Accelerated Multiple-View Reconstruction

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Frau Bing Liu
Master of Engineering, Northeastern University, Shenyang
aus Acheng (Heilongjiang), P.R.China

Mannheim, 2004

Dekan:	**Professor Dr. Jürgen Potthoff, Universität Mannheim**
Referent:	**Professor Dr. Reinhard Männer, Universität Mannheim**
Korreferent:	**Professor Dr. Bernd Jähne, Universität Heidelberg**

Tag der mündlichen Prüfung: 19. März 2004

Zusammenfassung

Diese Arbeit befasst sich mit dem Problem der schnellen 3D Rekonstruktion aus Bildfolgen. In der Forschung ist dies als *Multiple-View Rekonstruktion* bekannt. Es ist ein komplexer Prozess aus dem Bereich der *Computer Vision*. In dieser Arbeit werden drei Algorithmen entwickelt, welche drei grundlegende und wichtige Probleme innerhalb dieses Themas lösen.

Erstens wird in Kapitel 3 eine neue, einfache und lineare Methode vorgeschlagen um einen 3D-Punkt im Raum anhand seiner Projektionen in mehreren Ansichten zu rekonstruieren, deren Projektion Matrizen als genau bekannt angenommen werden. Diese Methode wird *1st-order MLE* genannt, da sie das ursprüngliche Problem umwandelt in Eines von linear begrenzter quadratischer Optimierung durch eine Approximation erster Ordnung zu den *epipolaren Randbedingungen*.

Kapitel 4 schlägt eine lineare, iterative Methode der kleinsten Quadrate für das Schätzen der Fundamentalmatrix zwischen zwei nicht kalibrierten Perspektiven vor. Auch hier wird das Problem durch eine Approximation erster Ordnung zu den epipolaren Randbedingungen in ein Problem der kleinsten Quadrate umgewandelt. Iterativ nähert sich die mittels der Least Square Methode minimierte Algebraische Kostenfunktion dem Geometrischen Fehler, wodurch man eine genauere Fundamentalmatrix erhält.

Die in Kapitel 5 dargestellten Techniken sind eine umfangreiche Anwendung der oben genannten 1st-order MLE Methode für das Problem *Bundle Adjustment*. Mit ihnen wird die Kostenfunktion des Bundle Adjustment teilweise linearisiert, wodurch der Minimierungsprozess beschleunigt wird.

Alle oben genannten Techniken bewahren den Fehler der gemessenen Bildpunkte und erlauben die Zuordnung einer individuellen Kovarianz zu jeder Bildmessung. Experimente zeigen, dass die Genauigkeit dieser Algorithmen mit der einer *maximum likelihood Schätzung* durch numerische Optimierung vergleichbar ist, jedoch bei wesentlich verringerten Berechnungskosten.

Aufbauend auf den oben eingeführten Techniken wird in Kapitel 6 eine zusätzliche Methode der *inkrementellen Multiple-View Rekonstruktion* entwickelt. Die höhere Leistungsfähigkeit der vorgeschlagenen Techniken erlaubt die Berücksichtigung von korrespondierenden Punkten aus vielen Ansichten, wodurch genauere Resultate erzielt werden. Bisherige Methoden betrachten nur korrespondierenden Punkten aus zwei oder drei Ansichten. Somit werden höhere Genauigkeit und Leistungsfähigkeit durch die vorgeschlagene Methode der inkrementellen Multiple-View Rekonstruktion erzielt.

Abstract

This dissertation deals with the problem of 3D reconstruction from image sequences in a more efficient manner. This technique is known as Multiple-View Reconstruction. It is a complex process in computer vision. Three techniques are developed in the dissertation, which solve respectively three fundamental problems within this topic.

Firstly, a new linear and non-iterative method to reconstruct a 3D-point in space from its projections in multiple views with known projection matrices is proposed in Chapter 3. This method is called *1st-order MLE*, since it converts the original reconstruction problem into one of linearly-constrained quadratic optimization through a first-order approximation to the epipolar constraints.

Chapter 4 proposes a linear iterative least-squares method for estimating the fundamental matrix between two un-calibrated perspective views. Like in chapter 3 the problem is converted into a least-squares problem by a first-order approximation to the epipolar constraints. The algebraic cost function of the least-squares is minimized iteratively to approach the geometric error, and a more accurate fundamental matrix is obtained accordingly.

The techniques presented in Chapter 5 are extensive applications of the above *1st-order MLE* method to the problem of *bundle adjustment*. With it the cost function of bundle adjustment is partly linearized, and thus the minimization process is accelerated.

All the above techniques preserve the error model of the measured image points, and allow the assignment of individual covariance to each image measurement. Experiments show that the accuracy of these algorithms is consistently comparable to that of a *maximum likelihood estimation* using numerical Newton-type optimization, however, at a much reduced computational cost.

Finally, based on the above techniques, an incremental multiple view reconstruction method is developed in Chapter 6. The higher efficiency of these techniques allows the incremental reconstruction method to take point-matches across multiple views into consideration, and thus more accurate results are achieved. This is different from previous approaches which consider only point matches across two or three views. Therefore both higher accuracy and efficiency can be achieved at the same time by the proposed multiple view reconstruction method.

Acknowledgements

I wish to give my sincere thanks to all the people who have supported me with this work. Especially, I would like thank

- Prof. Dr. Reinhard Männer, who gave substantial support and encouragement to my work during the last three and a half years, and is always ready to share his opinions and experiences, and gave me a lot of valuable advices,
- Gottlieb-Daimler-und-Karl-Benz Stiftung, for granting me two years' fellowship for my PhD work and giving me a great encouragement to start this work,
- my colleagues in ViPA group, Markus Schill, Thomas Ruf, Olaf Körnner, Nikolaj Nock, Clemens Wagner, Johannes Grimm, Nobert Hinckers, Marc Hennen, and Andreas Köpfle who have given me numerous helps not only with my work but also with my living since the first day when I came to Germany,
- Dennis Maier and Karsten Mühlmann both for their great helps with my work and for the sincere and valuable friendship,
- and my parents and Maoyuan for their enduring support.

Contents

Notation and Abbreviation

Notation

Scalars are denoted by italic lower-case letters, except the components of the world coordinates, which are small capital letters. Bold lower-case letters denote vectors, and bold capital letters denote matrices. A vector will by default refer to a column vector.

$\tilde{\mathbf{a}}$	homogeneous coordinates
$\mathbf{x} = (x, y)^\top$	inhomogeneous 2D-coordinates
$\tilde{\mathbf{x}} = (x, y, 1)^\top$	homogeneous 2D-coordinates
$\mathbf{X} = (\mathrm{X},\mathrm{Y},\mathrm{Z})^\top$	inhomogeneous 3D-coordinates
$\tilde{\mathbf{X}} = (\mathrm{X},\mathrm{Y},\mathrm{Z}, 1)^\top$	homogeneous 3D-coordinates
$\mathbf{P}$	the projective matrix
$\mathbf{K}$	the internal calibration matrix
f	the focal length in the world coordinates
f_x, f_y	the focal length in pixels
p_x, p_y	the principal point of a camera
s	the skew parameter of a camera
$\mathbf{R}$	a rotation matrix
$\mathbf{t}$	a translation vector
$\mathbf{C}$	projection center / camera center / optical center
$\mathbf{H}$	planar homography
$\mathbf{e}$	epipole
$\mathbf{F}$	the fundamental matrix
$\det(\mathbf{M})$	the determinant of square matrix $\mathbf{M}$
$\mathbf{M}^\top$	the transpose of matrix $\mathbf{M}$
$\mathbf{a}^\top$	the transpose of vector $\mathbf{a}$
$\mathrm{diag}(x_1, \cdots, x_n)$	a $n \times n$ diagonal matrix
$[\mathbf{x}]_\times$	the cross product matrix of a 3-vector $\mathbf{x}$
$\mathbf{0}_n$	a null column n-vector
$\mathbf{0}_{n\times m}$	a $n \times m$ null matrix
$\mathbf{I}_{n\times n}$	a $n \times n$ identity matrix
$\mathcal{R}^n$	n-dimensional Euclidean space
$\mathcal{P}^n$	n-dimensional projective space

Abbreviation

2D	Two Dimensional
3D	Three Dimensional
n-vector	n-dimensional vector
n-space	n-dimensional space
CCD	Charge Coupled Device
DLT	Direct Linear Transformation
DOF	Degree of Freedom
IEEE	Institute of Electrical and Electronic Engineers
LMS	Least-Mean Squares
LSM	Least-Squares Method
MLE	Maximum Likelihood Estimation
RANSAC	RANdom SAmple Consensus
RMS	Root Mean Square
SVD	Singular Value Decomposition

1 Introduction

1.1 Motivations

Over the past decade the interest in 3D models has dramatically increased. Computer-generated 3D models have been used in more and more applications. Although many tools are at hand to ease the generation of 3D models, it is still a time consuming and expensive process, and is hard to satisfy the increasing demand for more complex and realistic models. In many cases the models are copies of existing scenes or objects in the real world. Traditional solutions include stereo rigs, laser range scanners and other 3D digitizing devices. These devices require careful handling and complex calibration procedures, and are usually designed for a restricted depth range only. A flexible method of 3D-scene reconstruction is required.

Since the end of the last century researchers in computer vision have paid much attention to this problem, to automatize the 3D-model acquisition from real world. Their goal is focused on automatic and real-time extracting of a realistic 3D model by freely moving one or more cameras across a 3D scene, i.e. 3D-scene reconstruction from its 2D projections.

Human vision system could understand a three dimensional world naturally, only through its 2D projections. However, this seemingly effortless act of inferring 3D from 2D observations is in fact a non-trivial problem and is still far from being resolved scientifically.

Considerable efforts have been devoted to this topic in the recent years as seen from the number of publications and books [4] [15] [16] [28]. Pollefeys [53] gave an overview of the procedure for 3D modelling from multiple images in Fig. 1.1. The overall process may be stated in the following steps:

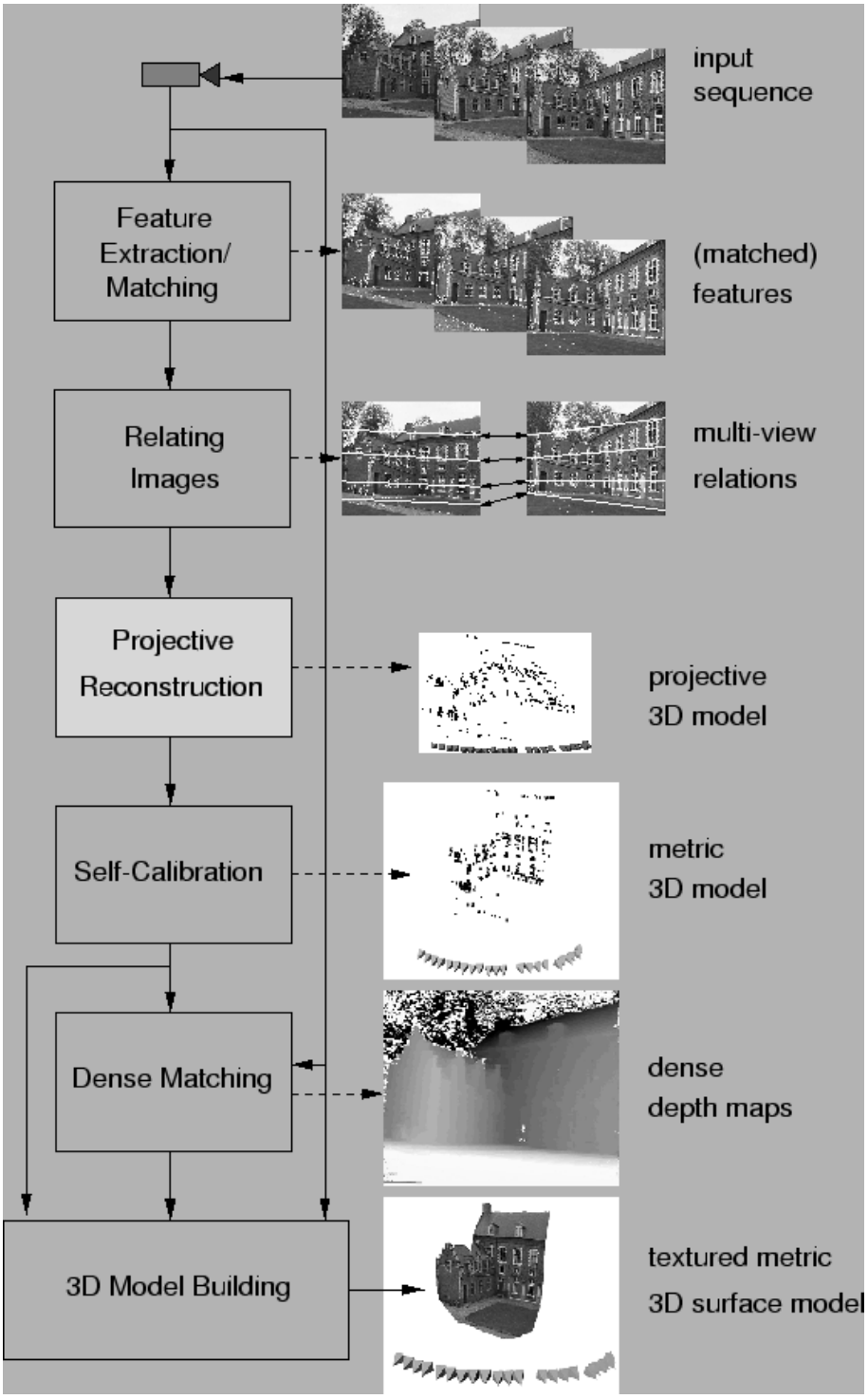


Figure 1.1: **Procedure for 3D modelling from images** (from [53]). *The highlighted step is the topic addressed in this dissertation.*

step 1 A sequence of images caught with one or more perspective cameras are the input to the modelling system.

