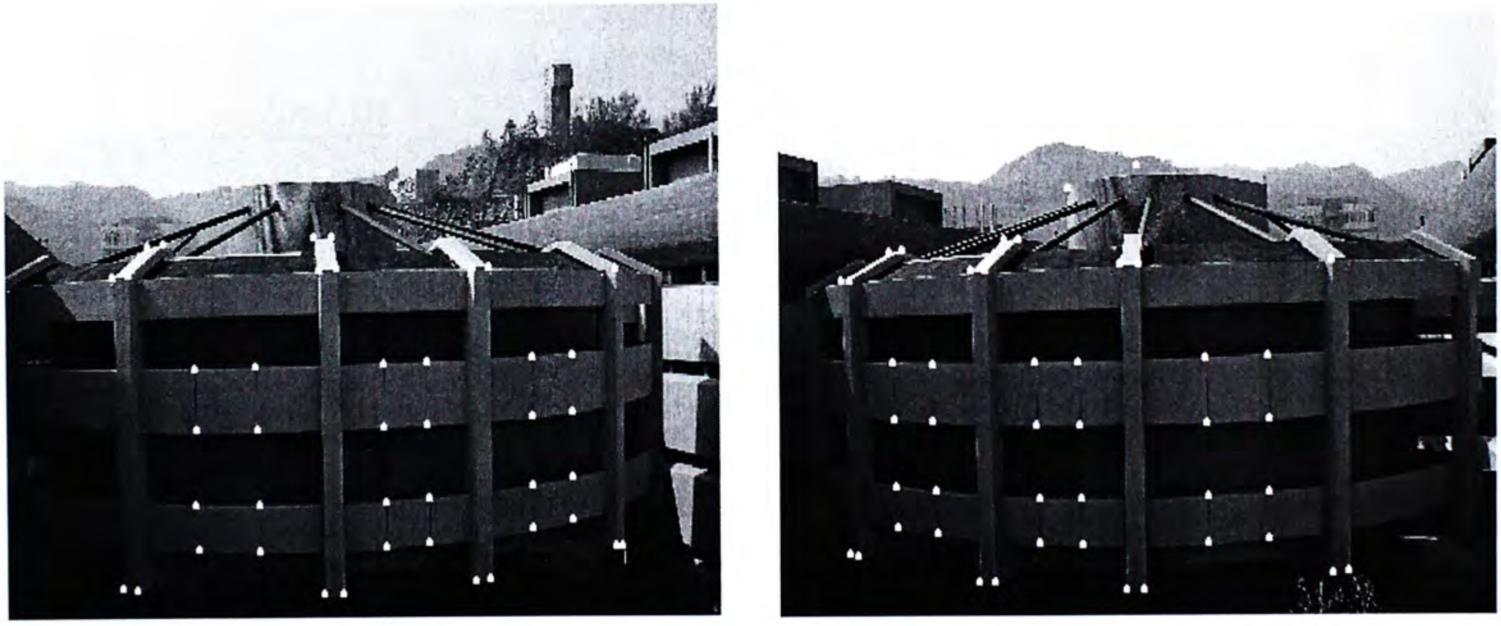


Figure 5.5 A stereo pair of a building that has a large curved surface.

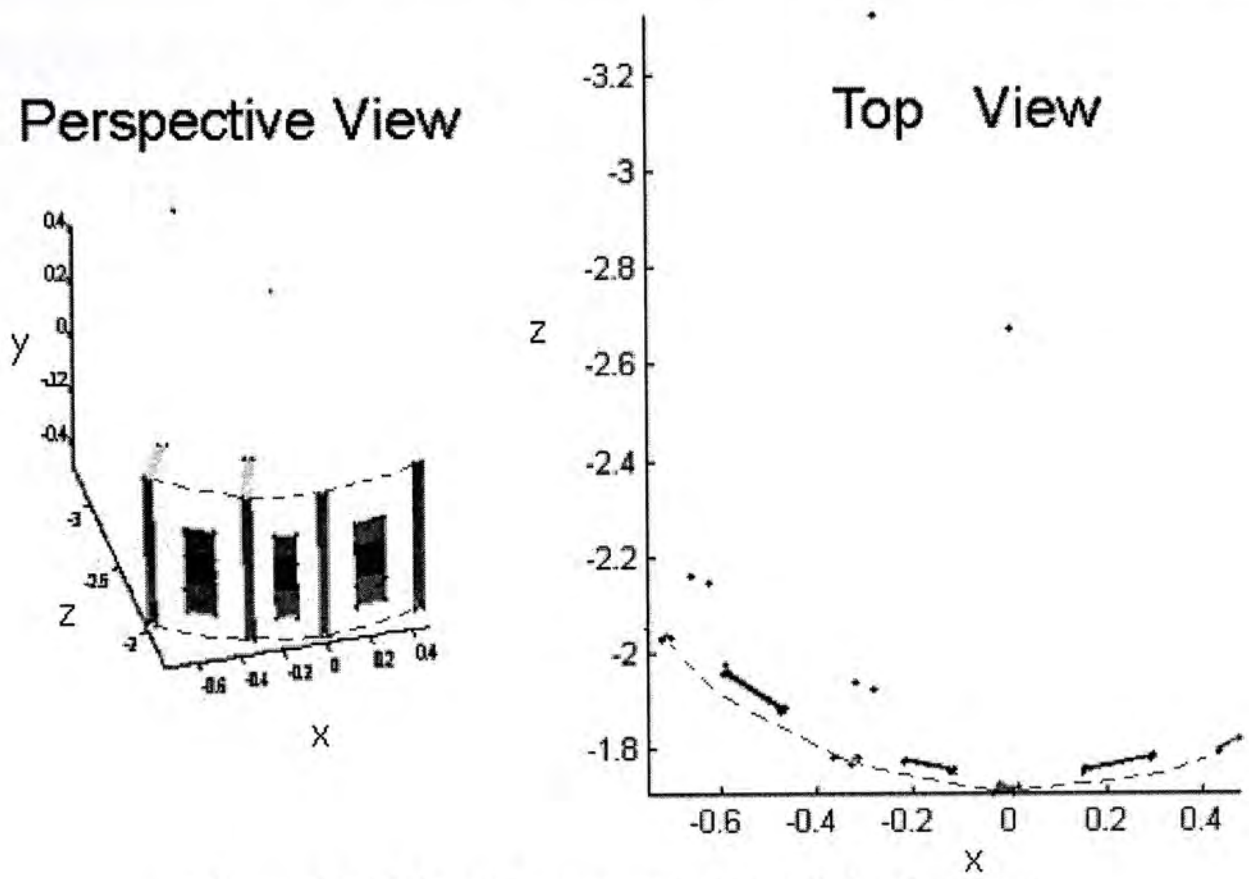


Figure 5.6 Metric reconstruction of the building.

5.4 Conclusion

The linear method of Newsam et al [6] for recovering focal lengths in self-calibration has a number of advantages as discussed in section 5.1 of this chapter. Our preliminary results show that the method is only reasonable in the accuracy for focal length estimate. We believe this accuracy can be further improved by improving the estimation of the fundamental matrix and other steps of the algorithm. Even with this calibration results for the focal lengths, we have shown that we can do some useful 3D reconstruction with good enough accuracy for visualization. Further re-search is required to develop and enhance the method into an easy-to-use package for 3D reconstruction of buildings.

Chapter 6

Refine the basic model with detail depth information by a Model-Based Stereo technique

6.1 Introduction

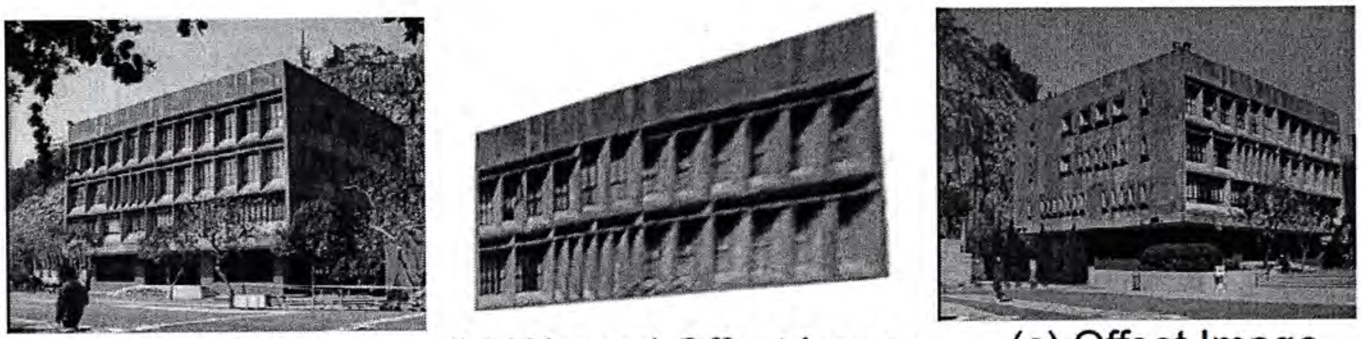
The modeling system described in chapter 2 allows the user to create a basic model of a scene, However the scene will in general have additional geometric detail (such as brickwork and curves) not captured in the model. This chapter presents a method to extend the basic model is generated by self-calibration as shown in chapter 4 and chapter 5 to a detail geometric model by extending the method proposed by Debevec et al's[7].

Model-based stereo differs from traditional stereo in that it measures amount of deviation of the structure of the scene from the approximate model, rather than to measure the structure of the scene without any prior information. The model serves to place the images into a common frame of reference that makes the stereo correspondence easier.

As in traditional stereo, given two images (which we call key and offset image), model-based stereo computes the associated depth map for the key image by determining corresponding points in the key and offset images. Like many stereo algorithms, Debevec et al's proposed method is correlation-based. It attempts to determine the corresponding point in the offset image by comparing small pixel neighborhoods around the points. As such, correlation-based stereo algorithms generally require the neighborhood of each point in the key image to resemble the neighborhood of its corresponding point in the offset image.

The problem we face is that when the key and offset images are taken from relatively far apart, it is difficult to get the corresponding pixel with accuracy. In Fig 6.1 (a) and (c), pixel neighborhoods toward the right of the key-image are foreshortened horizontally by nearly a factor of four in the offset-image.

The key observation in model-based stereo is that even though two images of the same scene may appear very different, they appear similar after being projected onto an approximate model of the scene. If we project the offset image onto the model and view it from the position of the key image produces what we call the warped offset image that appears similar to the key image. The geometrically detailed scene in Figure 6.1 was modeled as two flat surfaces with our modeling program, which also determined the relative camera positions. As expected, the warped offset image (Fig.6.1b) exhibits the same pattern of foreshortening as the key image.



(a) Key Image (b) Warped Offset Image (c) Offset Image

Figure 6.1 : (a) and (c) Two image of the same building in The Chinese University of Hong Kong

In model-based stereo, the neighborhoods are compared between the key and warped offset images rather than the key and offset images. When a correspondence is found, it is simple to convert its disparity to the corresponding disparity between the key and offset images and calculate the point's depth. The advantage of the Debevec's et al [7] proposed method is on simplifying stereo correspondence. The reduction of differences in foreshortening is just one of several ways that the warped offset image simplifies stereo correspondence.