step 2 Features that are sufficiently different from its neighborhoods are extracted from the images, e.g. the corner points [20], and matched between each pair of the successive images using similarity measurement based on sum-of-square-differences (SSD) [62] [67] or normalized cross-correlation (NCC) [39].

step 3 Since wrong correspondences are usually present, a robust algorithm is necessarily applied to filter out the outlying matches. Usually this process is done through establishing a reasonable relationship (2D homography / 2D projective transformation) between the consecutive views [77] [83] [82].

step 4 When *un-calibrated* cameras are used, (i.e. the intrinsic parameters of the cameras are unknown,) the structure of the scene can be determined up to an arbitrary projective transformation using image feature correspondences, e.g. points or lines. Accordingly, the procedure is called *projective reconstruction*. Many methods have been proposed to conduct projective reconstruction, as will be reviewed in detail in Chapter 2.

step 5 *Self-calibration* or *auto-calibration* is next conducted to restrict the ambiguity of the projective model to metric. Mostly self-calibration algorithms are concerned with unknown but constant intrinsic camera parameters, e.g. constant aspect ratio, skew or focal length [17] [21] [55] [54] [56] [29] [31] [72]. Many researchers also proposed specific self-calibration algorithms for restricted motions, such as pure translation [49] [2], pure rotations [23] or planar motion [1] [2]. Moreover, some self-calibration methods were proposed based on scene constraints [73] [14] [35].

When the cameras are *calibrated*, (i.e. the intrinsic parameters of the cameras are pre-calibrated,) the above two steps may be replaced by a direct *metric reconstruction*. That is, the structure of the scene and the motion of the cameras are determined up to an arbitrary scale factor, using the set of feature correspondences across the views. Approaches similar to those of projective reconstruction may be applied to metric reconstruction from calibrated images.

step 6 At this point enough information is available to go back to the images and search for correspondences for all the other image points, in order to get a dense depth estimate for the scene. The search is restricted to one dimension, since the line of sight corresponding to an image point is projected to a computable line in another image. The technique of *rectification* is usually used to simplify

the stereo matching and reduce the search to one row of the rectified images [5] [57].

From the correspondences, the distance/*depth* from the corresponding 3D points to the principal plane of the camera can be obtained through *triangulation* [27].

step 7 Finally, a textured 3D surface integrating all the results above can be generated, through approximating the depth map with a triangular wire frame [36] [76] [13].

As we have seen above, 3D-modelling from multiple images is a comprehensive task, and each step in the system is also a complex issue itself. The accuracy and efficiency of each step influence the whole system. This dissertation addresses the high-lighted step in Fig. 1.1, *projective reconstruction* for un-calibrated cameras, as well as the *metric reconstruction* with calibrated cameras. They together are known as *multiple-view reconstruction* in the field of Computer Vision.

1.2 Organization of the Dissertation

First, some background knowledge is introduced in **Chapter 2**, including the basic concepts in multiple-view geometry and some existing algorithms used in multiple-view reconstruction. It allows the interested readers to understand the material covered in the following.

Chapters 3, 4, 5 are the main contributions of this dissertation. They present three new techniques, dealing with three fundamental problems in multiple-view reconstruction.

Chapter 3 deals with 3D-point reconstruction from multiple views. A linear non-iterative algorithm is presented to reconstruct a 3D point directly from its projections in multiple views with known projection matrices. Experiments show that the linearization used in this algorithm does not reduce the accuracy of the reconstruction, and it is by far faster than other previous methods [43].

Chapter 4 proposes a linear and iterative method for estimating the fundamental matrix, which represents the epipolar geometry between two un-calibrated perspective images. It preserves the noise model of the observed image points, e.g. a Gaussian noise distribution. When noise in the measurement of image points is small, the accuracy of this method is comparable to that of non-linear Newton-type optimizers, but the computational cost is much reduced [41].

Chapter 5 discusses the problem of bundle adjustment, which refines the estimation of 3D structure and view parameters through minimizing the global reprojection error. In this chapter it is suggested to partially linearize the computation of the cost function first, and then with the help of the linearization two techniques of bundle adjustment can be accelerated at very little cost of accuracy, whether the cameras are calibrated or not. The two proposed methods for conducting bundle adjustment are not only tolerant of missing data, but also allow the assignment of individual covariance to each image measurement [42].

Afterwards the techniques proposed in the previous three chapters are integrated in **Chapter 6** to solve the problem of 3D-scene reconstruction from a sequence of images. An incremental-reconstruction technique is proposed. Because of the efficiency of the sub-procedures in the system, more information in the images is allowed to be taken into account in the computation, compared with previous approaches; and accordingly higher accuracy is achieved with the proposed technique of incremental reconstruction.

Finally, **Chapter 7** draws the conclusions for this dissertation.

2 Camera Geometry and Multiple-View Geometry

This chapter introduces and reviews some of the basic ideas and concepts in the area of multiple-view geometry.

The geometry of a perspective camera model is first introduced, as well as the algorithms to estimate the projection matrix of a perspective camera, given the coordinates of a set of world-to-image point correspondences. Then the properties of two or three camera views are described, and so is the possible 3D reconstruction from two or three camera images. Finally, the geometry of multiple views is introduced based on the previous knowledge.

Projective reconstruction for un-calibrated cameras and metric reconstruction for calibrated cameras are the main topics of this dissertation. The two kinds of reconstruction may be conducted in a similar way. At the end of this section, a detailed introduction to the literature of projective 3D-scene reconstruction is given.

2.1 Camera Geometry

A camera is a mapping from the 3D world to a 2D image. In this dissertation perspective camera models are used. A perspective camera model corresponds to an ideal pinhole camera. The geometric process for image formation in a pinhole camera has been nicely illustrated by Leon Battista Alberti (1404-1472). See Fig. 2.1. The process is completely determined by choosing a perspective projection center and a "retinal" plane (or image plane). The projection of a scene point is then obtained as the intersection of a line passing through this point and the projection center with the "retinal" plane. The 3D-2D projection can be represented by a 3×4 matrix which

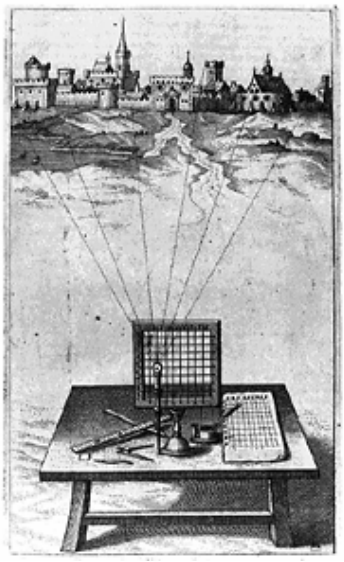


Figure 2.1: **Alberti's Grid - c.1450 (also known as "The Square Grid of the Renaissance").** *It offers a portable model for a perspective system, which represents three-dimensional objects on a two-dimensional surface.*

maps from homogeneous coordinates of a 3-space point to homogeneous coordinates of an imaged point on the image plane.

2.1.1 A Simple Model

Let the projection center be the origin of a Euclidean coordinate system, and the image plane be the plane $\mathrm{Z} = f$. See Fig. 2.2. The projection of a point in space with coordinates $\mathbf{X} = (\mathrm{X},\mathrm{Y},\mathrm{Z})^\top$ can be modelled as follows:

$$(\mathrm{X},\mathrm{Y},\mathrm{Z})^\top \mapsto (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z})^\top \tag{2.1}$$

The center of projection is called the *camera center* or *optical center*. The line from the camera center perpendicular to the image plane is called the *principal axis* or *principal ray* of the camera. The point where the principal axis meets the image plane is termed as the *principal point* of the camera.

Using homogeneous coordinates, Eq. 2.1 can be written in terms of matrix multi-

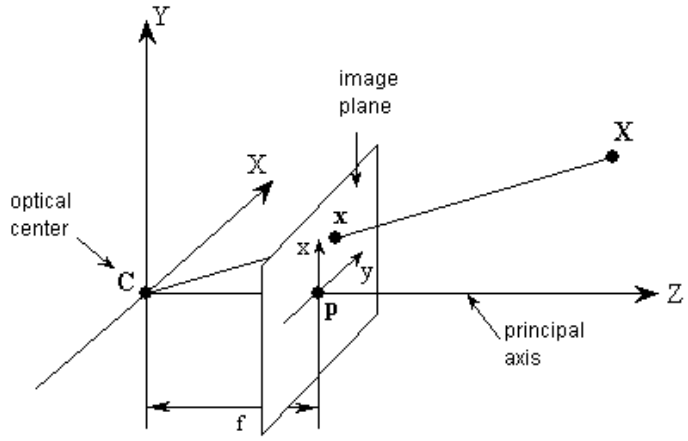


Figure 2.2: **Pinhole camera geometry.** **C** *is the camera center and* **p** *the principle point. The camera center is placed at the coordinate origin, and the image plane is placed at the plane* $\mathrm{Z} = f$.

plication as

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}. \tag{2.2}$$

We define $\tilde{\mathbf{X}} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^\top$, the homogeneous 4-vector of the world point, and $\tilde{\mathbf{x}} = (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z}, 1)^\top$, the homogeneous 3-vector of the image point. Then Eq. 2.2 can be written compactly as

$$\tilde{\mathbf{x}} \sim \mathbf{P}\tilde{\mathbf{X}} \tag{2.3}$$

where the 3×4 homogeneous matrix $\mathbf{P}$ is called the *camera matrix* or the *projection matrix* and

$$\mathbf{P} = \mathrm{diag}(f, f, 1)(\mathbf{I}_{3\times3}|\mathbf{0}_3). \tag{2.4}$$

$\mathbf{0}_3$ is a null 3-vector, and $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix. 3×3 diagonal matrix $\mathrm{diag}(f, f, 1)$ is the calibration matrix $\mathbf{K}$ of this camera model.

2.1.2 The Internal Calibration Matrix.

With an actual camera, the origin of coordinates in the image plane may not be at the principal point; the number of pixels per unit distance in both axial directions of the image plane may not be equal; and even the pixels could be non-rectangular.