Since images taken from relatively far apart can be compared, so the depth estimates are far less sensitive to noise in image measurements. And the places where the model occludes itself relative to the key image can be detected and indicated in the warped offset image easily. On the other hand, the linear epipolar geometry (section 6.2.1) exists between the key and warped offset images, despite the warping. In fact, the epipolar lines of the warped offset image coincide with the epipolar lines of the key image.

6.2 Model-Based Epipolar geometry

6.2.1 Overview

In traditional stereo, the *epipolar constraint* (see [43]) is often used to constrain the search for corresponding points in the offset image to a linear search along an epipolar line. This reduction of the search space from two dimensions to one not only speeds up the algorithm, but it also greatly reduces the number of opportunities to select a false matches. This section shows that taking advantage of the epipolar constraint is no more difficult in the model-based stereo case, despite the fact that the offset image is a non-uniformly warped version of the original offset image.

Fig. 6.2 shows the epipolar geometry for model-based stereo. If we consider a point P in the scene, there is a unique *epipolar plane* which passes through P and the centers of the key and offset cameras. This epipolar plane intersects the key and offset image planes in *epipolar lines* e_k and e_o. If we consider the projection p_k of P onto the key image plane, the epipolar constraint states that the corresponding point in the offset image must lie somewhere along the offset image's epipolar line.

In model-based stereo, neighborhoods in the key image are compared to the warped offset image rather than the offset image. Thus, to make use of the epipolar constraint, it is necessary to determine where the pixels on the offset image's epipolar line project to in the warped offset image.

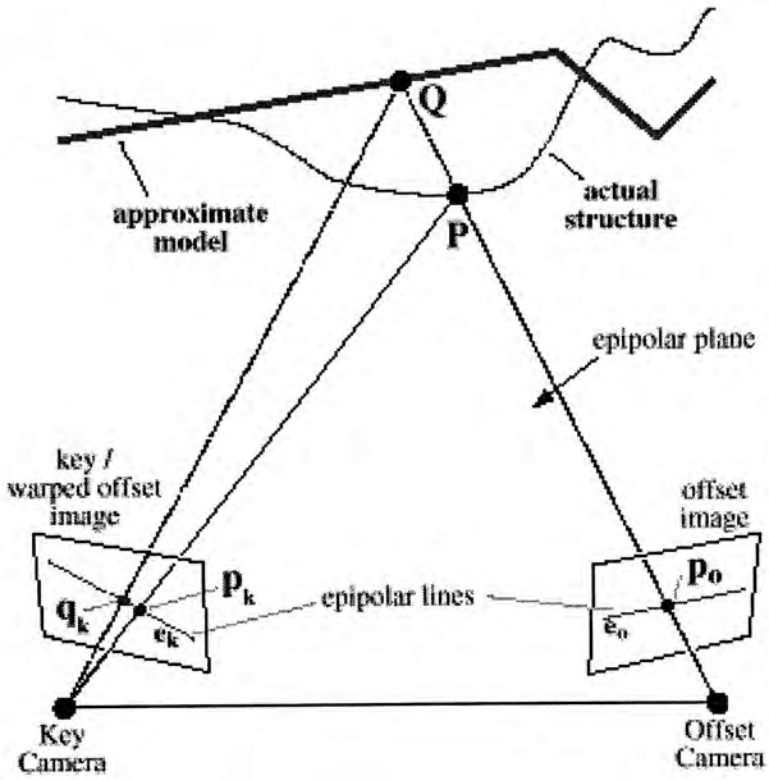
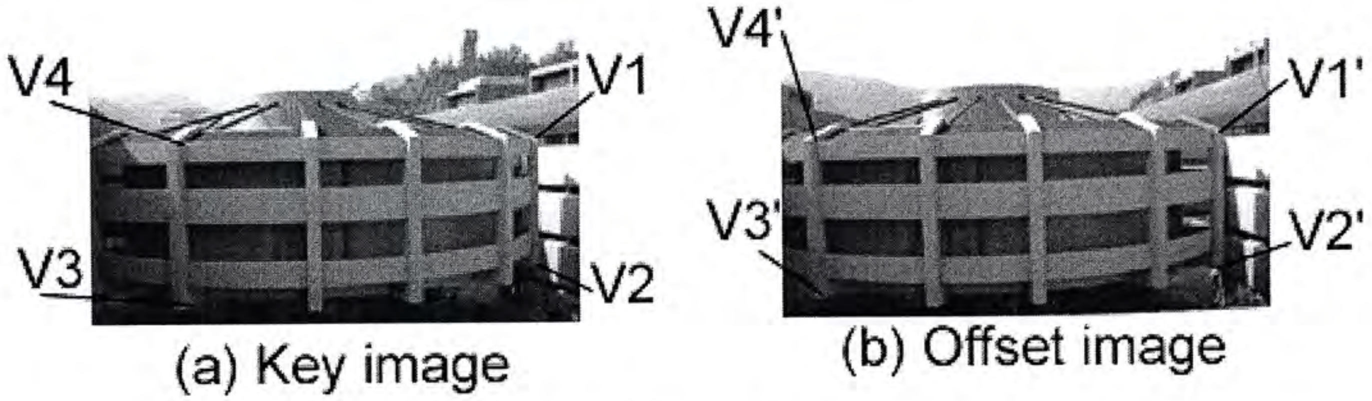


Figure 6.2 : The epipolar geometry for model-based stereo. This figure illustrates the formation of the warped offset image, and shows that points which lie on the model exhibit no disparity between the key and warped offset images. Furthermore, it shows that the epipolar line in the warped offset image of a particular point in the key image is simply that point's epipolar line in the key image. The text of this chapter provides a more detailed explanation of these properties.