The calibration matrix of any perspective camera can be written in a common form as

$$\mathbf{K} = \begin{bmatrix} f_x & s & x_0 \\ & f_y & y_0 \\ & & 1 \end{bmatrix} \tag{2.5}$$

where f_x and f_y represent the focal length of the camera in terms of the pixel dimension in x and y directions respectively, and $(x_0, y_0)^\top$ is the principal point in terms of pixels. The parameter s is referred to as the *skew* parameter, and $s = (\tan\alpha) f_y$ where α is the skew angle as indicated in Fig. 2.3. The ratio f_y/f_x is called the *aspect ratio* of the camera.

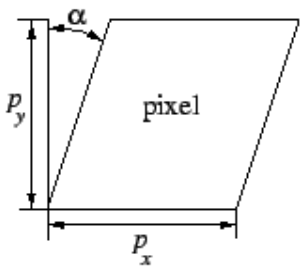


Figure 2.3: **A non-rectangular pixel.** *p_x and p_y are the width and the height of the pixel respectively, and α is the skew angle.*

In such a case, it can be derived that a 3-space point $\mathbf{X}_{\text{cam}}$ in the camera coordinate frame is projected to the image point

$$\tilde{\mathbf{x}} \sim \mathbf{K}[\mathbf{I}_{3\times3}|\mathbf{0}_3]\tilde{\mathbf{X}}_{\text{cam}} \tag{2.6}$$

where the *camera coordinate frame* refers to the Euclidean coordinate system with the camera center at the origin, and the principal axis of the camera straight down the Z-axis.

If the internal parameters of a camera are known, we say the camera is *calibrated*, or else it is *un-calibrated*.

2.1.3 Camera Motion

In general, points in space will be expressed in terms of the *world coordinate frame*. It does not necessarily coincide with the camera coordinate frame. The two frames are related via a rotation matrix and a translation vector. See Fig. 2.4. Let an inhomogeneous 3-vector $\mathbf{X}$ represent the coordinates of a point in the world coordinate

frame, and $\mathbf{X}_{\text{cam}}$ represent the same point in the camera coordinate frame, then we may write

$$\mathbf{X}_{\text{cam}} = \mathbf{R}(\mathbf{X} - \mathbf{C}), \tag{2.7}$$

where $\mathbf{C}$ represents the coordinates of the camera center in the world coordinate frame, and $\mathbf{R}$ is a 3×3 rotation matrix representing the orientation of the camera coordinate frame to the world coordinate frame. In homogeneous coordinates, Eq. 2.7 may be written as

$$\tilde{\mathbf{X}}_{\text{cam}} = \begin{pmatrix} \mathrm{X}_{\text{cam}} \\ \mathrm{Y}_{\text{cam}} \\ \mathrm{Z}_{\text{cam}} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \tilde{\mathbf{X}}. \tag{2.8}$$

2.1.4 A General Perspective Camera

Put Eq. 2.8 together with Eq. 2.6, and yield

$$\tilde{\mathbf{x}} \sim \mathbf{K}[\mathbf{R}| - \mathbf{RC}]\tilde{\mathbf{X}} = \mathbf{P}\tilde{\mathbf{X}}. \tag{2.9}$$

This is the general mapping given by a perspective camera. The 3×4 matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}| - \mathbf{RC}]$ is the general form of the *projective matrix* or *camera matrix*. It has rank 3 and 11 DOF: 5 for $\mathbf{K}$, 3 for $\mathbf{R}$, and 3 for $\mathbf{C}$. The parameters contained in $\mathbf{K}$ are called the *internal parameters* of the camera; the parameters of $\mathbf{R}$ and $\mathbf{C}$ which relate the camera orientation and position to the world coordinate system are called the *external parameters*.

For simplification, $\mathbf{t}$ is defined as $\mathbf{t} = -\mathbf{RC}$. Hence,

$$\mathbf{X}_{\text{cam}} = \mathbf{RX} + \mathbf{t} \tag{2.10}$$

and

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]. \tag{2.11}$$

It can be proved that the 3-vector $\mathbf{t}$ is the coordinate of the world frame center in the camera coordinate frame.

2.1.5 Computation of the Projective Matrix P

Given sufficiently many correspondences between 3D points $\mathbf{X}_i$ and their according images $\mathbf{x}_i$, the camera matrix $\mathbf{P}$ can be determined.

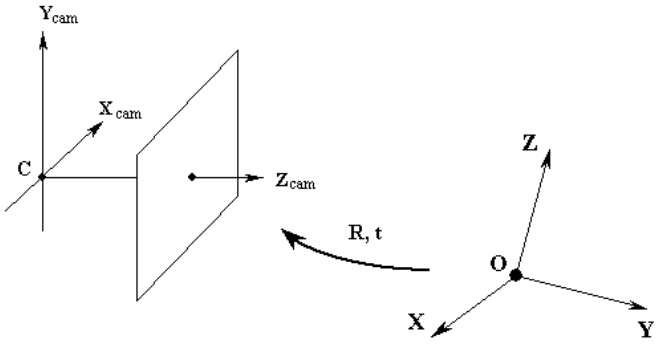


Figure 2.4: **Camera rotation and translation. R** *and* **t** *are the rotation matrix and the translation vector from the camera coordinate frame to the world coordinate frame.*

The Linear Algorithm

Given a set of corresponding points $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$ between 3D space points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$, it is required to find the camera matrix $\mathbf{P}$, namely a 3×4 matrix such that $\tilde{\mathbf{x}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$ for all i.

Let $\mathbf{x}_i = (x_i, y_i)^\top$. For each correspondence $\tilde{\mathbf{x}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$, and hence $\tilde{\mathbf{x}} \times \mathbf{P}\tilde{\mathbf{X}}_i = 0$ which is equivalent to

$$\begin{bmatrix} \mathbf{0}_4^\top & -\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ \mathbf{X}_i^\top & \mathbf{0}_4^\top & -x_i\mathbf{X}_i^\top \\ -y_i\mathbf{X}_i^\top & x_i\mathbf{X}_i^\top & \mathbf{0}_4^\top \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}_3 \tag{2.12}$$

where $\mathbf{P}^{i\top}$ is the i-th row of $\mathbf{P}$, i.e.

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{1\top} \\ \mathbf{P}^{2\top} \\ \mathbf{P}^{3\top} \end{bmatrix}.$$

The three equations in Eq. 2.12 are linearly dependent. Therefore, we may use only the first two equations.

$$\begin{bmatrix} \mathbf{0}_4^\top & -\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ \mathbf{X}_i^\top & \mathbf{0}_4^\top & -x_i\mathbf{X}_i^\top \end{bmatrix} \mathbf{p} = \mathbf{0}_2, \tag{2.13}$$

where $\mathbf{p} = (\mathbf{P}^{1\top}, \mathbf{P}^{2\top}, \mathbf{P}^{3\top})^\top$. For a set of n point correspondences, we obtain a $2n \times 12$ matrix $\mathbf{A}$ by stacking up the equations 2.13 for each correspondence, so that

the entries of matrix $\mathbf{P}$ may be computed through solving the set of linear equations $\mathbf{Ap} = \mathbf{0}_{2n}$.

One way to solve the equations $\mathbf{Ap} = \mathbf{0}_{2n}$ is using *Singular Value Decomposition* (*SVD*) to minimize $\| \mathbf{Ap} \|$ subject to $\| \mathbf{p} \| = 1$. $\mathbf{p}$ is computed as the unit singular vector corresponding to the smallest singular value of the matrix $\mathbf{A}$. The method to compute the singular value decomposition of a matrix can be found in book [19].

The Gold Standard Algorithm

Due to the noise in the measurement of point coordinates, there will not be an exact solution to the equations $\mathbf{Ap} = \mathbf{0}_{2n}$. Therefore, a solution to $\mathbf{P}$ may be obtained by minimizing the geometric error

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$$

where $\mathbf{x}_i$ is the measured image point and $\hat{\mathbf{x}}_i$ is the projection of the 3D point $\mathbf{X}_i$ under $\mathbf{P}$, i.e. $\tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$. $d(*, *)$ represents the distance between two points. When the world points $\mathbf{X}_i$ are known precisely and the measurement errors of the image points are Gaussian, the solution of

$$\min_{\mathbf{P}} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 \tag{2.14}$$

is the Maximum Likelihood Estimate (MLE) of $\mathbf{P}$, and $d(\mathbf{x}_i, \hat{\mathbf{x}}_i) = \| \mathbf{x}_i - \hat{\mathbf{x}}_i \|_{\Sigma_i}$ is called the *Mahalanobis distance* from $\mathbf{x}_i$ to $\hat{\mathbf{x}}_i$, where Σ_i is the covariance matrix for the measurement error of image points.

The complete Gold Standard algorithm for computing $\mathbf{P}$ is given in algorithm 2.1.

2.2 Two-View Geometry

Epipolar geometry is the intrinsic projective geometry between two perspective views. It is independent of scene structure, but depends on the cameras' internal parameters and their relative pose (position and orientation). There exists a 3×3-matrix F, termed as *fundamental matrix*, that encapsulates this intrinsic geometry.

In this section epipolar geometry is first introduced, and then the fundamental matrix between two views is derived. It will be shown that the projection matrices of two views can be retrieved from their fundamental matrix up to a projective transformation of 3-space. This result is the basis for projective reconstruction. The estimation of $\mathbf{F}$ from correspondences of imaged scene points is presented at the end of this section.

Algorithm 2.1 The Gold Standard algorithm for estimating $\mathbf{P}$ from a set of world-to-image point correspondences in the case that the world points are accurately known.

Objective

Given $n \geq 6$ world-to-image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood Estimate of the projection matrix $\mathbf{P}$ that minimizes $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$ where $\tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$.

Algorithm

1. **The linear algorithm.** Compute an initial estimate of $\mathbf{P}$ using a linear method:

   a) **Normalization:** Use a *similarity transformation*[a] $\mathbf{T}$ to normalize the image points $\mathbf{x}_i$, and a second similarity transformation $\mathbf{U}$ to normalize the space points $\mathbf{X}_i$. Suppose the homogeneous coordinates of the normalized image points are $\tilde{\mathbf{x}}_{iN} = \mathbf{T}\tilde{\mathbf{x}}_i$, and the normalized space points are $\tilde{\mathbf{X}}_{iN} = \mathbf{U}\tilde{\mathbf{X}}_i$.

   b) **Direct linear transformation method:** Form the $2n \times 12$ matrix $\mathbf{A}$ by stacking the equations (2.13) generated by the normalized correspondences $\{\mathbf{X}_{iN} \leftrightarrow \mathbf{x}_{iN}\}$. Write $\mathbf{p}_N$ for the vector containing the entries of the projection matrix $\mathbf{P}_N$ corresponding to $\{\mathbf{X}_{iN} \leftrightarrow \mathbf{x}_{iN}\}$. A solution of $\mathbf{A}\mathbf{p}_N = \mathbf{0}_{2n}$, subject to $\| \mathbf{p}_N \| = 1$, is obtained from the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value.

   c) **Denormalization.** The initial estimate of $\mathbf{P}$ for the original coordinates is obtained from $\mathbf{P}_N$ as $\mathbf{P} = \mathbf{T}^{-1}\mathbf{P}_N\mathbf{U}$.

2. **Minimize geometric error.** Use the linear estimate of $\mathbf{P}$ as a starting point to minimize the geometric error $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$ over $\mathbf{P}$, with an iterative optimization algorithm such as Levenberg-Marquardt optimizer, where $\tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$.

[a] The *similarity transformation* that normalizes a set of n-dimensional points is obtained through the following steps:

i. The points are translated so that their centroid is at the origin.

ii. The points are then scaled so that their average distance from the origin is equal to $\sqrt{n}$.

2.2.1 Epipolar Geometry

The geometric entities involved in the epipolar geometry between two perspective cameras are illustrated in Fig. 2.5. Each camera is indicated by its optical center $\mathbf{C}$ ($\mathbf{C}'$) and its image plane. The line joining the two camera centers is the *baseline*. The baseline intersects the image plane at the *epipole* $\mathbf{e}$ ($\mathbf{e}'$). Any plane containing the baseline is an *epipolar plane*, which intersects the two image planes at a pair of corresponding *epipolar lines*, e.g. $\mathbf{l}$ and $\mathbf{l}'$.