The warped offset image is formed by projecting the offset image onto the model, and then reprojecting the model onto the image plane of the key camera. Thus, the projection p_o of P in the offset image projects onto the model at Q, and then re-projects to q_k in the warped offset image. Since each of these projections occurs within the epipolar plane, any possible correspondence for p_k in the key image must lie on the *key* image's epipolar line in the warped offset image. In the case where the actual structure and the model coincide at P, p_o is projected to P and then re-projected to p_k , yielding a correspondence with zero disparity.

6.2.2 Warped offset image preparation



(a) Key image

(b) Offset image

Figure 6.3 : Two images (a) Key image and (b) Offset image of the same building in The Chinese University of Hong Kong

Assume we have image 1 (key) and image 2 (offset) as shown in the figure 6.3. The warped offset image project the regions (V1', V2', V3', V4') to (V1, V2, V3, V4). To get the warped offset image, we need to compute the "warping matrices" that maps regions (V1', V2', V3', V4') to (V1, V2, V3, V4).

Let H be the 3x3 matrices that represents this homography , then we have

$$\mathbf{HV_i'} = \alpha_i \mathbf{V_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.1$$

Where Vi ↔ Vi', αi is an unknown scalar that can be eliminated from the equation.

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} V_{ix}' \\ V_{iy}' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_i V_{ix} \\ \alpha_i V_{iy} \\ \alpha_i \end{bmatrix} \cdots\cdots\cdots (6.2)$$

From equation (6.1), we got the equation as following :

Expending equation (6.2), we get the equations (6.3-6.5) as follows :

$$H_{11}Vix' + H_{12}\,Viy' + H_{13} = \alpha_i Vix \ldots\ldots\ldots\ldots (6.3)$$

$$H_{21}Vix' + H_{22}\,Viy' + H_{23} = \alpha_i Viy \ldots\ldots\ldots\ldots (6.4)$$

$$H_{31}Vix' + H_{32}\,Viy' + H_{33} = \alpha_i \ldots\ldots\ldots\ldots\ldots (6.5)$$

To eliminate α_i , we substitute (6.5) into (6.3) and (6.4) and got equations (6.6 - 6.7) as following :

$$H_{11}Vix' + H_{12}\,Viy' + H_{13} - H_{31}Vix'\,Vix - H_{32}\,Viy'\,Vix - H_{33}\,Vix = 0 \;\ldots\ldots\ldots\; (6.6)$$

$$H_{21}Vix' + H_{22}\,Viy' + H_{23} - H_{31}Vix'\,Viy + H_{32}\,Viy'\,Viy - H_{33}\,Viy = 0 \;\ldots\ldots\ldots\; (6.7)$$

Thus, each corresponding point Vi ↔ Vi' gives us 2 equations for solving for the elements Hij of the homography H.

Rearranging the equations (6.6 and 6.7) into matrix-vector form as following :

$$\left[\underbrace{\begin{matrix} V_{ix}' & V_{iy}' & 1 & 0 & 0 & 0 & -V_{ix}'V_{ix} & -V_{iy}'V_{ix} & -V_{ix} \\ 0 & 0 & 0 & V_{ix} & V_{iy} & 1 & -V_{ix}'V_{iy} & -V_{iy}'V_{iy} & -V_{iy} \end{matrix}}_{9\times 2}\right] \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ H_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdots\cdots\cdots (6.8)$$

To solve the vector $[H_{11}\ldots\ldots H_{33}]^T$, we need at least 4 corresponding points (Each point gives 2 equations; so having 8 equations) to get a 8x9 matrix. Where the matrix have rank=8, the vector $[H_{11}\ldots\ldots H_{33}]^T$ can be solved by using SVD[43].

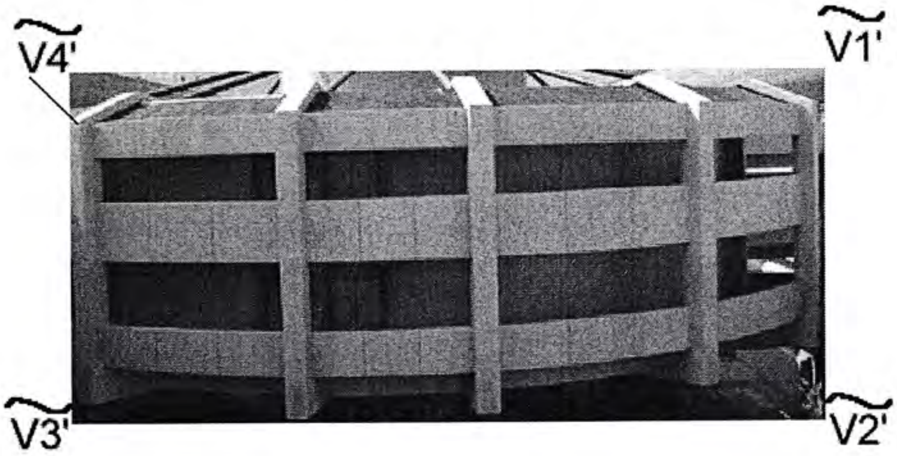


Figure 6.4 : Warped Offset Image

After we have computed matrix **H**, we create a warped offset image as shown in figure (6.2). For every point P inside area (V1',V2',V3',V4') in figure 6.1, we compute

$$\begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} = H \begin{pmatrix} P_x' \\ P_y' \\ P_z' \end{pmatrix} \cdots\cdots\cdots\cdots (6.9)$$

and [Px' Py' Pz']^T is normalised to x'=[Px'/Pz'] and y'=[Py'/Pz'].

The warped offset image can be formed by puting the intensity value at [Px Py] in the offset image to the pixel location [x' y'] in the warped offset image. At last the warped offset image as shown in figure 6.2.

6.2.3 Epipolar Line Calculation

From the figure 6.2, we observe the point p_o and q_k are in fact a pair of corresponding point. However, since we assume that the 3D surface is planar, we found p_k as the corresponding point for p_o. ie, $p_o = H\ q_k$ as equation (6.9).

To find a location of p_k, we can use Epipolar geometry to reduce the search area from 2D into 1D and increase the searching efficiency. To find the accurate epipolar line equation, we use the optimization method to compute the fundamental matrices (**F**) as describe in chapter 5 (section 5.2.2). The following are the two different proposed method to find a epipolar line.

6.2.3.1 Method (1) – Epipolar line from epipole (e_o and e_k) in (Fig 6.2)

Since the fundamental matrices **F** has been calculate before, so we have the epipolar equation as follow :

$$\mathbf{x}'^{T}\,\mathbf{F}\,\mathbf{x} = 0 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.10)$$

where **x**' is the corresponding pixel coordinate in key image and **x** is the coordinate in offset image.

Then we have $\mathbf{e_o} = \text{null}(\mathbf{F}^T)$ and $\mathbf{e_k} = \text{null}(\mathbf{F})$,

After that, we can get the epipolar line by join $\mathbf{e_k}$ and $\mathbf{q_k}$ in the Key image.

6.2.3.2 Method (2) - Get Epipolar line by Fundamental matrices(F)

Use point p in offset image to compute the epipolar line(δ) in the key image as following: Since $\mathbf{x}'^{T}\,\mathbf{F}\,\mathbf{x} = 0$, **x**' in the key image and x-point in the offset image,

$$\delta = \mathbf{Fp} = [\delta_x\ \delta_y\ \delta_z]^T \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.11)$$

Then the epipolar line equation in the key image is simply as the following :

$$\delta_x x + \delta_y y + \delta_z = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.12)$$

After we got the epipolar line equation, we need to find the matching point ($\mathbf{p_k}$) along the computed epipolar line in the key image as describe in section 6.2.4.

6.2.4 The matching algorithm to finding a actual corresponding point (p_k)

Once the warped offset image and the epipolar line in the key image is formed, stereo matching proceeds in a straightforward manner between the key and warped offset images to find out the actual corresponding point (**p_k**). The one complication is that the two images are not rectified in the sense of the epipolar lines being horizontal; instead, the epipolar lines which need to be searched along converge at a finite epipole. Since this epipole can be either within or outside of the borders of the key image, special care must be taken to ensure that the epipolar lines are visited in a reasonable fashion. The approach taken in this work is to traverse the pixels of the border of the key image in a clockwise manner, examining the corresponding epipolar line between the current border pixel and the epipole at each step.

The matching window we used was a 7 × 7 pixel neighborhood, and the matching function we used was the normalized correlation between the forty-nine pixel intensity values in the two regions. Normalized correlation makes a good stereo matching criterion because it is not sensitive to overall changes in brightness and contrast between the two images.

6.2.5 Actual 3D point generated by the Triangulation

Once we get the actual corresponding point (**p_k**) as shown in section 6.2.4, we can use triangulation to get the actual 3D point. Our current version of triangulation for 3D reconstruction still have room for optimization as describe in section 5.2.4.

6.3 Summary of the Algorithm

To get the actual corresponding point and the detail geometric model, the detail of the algorithm is described as follows :

(1) Model the curve surface of a building using a cylindrical section.

(2) Estimate four 3D points on the curve surface using stereo vision.

(3) Estimate the other 3D points on or close to the surface using the model-based approach.

(4) Prepare the key image and offset image for the same building.

(5) Use four corresponding point in key image (V1,V2,V3,V4) and (V1', V2', V3', V4') in the offset image to get the Homography (**H**).

(6) Project all the point within the image area (V1', V2', V3', V4') in the offset image into new image called "warped offset image".

(7) Use the Fundamental matrices (**F**) that we found between the key image and the offset image to calculated the epipolar line equation in the key image by the method 1 (Section 6.2.3.1) or method 2 (Section 6.2.3.2).

(8) Since Debevec et al's [7] have suggested that the epipolar line is identical between the warped offset image and the key image. Then the epipolar line equation in the key image is identical to warped offset image.

(9) Extract a point (**P$_o$**) from the offset image as shown in figure 6.5 in the coordinate (x,y) and transfer the color (gray level)of this point to warped offset image as point (**P$_o$'**) with coordinate (x",y"). The color (gray level) of this point (**P$_o$'**) is the reference point for the future stereo matching.

(10) The point (**P$_o$**') in the warped offset image will have the same coordinate as the point (**q$_k$**) in the key image. Then the coordinate (x",y") = (x',y').

(11) The actual corresponding point (**P$_k$**) is found by the stereo matching (section 6.2.4) through the epipolar line in the key image. Where the actual corresponding point is (**P$_k$**) in the coordinate (xa,ya).

(12) Use triangulation method to get the actual 3D reconstruction with texture by the actual corresponding point (**P$_k$**) and the point (**P$_o$**) with up to scale.