An image point $\mathbf{x}$ in the first view is back-projected to a ray in 3-space, which is defined by the first camera center $\mathbf{C}$ and the point $\mathbf{x}$. The ray is imaged as a line $\mathbf{l}'$ in the second view. The line $\mathbf{l}'$ is termed as the *epipolar line* corresponding to the point $\mathbf{x}$, and $\mathbf{x}$'s corresponding point $\mathbf{x}'$ in the second image always lies on the epipolar line $\mathbf{l}'$. The other way around, the point $\mathbf{x}$ lies on the epipolar line $\mathbf{l}$ corresponding to the point $\mathbf{x}'$.

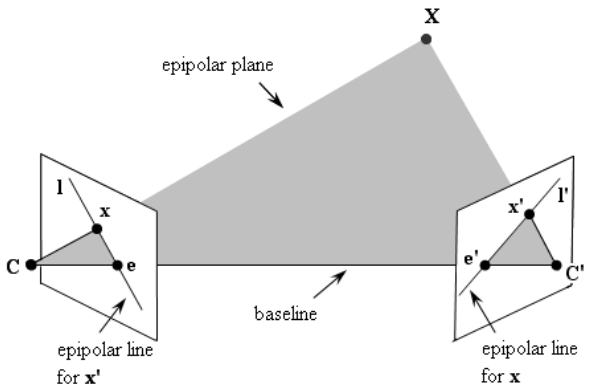


Figure 2.5: **The epipolar geometry.**

2.2.2 The Fundamental Matrix F

The fundamental matrix $\mathbf{F}$ is the algebraic representation of epipolar geometry. Given two images acquired by cameras with non-coincident centers, the fundamental matrix $\mathbf{F}$ from one image to another is a unique 3×3 rank-2 homogeneous matrix which satisfies

$$\tilde{\mathbf{x}}'^{\top} \mathbf{F} \tilde{\mathbf{x}} = 0 \tag{2.15}$$

for any pair of corresponding image points $\mathbf{x} \leftrightarrow \mathbf{x}'$.

For any point $\mathbf{x}$ in the first image, the homogeneous coordinates of its corresponding epipolar line in the second image is $\mathbf{l}' = \mathbf{F}\tilde{\mathbf{x}}$. Similarly, $\mathbf{l} = \mathbf{F}^\top\tilde{\mathbf{x}}'$ represents the epipolar line corresponding to $\mathbf{x}'$ in the second image.

Since the epipole $\mathbf{e}(\mathbf{e}')$ is on the epipolar line $\mathbf{l}(\mathbf{l}')$, it follows that $\mathbf{F}\tilde{\mathbf{e}} = \mathbf{0}_3$ and $\tilde{\mathbf{e}}'^\top\mathbf{F} = \mathbf{0}_3$.

3×3 matrix $\mathbf{F}$ has 9 elements, but only 7 DOF. The rank-2 constraint, i.e. $\det(\mathbf{F}) = 0$, removes one DOF; and the homogeneous definition removes another, for the common scaling is not significant.

Additionally, $\mathbf{F}$ can be computed from the two camera projective matrices $\mathbf{P}$, $\mathbf{P}'$:

$$\mathbf{F} = [\tilde{\mathbf{e}}']_\times \mathbf{P}'\mathbf{P}^+, \tag{2.16}$$

where $\mathbf{P}^+$ is the pseudo-inverse of $\mathbf{P}$, and $\tilde{\mathbf{e}}' = \mathbf{P}'\tilde{\mathbf{C}}$ with $\mathbf{P}\tilde{\mathbf{C}} = \mathbf{0}_3$. $[\tilde{\mathbf{e}}']_\times$ is the cross product matrix of 3-vector $\tilde{\mathbf{e}}'$.

2.2.3 Retrieving Camera Matrices from F

One of the most important properties of $\mathbf{F}$ is that it can be used to determine the camera matrices of the two views. Let us look at the following three results:

- $\mathbf{H}$ is a random 4×4 matrix representing a projective transformation of 3-space, then the fundamental matrix corresponding to a pair of camera matrices ($\mathbf{P}$, $\mathbf{P}'$) is the same as that corresponding to the pair of camera matrices ($\mathbf{PH}$, $\mathbf{P}'\mathbf{H}$).

- Let $\mathbf{F}$ be a fundamental matrix, and ($\mathbf{P}$, $\mathbf{P}'$) and ($\hat{\mathbf{P}}$, $\hat{\mathbf{P}}'$) be two pairs of camera matrices such that $\mathbf{F}$ is the fundamental matrix for either of the two pairs. Then there exists a non-singular 4×4 matrix $\mathbf{H}$ such that $\hat{\mathbf{P}} = \mathbf{PH}$ and $\hat{\mathbf{P}}' = \mathbf{P}'\mathbf{H}$.

- The fundamental matrix corresponding to a pair of camera matrices $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M}|\mathbf{m}]$ is equal to $[\mathbf{m}]_\times\mathbf{M}$.

Therefore, we know that, although a pair of camera matrices can uniquely determine a fundamental matrix, the converse is not true; and the pair of camera matrices can be determined at best up to a 3D projective transformation by a fundamental matrix. One may write down a particular solution for a pair of camera matrices corresponding to a fundamental matrix as

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}] \quad \textit{and} \quad \mathbf{P}' = [\mathbf{SF}|\mathbf{e}'],$$

where $\mathbf{e}'$ is the epipole such that $\mathbf{e}'^{\top}\mathbf{F} = \mathbf{0}_3$. It is suggested by Luong [45], a good choice for $\mathbf{S}$ is $\mathbf{S} = [\mathbf{e}']_{\times}$, since $\mathbf{e}'^{\top}\mathbf{e}' \neq 0$.

Given $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and the fundamental matrix $\mathbf{F}$, the general formula for the other camera matrix is

$$\mathbf{P}' = [[\mathbf{e}']_{\times} + \mathbf{e}'\mathbf{v}^{\top}|\lambda\mathbf{e}'] \tag{2.17}$$

where $\mathbf{v}$ is any 3-vector, and λ a non-zero scalar.

2.2.4 The Essential Matrix E

Consider a camera matrix decomposed as $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, and $\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}}$ is the projection of a space point $\mathbf{X}$. Let $\tilde{\mathbf{x}}_c = \mathbf{K}^{-1}\tilde{\mathbf{x}}$, then we have $\tilde{\mathbf{x}}_c = [\mathbf{R}|\mathbf{t}]\tilde{\mathbf{X}}$. $\mathbf{x}_c$ is termed as the *canonical image point* corresponding to image point $\mathbf{x}$ and 3×4 matrix $[\mathbf{R}|\mathbf{t}]$ is the *canonical camera matrix* or *canonical projection matrix* of the camera.

Now consider a pair of canonical camera matrices $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$. The fundamental matrix corresponding to the pair of canonical camera matrices is called the *essential matrix*, and it can be derived that

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K} \tag{2.18}$$

and

$$\tilde{\mathbf{x}}_c'^{\top}\mathbf{E}\tilde{\mathbf{x}}_c = 0 \tag{2.19}$$

where $\mathbf{x}_c$ and $\mathbf{x}'_c$ are the canonical image points corresponding to a pair of matched image points $\mathbf{x} \leftrightarrow \mathbf{x}'$. The essential matrix $\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$ has only 5 DOF: 3 for $\mathbf{R}$ and 3 for $\mathbf{t}$, but one removed for the overall scale ambiguity.

$\mathbf{E}$ has two equal non-zero singular values and one zero singular value. That means, $\mathbf{E}$ can be written in the form $\mathbf{E} \sim \mathbf{U}\mathrm{diag}(1, 1, 0)\mathbf{V}^{\top}$, where $\mathbf{U}$ and $\mathbf{V}$ are 3×3 orthogonal matrices. Given an essential matrix $\mathbf{E} = \mathbf{U}\mathrm{diag}(1, 1, 0)\mathbf{V}^{\top}$ and the first camera matrix $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, there are four possibilities for the second camera matrix, namely

$$\begin{array}{llll} \mathbf{P}' = [\mathbf{U}\mathbf{W}\mathbf{V}^{\top}| + \mathbf{u}_3] & \text{or} & \mathbf{P}' = [\mathbf{U}\mathbf{W}^{\top}\mathbf{V}^{\top}| + \mathbf{u}_3] & \text{or} \\ \mathbf{P}' = [\mathbf{U}\mathbf{W}\mathbf{V}^{\top}| - \mathbf{u}_3] & \text{or} & \mathbf{P}' = [\mathbf{U}\mathbf{W}^{\top}\mathbf{V}^{\top}| - \mathbf{u}_3] & \end{array}$$

where $\mathbf{u}_3$ is the last column of $\mathbf{U}$ and

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The four possible reconstructions from the essential matrix are illustrated in Fig. 2.6. In the image a camera is represented by its principle axis and its image plane.

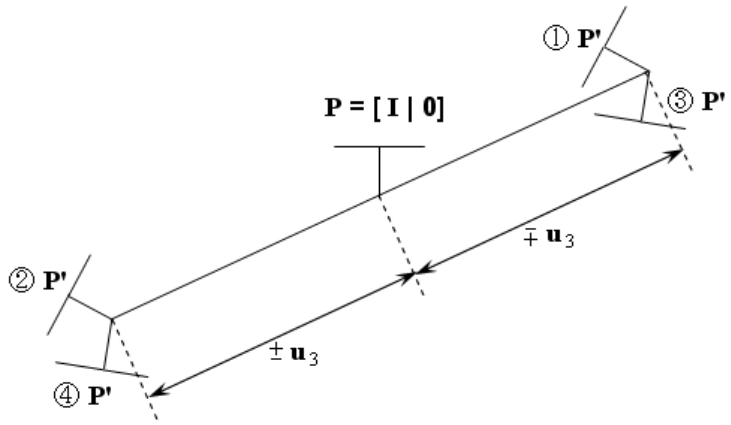


Figure 2.6: **Four possible reconstructions from E.** *The 3rd* $\mathbf{P}'$ *may be obtained by rotating the 1st* $\mathbf{P}'$ *by* 180° *around the baseline, and similarly, the 4th is obtained by rotating the 2nd by* 180°. *The 1st and the 2nd* $\mathbf{P}'$ *are parallel to the 3rd and the 4th* $\mathbf{P}'$ *respectively. Note that the principle axes of* $\mathbf{P}$ *and* $\mathbf{P}'$ *are not necessarily on the same plane.*

2.2.5 Computation of the Fundamental Matrix F

A fundamental matrix $\mathbf{F}$ is independent of scene structure, and can be computed from imaged point matches alone, without any a priori knowledge of the cameras' internal parameters or the relative pose between the two cameras.

The Linear Algorithm

Given a number of image point matches $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ in two images, we may compute the fundamental matrix $\mathbf{F}$ between the two images, namely a 3×3 rank-2 homogeneous matrix such that

$$\tilde{\mathbf{x}}_i'^{\top} \mathbf{F} \tilde{\mathbf{x}}_i = 0 \tag{2.20}$$

for all i.

Let $\mathbf{x}_i = (x_i, y_i)^{\top}$ and $\mathbf{x}'_i = (x'_i, y'_i)^{\top}$. We may rewrite Eq. 2.20 by

$$\begin{gathered} x'_i x_i f_{11} + x'_i y_i f_{12} + x_i f_{13} + y'_i x_i f_{21} + y'_i y_i f_{22} + y_i f_{23} + x_i f_{31} + y_i f_{32} + f_{33} = 0, \\ \text{i.e.} \quad (x'_i x_i, x'_i y_i, x'_i, y'_i x_i, y'_i y_i, y'_i, x_i, y_i, 1)\mathbf{f} = 0. \end{gathered} \tag{2.21}$$

where f_{ij} are the entries of $\mathbf{F}$, and $\mathbf{f}$ denotes the 9-vector made up of the entries of $\mathbf{F}$ in row-major order. From a set of n point matches, we can get n linear equations,

which may be written in the form of

$$\mathbf{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}_n. \tag{2.22}$$

Hence the entries of the matrix $\mathbf{F}$ can be computed up to scale by solving the set of linear equations $\mathbf{A}\mathbf{f} = \mathbf{0}_n$ using *SVD* [19].

When the rank of $\mathbf{A}$ is 8, the solution of $\mathbf{f}$ or $\mathbf{F}$ is unique. It means, at least 8 points are necessarily available for the estimation. Therefore, the algorithm is called the *8-point algorithm*.

Additionally, because $\mathbf{F}$ must be a rank-2 matrix, i.e. $\det(\mathbf{F}) = 0$, the obtained $\mathbf{F}$ from $\mathbf{A}\mathbf{f} = \mathbf{0}_n$ must be replaced by the closest singular matrix to $\mathbf{F}$ still using *SVD*. More specifically, let $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the *SVD* of $\mathbf{F}$, where $\mathbf{D} = diag(r, s, t)$ is a diagonal matrix subject to $r \geq s \geq t$. Then $\mathbf{F}' = \mathbf{U}diag(r, s, 0)\mathbf{V}^\top$ is the closest rank-2 matrix to $\mathbf{F}$, which minimizes the Frobenius norm of $\mathbf{F} - \mathbf{F}'$.

The Gold Standard Algorithm

Due to the noise in the identification of image points, there usually exists no exact solution to the equations $\mathbf{A}\mathbf{f} = \mathbf{0}_n$. Therefore, a solution to $\mathbf{F}$ may be obtained by minimizing the geometric error in the image

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \tag{2.23}$$

where $\mathbf{x}_i$ and $\mathbf{x}'_i$ are the measured correspondences in the two images, and $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ are the estimates of the "true" correspondences that satisfy $\tilde{\hat{\mathbf{x}}}_i'^\top \mathbf{F} \tilde{\hat{\mathbf{x}}}_i = 0$ for some rank-2 matrix $\mathbf{F}$. When the measurement error of the image points can be assumed Gaussian, the solution above is the MLE of $\mathbf{F}$.

The complete Gold Standard algorithm for computing $\mathbf{F}$ is given in algorithm 2.2.

2.3 Three-View Geometry

The trifocal tensor plays an analogous role in three views to that played by the fundamental matrix in two views. It is independent of scene structure, depending only on the projective relations between the three cameras. The camera matrices can be retrieved from the trifocal tensor up to a common projective transformation of 3-space, and the fundamental matrices for each view pair can be retrieved uniquely.

Algorithm 2.2 The Gold Standard algorithm for estimating $\mathbf{F}$ from image correspondences.

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the Maximum Likelihood Estimate of the rank-2 fundamental matrix $\mathbf{F}$ that minimizes $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$ subject to $\tilde{\hat{\mathbf{x}}}_i'^{\top} \mathbf{F} \tilde{\hat{\mathbf{x}}}_i = 0$.

Algorithm

1. **Linear initial estimate of F.**

   a) **Normalization:** Use two similarity transformations $\mathbf{T}$ and $\mathbf{T}'$ to normalize the measured image points in the two images respectively. Denote the homogeneous coordinates of the normalized point correspondences by $(\{\tilde{\mathbf{x}}_{iN} = \mathbf{T}\tilde{\mathbf{x}}_i) \leftrightarrow (\tilde{\mathbf{x}}'_{iN} = \mathbf{T}'\tilde{\mathbf{x}}'_i)\}$.

   b) **DLT:** Form the $n \times 9$ matrix $\mathbf{A}$ as in Eq. 2.22 generated from the normalized correspondences $\{\mathbf{x}_{iN} \leftrightarrow \mathbf{x}'_{iN}\}$. Write $\mathbf{f}_N$ for the 9-vector containing the entries of the 3×3 matrix $\mathbf{F}_N$. A solution to $\mathbf{A}\mathbf{f}_N = \mathbf{0}_n$, subject to $\| \mathbf{f}_N \| = 1$, is obtained from the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value.

   c) **Rank-2 constraint enforcement.** Replace $\mathbf{F}_N$ by $\mathbf{F}'_N$ such that $\det(\mathbf{F}'_N) = 0$ using *SVD* (see *p*21).

   d) **Denormalization.** The initial estimate of $\mathbf{F}$ for the original image correspondences is obtained from $\mathbf{F}'_N$ as $\mathbf{F} = \mathbf{T}^{-1}\mathbf{F}'_N\mathbf{T}'$.

2. **An initial estimate of the subsidiary variables $\mathbf{X}_i$.**

   a) Choose two projection matrices $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}' = [[\tilde{\mathbf{e}}']_\times \mathbf{F}|\tilde{\mathbf{e}}']$, where $\tilde{\mathbf{e}}'$ is computed as the left null-space of $\mathbf{F}$ using *SVD*.

   b) Reconstruct 3D point $\mathbf{X}_i$ from the correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ and the pair of projection matrices $\mathbf{P}$ and $\mathbf{P}'$ using the triangulation method as described in Sect. 2.4.1.

3. **Minimize the geometric error.** Using the above estimate $\mathbf{P}'$ and $\mathbf{X}_i$ as a starting point minimize the geometric error $\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$ with an iterative algorithm such as Levenberg-Marquardt, over $3n + 12$ variables: $3n$ for the n 3D points $\mathbf{X}_i$ and 12 for the camera matrix $\mathbf{P}' = [\mathbf{M}|\mathbf{t}]$, and $\tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$, $\tilde{\hat{\mathbf{x}}}'_i \sim \mathbf{P}'\tilde{\mathbf{X}}_i$. The final MLE of the fundamental matrix is then $\mathbf{F} = [\mathbf{t}]_\times \mathbf{M}$.

This section begins with an introduction to the geometric and the algebraic properties of the trifocal tensor; then it is shown how the tensor represents the relations between image correspondences for points and lines and how the camera matrices and the fundamental matrices are retrieved from the tensor. The computation of the trifocal tensor from point correspondences over three-views is described at the end of this section.

2.3.1 The Trifocal Tensor

Similar to what was derived for two views, there are multi-linear relationships between the image correspondences for points and lines in three images [65]. These multi-linear relationships between points [61] and lines [22] or any combination thereof [24] can be represented by the *trifocal tensor* [68] [69].

Let the projection matrices for three cameras be

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}], \quad \mathbf{P}' = [\mathbf{A}|\mathbf{a_4}], \quad \mathbf{P}'' = [\mathbf{B}|\mathbf{b_4}],$$

where $\mathbf{A}$ and $\mathbf{B}$ are 3×3 matrices, and the vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ are the i-th columns of the respective camera matrices for $i = 1, \cdots, 4$. Then from the incidence relation between a set of corresponding lines $\mathbf{l} = (l_1, l_2, l_3)^\top \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ in the three views (see Fig. 2.7), we can derive the following equations

$$l_i = \mathbf{l}'^\top(\mathbf{a}_i\mathbf{b}_4^\top)\mathbf{l}'' - \mathbf{l}'^\top(\mathbf{a}_4\mathbf{b}_i^\top)\mathbf{l}''.$$

Therefore, a set of rank-2 3×3 matrices

$$\mathbf{T}_i = \mathbf{a}_i\mathbf{b}_4^\top - \mathbf{a}_4\mathbf{b}_i^\top \qquad \text{for} \quad i = 1, 2, 3 \tag{2.24}$$

are introduced, and the line incidence relation above can be written as

$$l_i = \mathbf{l}'^\top\mathbf{T}_i\mathbf{l}''.$$

The set of matrices $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$ constitutes a $3 \times 3 \times 3$ tensor, i.e. the *trifocal tensor* $\mathcal{T}$, also written as $[\mathcal{T}_{ijk}]$. The tensor contains 27 parameters, but only 18 of these are independent due to additional nonlinear constraints.

With the trifocal tensor $\mathcal{T}$, the incidence relations between lines and points in three views are summarized as follows:

(i) Line-line-line correspondence ($\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$)

$$\mathbf{l}'^\top[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{l}'' \sim \mathbf{l}^\top \qquad \text{or} \qquad (\mathbf{l}'^\top[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{l}'')[\mathbf{l}]_\times = \mathbf{0}_3^\top, \tag{2.25}$$

where $\mathbf{l}'^\top[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{l}''$ represents the vector $(\mathbf{l}'^\top\mathbf{T}_1\mathbf{l}'', \mathbf{l}'^\top\mathbf{T}_2\mathbf{l}'', \mathbf{l}'^\top\mathbf{T}_3\mathbf{l}'')$.

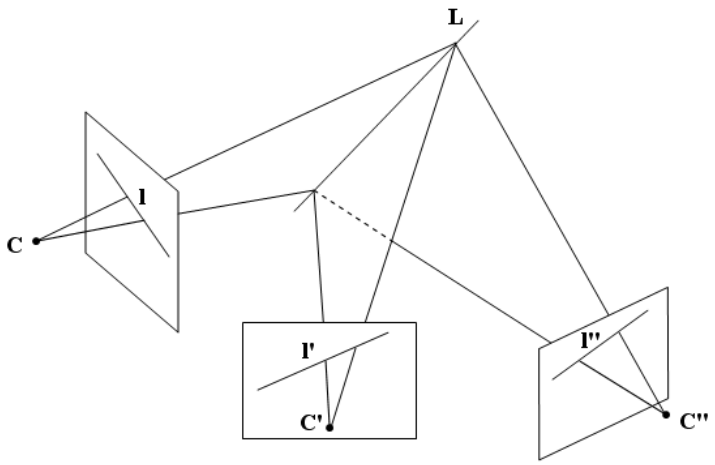


Figure 2.7: **Line correspondences in three views.** *A line* $\mathbf{L}$ *in 3-space is imaged as the corresponding triplet* $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ *in three views; and conversely, the back-projected planes from the three corresponding lines intersect in a single line in 3-space.*

(ii) Point-line-line correspondence ($\tilde{\mathbf{x}} = (x_1, x_2, x_3)^\top \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$)

$$\mathbf{l}'^\top (\sum_{i=1}^{3} x_i \mathbf{T}_i) \mathbf{l}'' = 0. \tag{2.26}$$

(iii) Point-point-line correspondence ($\tilde{\mathbf{x}} \leftrightarrow \tilde{\mathbf{x}}' \leftrightarrow \mathbf{l}''$)

$$\tilde{\mathbf{x}}' \sim (\sum_{i=1}^{3} x_i \mathbf{T}_i) \mathbf{l}'' \qquad \text{or} \qquad [\tilde{\mathbf{x}}']_\times (\sum_{i=1}^{3} x_i \mathbf{T}_i) \mathbf{l}'') = \mathbf{0}_3. \tag{2.27}$$

(iv) Point-line-point correspondence ($\tilde{\mathbf{x}} \leftrightarrow \mathbf{l}' \leftrightarrow \tilde{\mathbf{x}}''$)

$$\mathbf{l}'^\top (\sum_{i=1}^{3} x_i \mathbf{T}_i) \sim \tilde{\mathbf{x}}'' \qquad \text{or} \qquad (\sum_{i=1}^{3} x_i \mathbf{T}_i) \mathbf{l}'') [\tilde{\mathbf{x}}'']_\times = \mathbf{0}_3^\top. \tag{2.28}$$

(v) Point-point-point correspondence ($\tilde{\mathbf{x}} \leftrightarrow \tilde{\mathbf{x}}' \leftrightarrow \tilde{\mathbf{x}}''$)

$$[\tilde{\mathbf{x}}']_\times (\sum_{i=1}^{3} x_i \mathbf{T}_i) \mathbf{l}'') [\tilde{\mathbf{x}}'']_\times = \mathbf{0}_{3\times 3}. \tag{2.29}$$

Given the trifocal tensor $\mathcal{T}$, we can also compute the epipolar lines, and retrieve the epipoles, the fundamental matrices and the projection matrices:

- If $\tilde{\mathbf{x}} = (x_1, x_2, x_3)^\top$ is a point in the first image, and $\mathbf{l}'$ and $\mathbf{l}''$ are the corresponding epipolar lines in the second and third images respectively, then

$$\mathbf{l}'^\top(\sum_{i=1}^{3} x_i \mathbf{T}_i) = \mathbf{0}_3^\top \qquad \text{and} \qquad (\sum_{i=1}^{3} x_i \mathbf{T}_i)\mathbf{l}'' = \mathbf{0}_3^\top,$$

  i.e. the epipolar lines $\mathbf{l}'$ and $\mathbf{l}''$ are the left and right null-vectors of the matrix of $\sum_i x_i \mathbf{T}_i$, respectively.