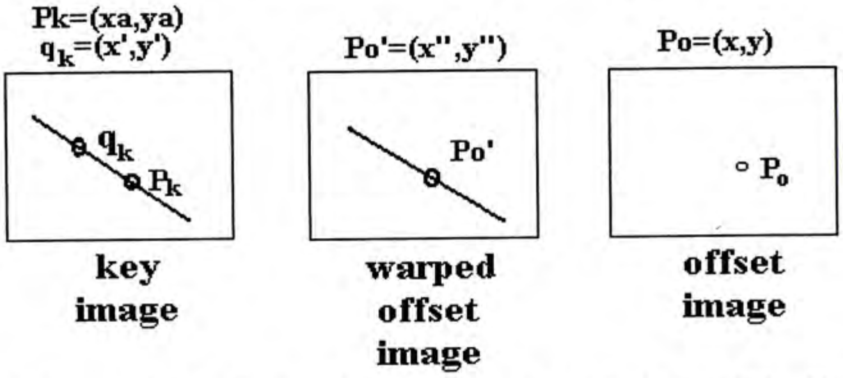


Figure 6.5 : Actual Corresponding Point Finding

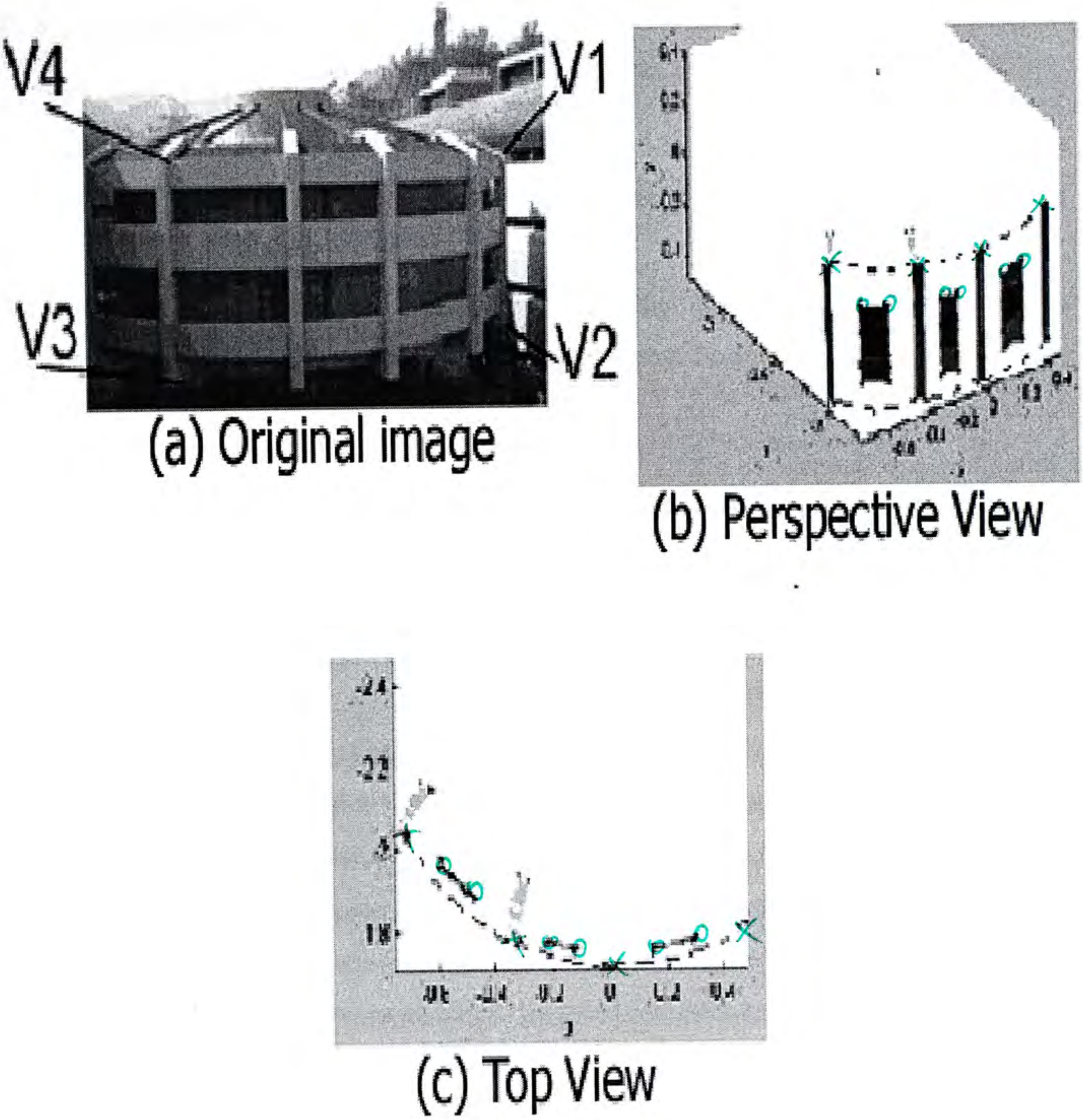
6.4 Experiments and discussions

Experiment on real images of buildings with curve surfaces were conducted. The camera was calibrated for all parameters except the focal length. The focal length was estimated using the self-calibration method of Newsam et al[6].

Figure 6.3 shows a pair of images of a curve building which has a large curved surface which can be modeled by a section of a cylinder. Using the self-calibration method [9], the focal lengths of the camera for the key and offset images were computed to be 1804.30 and 1841.90 pixels respectively.

To verify our algorithm, we get the homography (**H**) between the 4 corresponding points (V1-V4) in the key image and (V1'- V4') in the offset image as shown in figure 6.3 . A new warped offset image is computed using this homography as shown in figure 6.4. From the algorithms that describe in section 6.3, we use the warped offset image and fundamental matrices(**F**) between the key image and the offset image to get the 3D reconstruction of a numbers of detail geometry of the cylindrical surface as shown in Figure 6.6. While we use stereo vision to get the 3D points V1-V4 and use model-based approach to get another 20 3D points lying approximately in a cylindrical surface. These points are labeled with the symbol "x" and "o". It is noted that V1, V2, V3, V4 and these puts marked with "x" is one surface while these puts marked with "o" is another surface. This is particularly observes in fig 6.6 with shows the top view.

The result is extremely important as it demonstrates that our method can detect small fluctuation in depths on the surface of a building.