- The epipole $\tilde{\mathbf{e}}'$ in the second image with respect to the first image is the common intersection of the left null-vectors of the matrices $\mathbf{T}_i$, $i = 1, 2, 3$. Similarly the epipole $\tilde{\mathbf{e}}''$ in the third image with respect to the first is the the common intersection of the right null-vectors of the $\mathbf{T}_i$.

- The fundamental matrices $\mathbf{F}_{21}$ and $\mathbf{F}_{31}$ are computed as

$$\mathbf{F}_{21} = [\tilde{\mathbf{e}}']_\times [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\tilde{\mathbf{e}}'' \qquad \text{and} \qquad \mathbf{F}_{31} = [\tilde{\mathbf{e}}'']_\times [\mathbf{T}_1^\top, \mathbf{T}_2^\top, \mathbf{T}_3^\top]\tilde{\mathbf{e}}'.$$

- With $\mathcal{T}$ and $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{P}'$ and $\mathbf{P}''$ may be retrieved by

$$\mathbf{P}' = [[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\tilde{\mathbf{e}}''|\tilde{\mathbf{e}}'] \quad \text{and} \quad \mathbf{P}'' = [(\tilde{\mathbf{e}}''\tilde{\mathbf{e}}''^\top - \mathbf{I})[\mathbf{T}_1^\top, \mathbf{T}_2^\top, \mathbf{T}_3^\top]\tilde{\mathbf{e}}'|\tilde{\mathbf{e}}''].$$

2.3.2 Computation of the Trifocal Tensor $\mathcal{T}$

As with the fundamental matrix, the trifocal tensor is independent of scene structure, and invariant to 3D projection transformation. It can be computed from image correspondences over three views. Then the cameras and 3D scene can be reconstructed up to a projective transformation of 3-space.

The Linear Algorithm

Given several point or line correspondences between three images, a set of the linear equations involving the trifocal tensor may be generated according to the incidence relations as described in Sect. 2.3.1. All of these equations are linear in the entries of the trifocal tensor $\mathcal{T}$, and therefore may be written in the form of $\mathbf{At} = \mathbf{0}$, where $\mathbf{t}$ is the 27-vector made up of the entries of $\mathcal{T}$. As with the computation of $\mathbf{P}$ in Sect. 2.1.5 and $\mathbf{F}$ in Sect. 2.2.5, the entries of $\mathcal{T}$ may be solved through minimizing $\|\mathbf{At}\|$ using *SVD*.

But not all these equations are necessary to be used. For instance in the case of a point-point-point correspondence, there are a total of 9 equations from Eq. 2.29, but only 4 of them are linearly independent and may be obtained by choosing any

Algorithm 2.3 The normalized linear algorithm for computation of $\mathcal{T}$.

Objective

Given $n \geq 7$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$, compute the trifocal tensor.

Algorithm

1. **Normalization:** Use three similarity transformations $\mathbf{U}$, $\mathbf{U}'$, $\mathbf{U}''$ to normalize the measured image points in the three images respectively. Denote the homogeneous coordinates of the normalized point correspondences by $(\{\tilde{\mathbf{x}}_{iN} = \mathbf{U}\tilde{\mathbf{x}}_i) \leftrightarrow (\tilde{\mathbf{x}}'_{iN} = \mathbf{U}'\tilde{\mathbf{x}}'_i) \leftrightarrow (\tilde{\mathbf{x}}''_{iN} = \mathbf{U}''\tilde{\mathbf{x}}''_i)\}$.

2. **DLT:** Generate the set of linear equations in the form $\mathbf{A}\mathbf{t}_N = \mathbf{0}_{4n}$ according to the independent equations in Eq. 2.29, using the normalized correspondences $\{\mathbf{x}_{iN} \leftrightarrow \mathbf{x}'_{iN} \leftrightarrow \mathbf{x}''_{iN}\}$, where $\mathbf{A}$ is a $4n \times 9$ matrix and $\mathbf{t}_N$ is the 27-vector representing the 27 entries of the trifocal tensor. Solve $\mathbf{t}_N$ through minimizing $\|\mathbf{A}\mathbf{t}_N\|$ using *SVD*.

3. **Enforcement of the trifocal tensor constraints** as described in the linear algorithm (*p*28).

4. **Denormalization.** Compose the trifocal tensor $\{\mathbf{T}_{1N}, \mathbf{T}_{2N}, \mathbf{T}_{3N}\}$ with $\mathbf{t}_N$.

   a) Define 3×3 matrices $\mathbf{T}'_{kN} = \mathbf{U}'^{-1}\mathbf{T}_{kN}\mathbf{U}''^{-\top}$ for $k = 1, 2, 3$, and $\mathbf{T}_{kN}^{ij\,\prime}$ the entry of $\mathbf{T}'_{kN}$ at the i-th row and j-th column.

   b) Let $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$ be the initial estimate of the trifocal tensor for the original image correspondences, and $\mathbf{T}_k^{ij}$ represent the entry of matrix $\mathbf{T}_k$ at the i-th row and j-th column. Then the entries of $\mathcal{T}$ may be computed as follows:

$$\begin{pmatrix} \mathbf{T}_1^{ij} \\ \mathbf{T}_2^{ij} \\ \mathbf{T}_3^{ij} \end{pmatrix} = \mathbf{U}^\top \begin{pmatrix} \mathbf{T}_{1N}^{ij\,\prime} \\ \mathbf{T}_{2N}^{ij\,\prime} \\ \mathbf{T}_{3N}^{ij\,\prime} \end{pmatrix} \qquad \text{for } i = 1, 2, 3 \text{ and } j = 1, 2, 3.$$

Algorithm 2.4 The Gold Standard algorithm for estimating $\mathcal{T}$.

Objective

Given $n \geq 7$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$, determine the Maximum Likelihood Estimate of the trifocal tensor that minimizes

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 + d(\mathbf{x}''_i, \hat{\mathbf{x}}''_i)^2.$$

The set of point correspondences $\{\hat{\mathbf{x}}_i,\ \hat{\mathbf{x}}'_i$ and $\hat{\mathbf{x}}''_i\}$ are the estimated "true" correspondences that exactly satisfy the trifocal constraints in Eq. 2.29 with respect to the estimated trifocal tensor.

Algorithm

1. Initial estimate of $\mathcal{T}$ using the linear algorithm 2.3.
2. Initial estimate of the subsidiary variables $\mathbf{X}_i$.
   a) Let $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, and retrieve the camera matrices $\mathbf{P}'$ and $\mathbf{P}''$ from $\mathcal{T}$.
   b) Reconstruct 3D point $\mathbf{X}_i$ from the correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$ and $\mathbf{P}$, $\mathbf{P}'$ and $\mathbf{P}''$ using the triangulation method as described in Sect. 2.4.1.
3. Using the above retrieved $\mathbf{P}'$, $\mathbf{P}''$ and estimated $\mathbf{X}_i$ as a starting point minimize the geometric error

   $$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 + d(\mathbf{x}''_i, \hat{\mathbf{x}}''_i)^2.$$

   with an iterative algorithm such as Levenberg-Marquardt, over $3n + 24$ variables: $3n$ for the n 3D point $\mathbf{X}_i$and 24 for the elements of the two camera matrices $\mathbf{P}'$ and $\mathbf{P}''$, where $\hat{\tilde{\mathbf{x}}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$, $\hat{\tilde{\mathbf{x}}}'_i \sim \mathbf{P}'\tilde{\mathbf{X}}_i$ and $\hat{\tilde{\mathbf{x}}}''_i \sim \mathbf{P}''\tilde{\mathbf{X}}_i$. The final MLE of the trifocal tensor is then obtained as given in Eq. 2.24 with the new-estimated $\mathbf{P}'$ and $\mathbf{P}''$.

two equations from any two columns or rows. Similarly, two of the three equations in Eq. 2.27, Eq. 2.28 or Eq. 2.25 are linearly independent.

The $\mathcal{T}$ that minimizes $\|\mathbf{At}\|$ does not consider the constraints on $\mathcal{T}$ discussed in Sect. 2.3.1. These constraints may be enforced through the following steps:

(i) **Retrieve the epipoles $\tilde{\mathbf{e}}'$, $\tilde{\mathbf{e}}''$.**

a) For each $i = 1, 2, 3$ find the unit vector $\mathbf{v}_i$ that minimizes $\|\mathbf{T}_i\mathbf{v}_i\|$. Form the matrix $\mathbf{V}$, the i-th row of which is $\mathbf{v}_i^\top$.

b) Compute $\tilde{\mathbf{e}}''$ (the epipole in the third image with respect to the first image) as the unit vector that minimizes $\|\mathbf{V}\tilde{\mathbf{e}}''\|$.

Similarly, the epipole $\tilde{\mathbf{e}}'$ in the second image with respect to the first image is computed using $\mathbf{T}_i^\top$ instead of $\mathbf{T}_i$.

(ii) **Retrieve the camera matrices $\mathbf{P}'$, $\mathbf{P}''$ with $\mathbf{P} = (\mathbf{I}|\mathbf{0})$.** Given $\mathbf{P} = (\mathbf{I}|\mathbf{0})$, it can be derived that $\tilde{\mathbf{e}}'$ equals the last column of $\mathbf{P}'$ (up to scale), and so does $\tilde{\mathbf{e}}''$ for $\mathbf{P}''$. Then from Eq. 2.24, it may be seen that once the epipoles $\tilde{\mathbf{e}}' = \mathbf{a}_4$ and $\tilde{\mathbf{e}}'' = \mathbf{b}_4$ are known, the entries of $\mathcal{T}$ may be represented linearly in terms of the remaining entries of $\mathbf{P}'$ and $\mathbf{P}''$. Let a 27×18 matrix $\mathbf{E}$ express the known linear relationship between the 27-vector $\mathbf{t}$ (the entries of $\mathcal{T}$) and the 18-vector $\mathbf{q}$ (the remaining entries of $\mathbf{P}'$ and $\mathbf{P}''$), i.e. $\mathbf{t} = \mathbf{Eq}$. Hence, $\mathbf{q}$ may be solved through $\min_{\mathbf{q}} \|\mathbf{At}\| = \|\mathbf{AEq}\|$ using *SVD* with known $\mathbf{A}$ and $\mathbf{E}$, and $\mathbf{t} = \mathbf{Eq}$ follows. The solution $\mathbf{t} = \mathbf{Eq}$ represents the trifocal tensor subject to all the constraints upon a valid trifocal tensor.

The normalized linear algorithm for computing $\mathcal{T}$ is summarized in algorithm 2.3.

The Gold Standard Algorithm

As with the fundamental matrix, the best estimation of $\mathcal{T}$ may be obtained by the Maximum Likelihood Estimation. Given a set of point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i' \leftrightarrow \mathbf{x}_i''\}$ in three views, the cost function to be minimized is

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}_i', \hat{\mathbf{x}}_i')^2 + d(\mathbf{x}_i'', \hat{\mathbf{x}}_i'')^2 \tag{2.30}$$

where $\hat{\mathbf{x}}_i$, $\hat{\mathbf{x}}_i'$ and $\hat{\mathbf{x}}_i''$ are the estimated "true" correspondences that satisfy the trifocal constraints in Eq. 2.29 exactly with respect to the estimated trifocal tensor. The complete algorithm for estimating $\mathcal{T}$ is given in algorithm 2.4.

2.4 Multiple-View Reconstruction

This section will show that it is possible to reconstruct a 3D scene from two or more images captured from different positions by one or more cameras. The most commonly-used approaches for scene reconstruction are feature-based. Features refer to points, lines, or other primitives in the images.

In this section, some basic algorithms are reviewed to reconstruct 3D points and lines from their projections in two or more camera images with known projection matrices. Then the existing techniques of projective reconstruction from a set of un-calibrated camera images are described.

2.4.1 3D Point and Line Reconstruction

In this section the reconstruction of 3D points and lines is discussed, given their images in $N(\geq 2)$ views and the projection matrices of those views. It is assumed that noise occurs only in the measured image coordinates, but not the camera matrices.

Point Reconstruction

Suppose $\mathbf{x}_i = (u_i, v_i)$ be the image point in view i, for $i = 1, \ldots, N$, and their back-projected rays intersect at a single 3D-point $\mathbf{X}$ in space. Then we have

$$\tilde{\mathbf{x}}_i \sim \mathbf{P}_i\tilde{\mathbf{X}} \quad \text{or} \quad w_i\tilde{\mathbf{x}}_i = \mathbf{P}_i\tilde{\mathbf{X}} \tag{2.31}$$

where w_i is an unknown scale factor and $\mathbf{P}_i$ is the projection matrix of view i and supposed to be known a priori.

Denote the j-th row of $\mathbf{P}_i$ by $\mathbf{P}_i^{j\top}$. Then Eq. 2.31 may be rewritten as

$$w_i u_i = \mathbf{P}_i^{1\top}\tilde{\mathbf{X}}, \quad w_i v_i = \mathbf{P}_i^{2\top}\tilde{\mathbf{X}}, \quad w_i = \mathbf{P}_i^{3\top}\tilde{\mathbf{X}}$$

Eliminate w_i by substituting the third equation into the first two, which yields the following two independent linear equations

$$\begin{matrix}(u_i\mathbf{P}_i^{3\top} - \mathbf{P}_i^{1\top})\tilde{\mathbf{X}} = 0 \\ (v_i\mathbf{P}_i^{3\top} - \mathbf{P}_i^{2\top})\tilde{\mathbf{X}} = 0\end{matrix} . \tag{2.32}$$

Then for N views, there are $2N$ linear equations in terms of $\tilde{\mathbf{X}}$. They may be written in the form of

$$\mathbf{A}\tilde{\mathbf{X}} = \mathbf{0}_{2N}, \tag{2.33}$$

where $\mathbf{A}$ is a $2N \times 4$-matrix and

$$\mathbf{A} = \begin{bmatrix} \cdots \\ u_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{1\top} \\ v_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{2\top} \\ \cdots \end{bmatrix}. \tag{2.34}$$

As before, $\mathbf{X}$ can be computed by solving the set of linear equations $\mathbf{A}\tilde{\mathbf{X}} = \mathbf{0}_{2N}$ using *SVD*.

An alternative method to solve $\mathbf{A}\tilde{\mathbf{X}} = \mathbf{0}_{2N}$ is using the so-called *normal equations* since the last element of $\tilde{\mathbf{X}}$ is known to be 1. That is to minimize $\| \mathbf{A}'\mathbf{X} - \mathbf{b} \|$, where $\mathbf{A}'$ is the first three columns of $\mathbf{A}$ and $\mathbf{b}$ is its last column, and hence $\mathbf{X}$ may be solved as

$$\mathbf{X} = (\mathbf{A}'^{\top}\mathbf{A}')^{-1}\mathbf{A}'^{\top}\mathbf{b}. \tag{2.35}$$

When not all the optical centers of the views are collinear with the 3D-point $\mathbf{X}$, the 3×3 matrix $\mathbf{A}'^{\top}\mathbf{A}'$ is invertible. Otherwise, the 3D-point is un-reconstructable.

The above linear method for reconstructing a point in 3-space is called *Linear Triangulation Method* or Least-Squares Method, in which an algebraic error $\| \mathbf{A}\tilde{\mathbf{X}} \|$ or $\| \mathbf{A}'\tilde{\mathbf{X}} - \mathbf{b} \|$ is minimized.

The above presents a simple linear solution to 3D point reconstruction, in which the algebraic distance (as defined by [26]) between measured image points and estimated points is minimized. The geometric error for a reconstructed 3D point $\mathbf{X}$ is computed as follows

$$\sum_{i=1}^{N} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 \quad \text{subject to} \quad \tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}_i\tilde{\mathbf{X}}, \tag{2.36}$$

where $\mathbf{x}_i$ is the measured image point in view i, and $\hat{\mathbf{x}}_i$ is its estimated "true" image point. When a Gaussian error distribution can be assumed for the measurement of the image points, the points $\hat{\mathbf{x}}_i$ for $i = 1, \cdots, N$ that minimize the cost function in Eq. 2.36 are Maximum Likelihood Estimates (MLE) for the true image point correspondences. The minimization may be carried out with the a numerical optimization method such as Levenberg-Marquardt [50]. Once $\hat{\mathbf{x}}_i$ for $i = 1, \cdots, N$ are obtained, the MLE of the 3D point $\hat{\mathbf{X}}$ may be computed by any triangulation method.

Line Reconstruction

Suppose a line in 3-space is projected to 2D lines in N views. Given the projection matrices $\mathbf{P}_i$ of those views and the imaged lines $\mathbf{l}_i$ in the views, the line in 3-space

can be reconstructed as the intersection line of the back-projected planes from the N imaged lines.

With the homogeneous coordinates, the planes defined by the back-projection of the imaged lines are $\pi_i = \mathbf{P}_i^\top \mathbf{l}_i$, for $i = 1, \cdots, N$. Without noise, these planes should intersect at a single 3D line in space, and hence the $4 \times N$ matrix

$$\mathbf{A} = \begin{bmatrix} \pi_1 & \pi_1 & \cdots & \pi_N \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^\top l_1 & \mathbf{P}_2^\top l_2 & \cdots & \mathbf{P}_N^\top l_N \end{bmatrix}$$

has rank 2. The line in space may be parameterized by any two of the N planes defined by the image lines, as long as the two planes are distinct.

However, in the presence of noise, the rank of the $4 \times N$ matrix $\mathbf{A}$ is generally greater than 2 when $N > 2$. To reconstruct the line, we can also use the technique of *SVD* [28] [40]. Let $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the *SVD* of $\mathbf{A}$. The two columns of $\mathbf{U}$ corresponding to the two largest singular values span the best rank-2 approximation to $\mathbf{A}$ and define the best intersection line of the planes.

2.4.2 Projective Reconstruction from Multiple Images

Consider a situation in which a set of 3D points $\mathbf{X}_j$ are viewed by a set of cameras with projection matrices $\mathbf{P}^i$. Denote by x_j^i the coordinates of the j-th point in the view of the i-th camera. We are going to solve the following reconstruction problem: given a set of image-point correspondences $\mathbf{x}_j^i$, find the set of camera matrices $\mathbf{P}^i$ and the 3D points $\mathbf{X}_j$ such that $\mathbf{P}^i\tilde{\mathbf{X}}_j \sim \tilde{\mathbf{x}}_j^i$. Without any further constraint on the scene or cameras, the reconstruction could only be retrieved up to a projective transformation. Accordingly, this reconstruction problem is called *projective reconstruction*. To understand it, it is necessary to master the knowledge introduced in the previous sections of this chapter.

This section addresses the projective reconstruction of a 3D scene from multiple images captured from it. When calibrated cameras are used, it is possible to reconstruct the scene up to scale from the images, i.e. *metric reconstruction*, in a manner similar to that of *projective reconstruction*. It will not be repeated in this section.

The optimal way to perform projective reconstruction from multiple images is to use *bundle adjustment*, which involves the minimization of the total reprojection error over the camera matrices and the 3D structure:

$$\min_{\mathbf{P}^i, \mathbf{X}_j} \sum_{ij} d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)^2 \tag{2.37}$$

where $\hat{\mathbf{x}}_j^i$ is the estimated image point computed from the estimated projection matrix $\mathbf{P}^i$ and the estimated 3D point $\mathbf{X}_j$, i.e. $\hat{\tilde{\mathbf{x}}}_j^i \sim \mathbf{P}^i\tilde{\mathbf{X}}_j$. The techniques for

conducting the bundle adjustment will be discussed in detail in chapter 5. However, bundle adjustment does not give a direct solution; it is a refining process involving a non-linear optimization which requires a good starting point, i.e. the initial reconstruction. The strategies to perform the initial reconstruction are generally classified into the following three categories.

1. Factorization

Factorization using singular value decomposition is often used for recovering 3D space and motion from image correspondences across multiple frames. Its basic idea is as follows:

Assuming that each point is visible in each view, i.e. $\mathbf{x}_j^i$ is known for all i, j, we may write the complete set of the equations $\tilde{\mathbf{x}}_j^i \sim \mathbf{P}^i \tilde{\mathbf{X}}_j$ as

$$\begin{bmatrix} \lambda_1^1\tilde{\mathbf{x}}_1^1 & \lambda_2^1\tilde{\mathbf{x}}_2^1 & \cdots & \lambda_n^1\tilde{\mathbf{x}}_n^1 \\ \lambda_1^2\tilde{\mathbf{x}}_1^2 & \lambda_2^2\tilde{\mathbf{x}}_2^2 & \cdots & \lambda_n^2\tilde{\mathbf{x}}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^m\tilde{\mathbf{x}}_1^m & \lambda_2^m\tilde{\mathbf{x}}_2^m & \cdots & \lambda_n^m\tilde{\mathbf{x}}_n^m \end{bmatrix} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \vdots \\ \mathbf{P}^m \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}_1 & \tilde{\mathbf{X}}_2 & \cdots & \tilde{\mathbf{X}}_n \end{bmatrix} \tag{2.38}$$

where weighting scalars λ_j^i are called the *projective depths* of the points. When these depths λ_j^i are known, $\mathbf{P}^i$ and $\mathbf{X}_j$ can be computed using SVD. In detail, denote the $3m \times n$-matrix on the left side of Eq. 2.38 by a measurement matrix $\mathbf{W}$. The rank of $\mathbf{W}$ should be less than 4, since it is a product of two matrices with 4 columns and rows respectively. Let $\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the SVD of $\mathbf{W}$, and $\hat{\mathbf{D}}$ be the diagonal matrix resulting from setting all but the first four diagonal entries of $\mathbf{D}$ to zero. Then the measurement matrix $\mathbf{W}$ is adjusted to be a rank-4 matrix $\hat{\mathbf{W}} = \mathbf{U}\hat{\mathbf{D}}\mathbf{V}^\top$. The camera matrices are hence retrieved from $[\mathbf{P^1}^\top, \mathbf{P^2}^\top, \ldots, \mathbf{P^m}^\top]^\top = \mathbf{U}\hat{\mathbf{D}}$ and the 3D points from $[\tilde{\mathbf{X}}_1^\top, \tilde{\mathbf{X}}_2^\top, \ldots, \tilde{\mathbf{X}}_n^\top] = \mathbf{V}^\top$. As the SVD of $\mathbf{W}$ is not unique, the factorization is not unique either and could only be determined up to a 4×4 projective transformation. That is, the reconstruction is projective. However, the projective depths λ_j^i are unknown, so they are given an initial estimate, e.g. $\lambda_j^i = 1$; and the above factorization process is iterated while projective depths λ_j^i are re-estimated in each iteration through the equations $\lambda_j^i \tilde{\mathbf{x}}_j^i = \mathbf{P}^i \tilde{\mathbf{X}}_j$ using the new estimated $\mathbf{P}^i$ and $\mathbf{X}_j$.

The factorization algorithm was first proposed for orthographic projection by Tomasi and Kanade in paper [66]. Method of iteration using this approach was proposed in [71] [30] [32]. Irani and Anandan proposed the covariance-weighted factorization, dealing with noisy feature correspondences with high degree of directional

uncertainty [33]. Factorization with line correspondences was proposed as well [59] [51].

The biggest disadvantage of factorization is that it requires each image feature must be visible in all the views. However, this is rarely the case in real data. Additionally, an algebraic error instead of the reprojection error is minimized in the process of factorization, therefore its result is not optimal and bundle adjustment should be conducted afterwards.

2. Hierarchical Merging of Sub-Sequences

The hierarchical technique is usually used for the reconstruction from a long image sequence. The basis idea is to first partition the sequence into manageable sub-sequences. There could be several hierarchical layers of the partition [63], and the sub-sequences in one layer may share overlapping images [38] [18] or may not [63]. The reconstruction is computed for each sub-sequence separately, and then they are "zipped" (merged) together using resection or triangulation [86] [38] [3]. Whenever several sub-sequences are registered into a longer sub-sequence or the complete sequence, bundle adjustment is usually necessary to be conducted upon the re-combined sequence.