(a) Original image

(b) Perspective View

(c) Top View

Figure 6.6 : 3D Reconstruction of the curve shape building

6.5 Conclusion

To conclude, we have presented an efficient method to detect small depth changes on a surface. We used the model-based idea of Debevec et al's[7] for this purpose.

The advantage of this approach as follows. Our experiment show that the model-based idea can largely remove the error due to foreshortening. Second, the actual corresponding point (P_k) and the initial estimate of the corresponding point (q_k) is close and on the same epipolar line. This can increase the search efficiency in the correspondence computation. matching process. On average, a pair of corresponding point can found within 0.05 second on a SGI indigo II. When compared with the results shown in Faugeras experiments[5], we found that our algorithm is more robust and efficient to get the detail geometry of a building.

Chapter 7

Conclusions

7.1 Conclusion and Discussion

We have proposed a general frame work for 3D reconstruction of buildings in this thesis. An initial basic 3D model is obtained using either a 1D self calibration technique proposed by Faugeras et al [5] or a linear method due to Newsam et al's[6]. This initial 3D model is then refined by adding fine depth variations onto the initial 3D model by extending the model based stereo technique of [7]. Extensive real experiments are performed to verify our approach. Good results are obtained.

The first method of building an initial model is based on the algorithms proposed by Faugeras et al[5] for camera self-calibration by restricting the camera to planar motions. A tripod mounted camera with variable focal length was used to capture the images of a building at different distances and viewing angles dictated by the environment. This set up allows the camera to move in many planes by rotating the tripod and tilting the camera. A new and simpler planar motion detection algorithm was proposed. We have also compared the accuracy and robustness to noise for the two versions of the self-calibration method: (1) the camera moves only in horizontal planes to compute three parameters (focal length and the image coordinates of the principal point) and (2) the camera moves in three different non-parallel planes to compute five parameters (focal length in horizontal and vertical direction, image coordinates of the principal point and the skew). We found the estimated average camera parameters error is smaller (0.9%) if the camera is moving in one horizontal plane only. On the other hand, the average parameter error is large (7.5%) if the camera is moving in three different non-horizontal planes. The noise level is assumed to be less than 1.5 pixel. In real experiments, the average reconstruction error is 2.1% for a building. This compared favorably with the 2.3 % errors of the method by Pollefeys[29] and the 2.9% error for the method by Faugeras [5].

The second method of building an initial model is based on a linear algorithm proposed by Newsam et al's[6] for self calibration. No experiment, simulation or real, was reported by their paper. We adopted their algorithms for our real experiments on the 3D reconstruction of a buildings. From our experiments, the reconstruction error is 2.2%. This method allows 3D shapes to be reconstructed from two images while giving the camera the freedom to vary its focal length. We believe this accuracy can be further improved by improving the estimation of the fundamental matrix and other steps of the algorithm. The accuracy of focal lengths estimation is not very high (4.73%). However, the 3D reconstruction error(2.2%) is small enough for visualization. When compared with the non-linear self-calibration method using absolute conic this method is fast. However, it can only take care of the change in focal length.

A model-based approach making use of an idea in Debevec et al's[7] is used to add fine depth variations on an initial model obtained above. Our experiments show that it is very effective in capturing the changes in depth on a model surface. 3D points on two different cylindrical surfaces in a building are found by the method as shown in figure 6.6. There are some other advantages of this approach. First, our experiment shows that the errors due to foreshortening can be largely removed. Second, we observe that the distance between the actual corresponding point (**P**$_k$) and its initial estimated point (**q**$_k$) (in figure 6.2) are usually close to each other and are lying on the same epipolar line. Obviously, this will increase the search efficiency and accuracy in the correspondence computation.

In conclusion, we have found that our simpler methods of self-calibration can produce similar results as the complex self-calibration methods of Faugeras [4] and Pollefeys [41] in our experiments on 3D reconstruction of a building. Fine depth details on a largely planar surface can be recovered using our approach.

7.2 Future Work

Although the theory and algorithms presented here have been shown to perform successfully, there are still many areas which require further work. Uncertainty analysis which gives a measure of the confidence in the computed parameters is important. However, it is not well understood in the area of self-calibration and scene reconstruction. Some work could be done in the future in this direction. The accuracy of the reconstruction model is affected by the estimation of the essential matrix and the triangulation method. Our current version of triangulation for 3D reconstruction is not yet optimal. Triangulation in the presence of image noise has been discussed by Weng et al [43] and recently by Hartley and Sturm [34]. We will conduct further investigation on this issue. In addition, we shall conduct further experiments on buildings with different curve surfaces such as hemisphere, cone and the onion shape of the Moscow palace.

BIBLIOGRAPHY

[1] Beardsley, P. Zisserman. A, Murray D.. Navigation using affine structure and motion. ECCV, P.84-96. Springer-Verlag,1994.

[2] Taylor M. Visually Guided Grasping. PhD thesis, University of Oxford, England,1995.

[3] Zisserman, A., Forsyth,D., 3D object reconstruction using invariance. Artificial Intelligence, 78(1-2):239-287,1995.

[4] O.Faugeras, Q-T Luong, and S. Maybank. Camera Self-calibration : theory and experiments. ECCV, LNCS 588, pages 321-334, Springer-Verlag,1992.

[5] Faugeras. O.D., L. Quan and P.Sturm. Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera. ECCV, 36-52, 1998.

[6] GN. Newsam, D.Q.Huynh, M.J.Brooks, and H.-P. Pan. Recovering unknown focal lengths in self-calibration : An essentially linear algorithms and degenerate configuration. In Int. Archives of Photogrammetry and remote sensing, Volume XXXI, part B3, commission III, pages 575-580. Jul 1996.

[7] P.E.Debevec, Camillo J.Taylor and Jitendra Malik. Modeling and Rendering Architecture from Photographs : A hybrid geometry- and image-based approach. SiGGRAPH, 1986.

[8] Bailey Chou, L.Lu, H.T.Tsui, Z.Y.Hu., A practical 3D Euclidean reconstruction method by an un-calibrated camera under planar motions. Technical report. Electronic department of The Chinese University of Hong Kong. 1999.

[9] D.Q.Huynh. Bailey Chou, H.T.Tsui, Semi-autometic metric reconstruction of buildings from self-calibration : Preliminary results on the evaluation of a linear camera self-calibration method. ICPR,2000.

[10] R.Y.Tsai. A Versatile Camera Calibration Technique For High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV camera And Lenses. IEEE Journal of Robotics and Automation, RA-3(4):323-344, Aug 1987.

[11] Z.Zhang, R.Deriche, O.Faugeras, and Q-T Luong. A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry. Artificial Intelligence, 75(1-2):87-120,1995.

[12] MCLauchlan P. and Murray D. A Unifying Framework for Structure from Motion recovery from Image Sequences. ICCV, Page 314-320,1995.

[13] Z.Zhang and O.Faugeras. 3D Dynamic Sene Analysis, Springer-Verlag,1992.

[14] Semple, J. and Kneebone, G. Algebraic Projective Geometry. Oxford University Press 1979.

[15] O.Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? ECCV, LNCS 588, page 563-578. Springer-Verlag,1992.

[16] R.Hartley. Estimation of relative camera positions for uncalibrated cameras. ECCV, page 579-587, 1992.

[17] Mundy J. and Zisserman, A geometric Invariance in Computer Vision. MIT Press, 1992.

[18] D.Marr and T.Poggio, "Cooperative Computation of Stereo Disparity" Science, 194, page 283-287. 1976.

[19] D.Marr and T.Poggio, "A Computational Theory of Human Stereo Vision" Royal Science of London, B-204, page 301-328. 1979.

[20] W.E.Grimson. From Image to Surfaces : A Computational Study of the Human Early Visual System" Cambridge. Massachusetts. MIT Press, 1981.

[21] S.Maybank and O.Faugeras. A theory of Self-calibration of a moving camera. IJCV, 8:123-151,1992.

[22] G. Strang. Linear Algebra and Its Applications. Harcourt Brace Jovanovich, San Diego, 1988.

[23] Q-T. Luong and O.Faugeras. The fundamental matrix : theory, algorithms, and stability analysis. IJCV, 17(1):43-76, 1996.

[24] S.Maybank. Theory of Reconstruction from Image Motion. Springer-Verlag. Berlin, 1993.

[25] R.Hartley. Self-calibration from Muitiple views with a rotating camera. ECCV.LNCS 800/801 Springer-Verlag, 1994.

[26] M. Polleyfeys., Van Gool., T.Moons. Euclidean 3D reconstruction from stereo sequences with variable focal length. ACCV, Vol. II, pages 6-10, 1995.

[27] M.Polleyfeys., Van Gool. And A. Oosterlinck. The modulus constraint : a new constraint for self-calibration. ICPR., 1996.

[28] M.Polleyfeys., Van Gool., and M. Proesmans. Euclidean 3D reconstruction from image sequences with variable focal lengths. ECCV., Springer-Verlag, 1996.

[29] M. Pollefeys, R. Koch and L. Van Gool. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Pa-rameters, Proc.ICCV'98 (international Conference on Computer Vision), pp.90-95, Bombay, 1998. joint winner of the David Marr prize (best pa-per).

[30] J.G. Semple and G.T. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, 1952.

[31] B. Triggs, "The Absolute Quadric", Proc. 1997 Conference on Computer Vision and Pattern Recognition, IEEE Computer Soc. Press, pp. 609-614, 1997.

[32] J.Maybank, O.Faugeras. A theory of self-calibration of a moving camera. IJCV 8:2 p.123-151, 1992.

[33] Q.T.Luong, O.Faugeras. Camera calibration, scene motion and structure recovery from point correspondences and fundamental metrices. IJCV. 22(3):261-289,1997.

[34] R.Hartley and P.Sturm. Triangulation. CAIP, 190-197, 1995.

[35] M.Armstrong, A.Zisserman, R.Hartley. Self-calibration from image triplets. ECCV, 1996.

[36] R.Hartley. In defence of the 8 point algorithm. ICCV, 1995.

[37] R.Hartley. Self Calibration of stationary cameras. IJCV. 22(1):5-23, 1997.

[38] H.C.Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. Nature 293:10, 433-435, 1981.

[39] A.Heyden. Reconstruction and prediction from three images of uncalibrated cameras. SCIA, 1995.

[40] M.Pollefeys, L. Van Gool, A. Oosterlinck. The modulus constraint: a new constraint for self-calibration. ICPR 349-353, 1996.

[41] M.Pollefeys, R.Koch, L.Van Gool. Self-Calibration and Metric reconstruction in spite of varying and unknown internal camera parameters. ICCV. 90-95, 1998.

[42] M.Pollefeys. L.Van Gool. A stratified approach to metric self-calibration. Technical Report, 1997.

[43] J.Weng, T.S.Huang, and N. Ahuja. Motion and Structure from Image Sequences. Springer Series in Information Sciences. Springer-Verlag, Berlin Heidelberg, 1993.

[44] Z. Zhang. Motion and Structure of Four Points from One Motion of a Stereo Rig with Unknown Extrinsic Parameters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1222–1227, Dec 1995.

[45] Olivier Faugeras. Three-Dimensional Computer Vision. MIT Press, 1993.

[46] Olivier Faugeras, Stephane Laveau, Luc Robert. 3-D Reconstruction of Urban Scenes from Image Sequences. ISSPR, page 3-29, August,1997