In such a way, the hierarchical technique distributes the camera and structure "error" throughout the sequence of images and to some extend reduces the error accumulation from the first to the last image in the sequence; and accordingly it provides the final bundle adjustment quick convergency to a good minimum. But the advantage of the hierarchical technique is based on the costly expense of computational effort.

Another advantage of the hierarchic reconstruction is the possibility of parallel computation. The reconstructions of the sub-sequences are independent from each other and may be conducted in a parallel computing system synchronously, and so are the merging procedures. In such a way, the computational speed can be increased significantly.

The disadvantages of the hierarchical technique result from the merging procedure: 1) point matches between two views are usually ignored in this technique, but they are more common than the multiple-view point matches; 2) when there are two or more overlapping images between two successive sub-sequences, there is generally no transformation consistent with all the overlapping images between the two sub-sequences; therefore an additional non-linear minimization is necessarily conducted in order to maximize the consistency [18].

3. Incremental Reconstruction

A classical incremental reconstruction algorithm is given in the tutorial [53]. In the algorithm, the structure and view reconstruction is first conducted for two/three selected views (images) using a 2-view/3-view estimation method as stated in Sect. 2.2 and Sect. 2.3. For every additional view, the correspondences between the reconstructed 3D points in space and the image points in this additional view are set up and used to estimate the camera matrix for the added view. Then the 3D structure is reconstructed again with this added view. At last the overall reconstruction may be refined optionally through a global bundle adjustment over all the established views and the 3D structure. Obviously, the method has the disadvantage of error accumulation.

Avidan and Shashua [3] proposed another type of incremental reconstruction, i.e. threading two consecutive fundamental matrices using the trifocal tensor. In this algorithm, the view and structure error is distributed to each pair of consecutive view.

One may call the second incremental-reconstruction algorithm above a two-layered hierarchical-reconstruction algorithm, in which each sub-sequence consists of two views and every pair of successive sub-sequences share one common view. It shows that, there is some kind of connection between incremental reconstruction and two-layered hierarchical reconstruction. For incremental reconstruction, each incremental step estimates only one view; whereas a step in the ground layer of hierarchical reconstruction may estimate one or more views. Although it is claimed hierarchical reconstruction distributes the structure and view error throughout the sequence, the error is in fact still accumulated from the first to the last view after the sub-sequences are stitched together if the sequence is not closed. In other words, for a long open image sequence, error accumulation is an unavoidable inherent fact.

Zhang proposed another incremental reconstruction method in paper [85], which works on a sliding window of triplets of images. This method is related to the hierarchical reconstruction method proposed by Fitzgibbon and Zisserman in [18] and the incremental method proposed in [3]. The advantages of this algorithm over the other two algorithms are that 1) each incremental view is added through a local optimal estimation over three views, and that 2) it takes the three-view point matches as well as the two-view point matches into account.

In Chapter 6, another incremental reconstruction method will be proposed and implemented. It makes use of not only the two-view and three-view point matches, but also all the multiple-view point matches, at no extra expense of the computa-

tional effort.

Additionally, projective reconstruction with minimal feature correspondences were also researched on [58] [60]. It may be used to bootstrap robust estimation, such as RANSAC and LMS algorithms, to filter out the outliers for multiple-view reconstruction.

2.5 Conclusions

In this chapter some basic concepts in multiple-view geometry are introduced and some algorithms for multiple-view reconstruction are reviewed, including the projective geometry of a perspective camera model, the epipolar geometry between two camera views, the trifocal tensor between three views, and the 3D reconstruction from two or more camera images. For more details about the multiple-view geometry, readers are referred to books [15] [28] or [16].

3D-scene reconstruction from multiple views is a comprehensive research topic and concerned with various concepts and problems in computer vision. In the next three chapters, several techniques will be proposed to deal with some basic problems in the multiple-view reconstruction.

3 First-Order MLE Method for 3D-Point Reconstruction from Multiple Views

This chapter deals with the problem of finding the position of a 3D point in space given its projections in multiple images taken by cameras with known calibration and pose (position and orientation). Ideally the 3D point can be obtained as the intersection of the multiple known space rays back-projected from its projections. However, with noise the rays will not meet at a single 3D point generally. Therefore, it is necessary to find a best point of intersection.

In this chapter a new algorithm is proposed to obtain the *Maximum Likelihood Estimate* (*MLE*) of the true position of a 3D point in space from its projections in multiple views with known calibration and pose. The algorithm is based on the first-order approximation to the geometric error. In the case of two views, it is exactly the same as the *Sampson approximation* [27]. It is linear, non-iterative, simple in concept, and straightforward to implement. Through a series of experiments, the algorithm was extensively tested against many other reconstruction methods. It consistently obtains more accurate results than other linear methods, and its computational cost is also relatively low.

3D-point reconstruction is a basic problem in computer vision, but of great importance to multiple-view reconstruction, such as the *incremental motion estimation* [28] [85] and the *bundle adjustment* [75] [46] [28] [85] [7], as will be further discussed in chapter 5 and chapter 6.

In the following of this chapter, the reconstruction problem is briefly stated in Sect. 3.1. Then several reconstruction algorithms are reviewed in Sect. 3.2. Mean-

while, a generalized iterative least-squares method is proposed in Sect. 3.2.4, but it is not the main suggested method in this chapter. In Sect. 3.3 a special minimization criterion is presented for the reconstruction problem. According to this criterion, the *first-order MLE* method is proposed in Sect. 3.4. Experimental results on both simulated and real data are given in Sect. 3.5, to compare the proposed method with other methods. Finally the conclusions are given in Sect. 3.6.

3.1 Problem Statement

Suppose that a point $\mathbf{X}$ in 3-space is visible in N (≥ 2) views. Given its projections (or image points) $\mathbf{x}_i\,(i = 1, 2, \cdots, N)$ in the N views and the projection matrices of the views $\mathbf{P}_i$, we are expected to estimate the position of the 3D point $\mathbf{X}$. It is assumed that noise occurs only in the identification of the image points but not the camera matrices.

With $\mathbf{P}_i$ and $\mathbf{x}_i$, we can compute the N back-projected rays in space, which run from the N optical centers to the N corresponding image points, respectively. Hence, the problem of reconstructing a 3D-point is to find the intersection of the N rays in space. In the presence of noise, the rays are not guaranteed to intersect at a single point, hence a best choice of the intersection point is necessary to be found.

3.2 State of the Art

3.2.1 Numerical Optimization

Since it is assumed that noise only occurs in the measurement of image points, the maximum likelihood estimate of the 3D point is the one that minimizes the reprojection error, i.e. the summed squared distance between the measured image point $\mathbf{x}_i$ and the reprojection $\hat{\mathbf{x}}_i$ of the reconstructed 3D point $\mathbf{X}$

$$\mathcal{J}_0 = \sum_{i=1}^{N} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 \tag{3.1}$$

where $\tilde{\hat{\mathbf{x}}}_i \sim \mathbf{P}_i\tilde{\mathbf{X}}$. Function $d(\mathbf{x}_i, \hat{\mathbf{x}}_i)$ represents the geometric distance between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$.

It is commonly assumed that the measurement noise obeys a Gaussian distribution and is independent from view to view [28]. Let $\mathbf{\Sigma}_i$ be the covariance matrix for

measured image point $\mathbf{x}_i$. Then $d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$ is the *Mahalanobis distance* (or *covariance-weighted error*) between the two image points, i.e.

$$d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 = \| \mathbf{x}_i - \hat{\mathbf{x}}_i \|^2_{\mathbf{\Sigma}_i} = (\mathbf{x}_i - \hat{\mathbf{x}}_i)^\top {\mathbf{\Sigma}_i}^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i).$$

When an isotropic Gaussian noise distribution can be assumed, $d(*, *)$ refers to the Euclidean distance.

The minimization of the reprojection error Eq. 3.1 over the 3 parameters of $\mathbf{X}$ may be carried out with the a Newton-type optimizer such as Levenberg-Marquardt [50]. In addition, an initial estimate of $\mathbf{X}$ is necessarily computed before the minimization, through one of the following linear triangulation methods.

Numerical optimization is relatively slow in computation, and thus many linear methods as follows were proposed instead to solve the reconstruction problem.

3.2.2 Least-Squares Method

In Sect. 2.4.1, we have reviewed the *Least-Squares Method* or *Linear Triangulation* for reconstructing a 3D point $\mathbf{X}$ in space, given its projections ($\mathbf{x}_i = (u_i, v_i)^\top$) in N (≥ 2) views and the projection matrices ($\mathbf{P}_i$) of those views. In the method, the 3D point $\mathbf{X}$ is computed through minimizing the least-squares $\|\mathbf{A}\tilde{\mathbf{X}}\|$, where $\mathbf{A}$ is a $2N \times 4$-matrix composed of functions of $\mathbf{x}_i$ and $\mathbf{P}_i$ (see Eq. 2.34).

However, this algebraic solution of *Least-Squares Method* has no geometric meaning, since its result varies with the weights upon the rows of the matrix $\mathbf{A}$. Or theoretically, this linear method is non-projective invariant. That means that, under a projective transformation $\mathbf{H}$ of the space the original solution $\mathbf{X}$ does not correspond to a solution $\mathbf{HX}$ for the transformed problem.

3.2.3 Iterative Least-Squares Method

An alternative linear solution to the reconstruction problem is *Iterative Least-Squares Method* (*ILSM*) as proposed in [27] [7]. It changes the weights of the rows of the matrix $\mathbf{A}$ in each iteration, such that the least-squares of the weighted equations approaches the geometric errors adaptively. In detail, it aims at seeking a 3D-point $\mathbf{X}$ that minimizes

$$\mathcal{J}_I = \sum_{i=1}^{N} \| w_i \begin{bmatrix} u_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{1\top} \\ v_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{2\top} \end{bmatrix} \tilde{\mathbf{X}} \|^2 \tag{3.2}$$

where the weights $w_i = (\mathbf{P}_i^{3\top} \tilde{\mathbf{X}})^{-1}$, for $i = 1, \cdots, N$. Note that $\mathbf{P}_i^{3\top} \tilde{\mathbf{X}}$ is in fact the depth of 3D-point $\mathbf{X}$ in the coordinate system of view i (see Fig. 3.2). It is easy

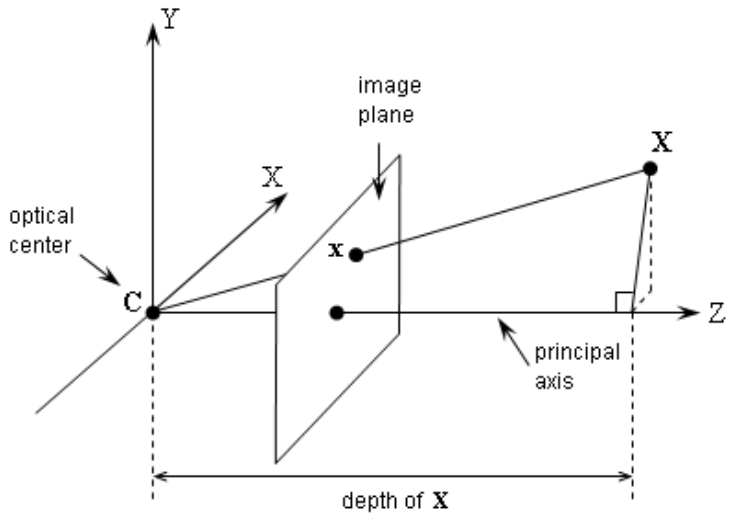


Figure 3.1: Depth of a 3D point in the coordinate system of a view.

to prove [27] that $\mathcal{J}_I$ is equal to the geometric reprojection error given in Eq. 3.1, when an isotropic Gaussian error distribution can be assumed for the measurement of the image points. $\mathcal{J}_I$ may be rewritten as the least-squares $\| \mathbf{B}\tilde{\mathbf{X}} \|^2$, where $\mathbf{B}$ is a $2N \times 4$-matrix

$$B = \begin{bmatrix} \cdots \\ w_i \begin{bmatrix} u_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{1\top} \\ v_i \mathbf{P}_i^{3\top} - \mathbf{P}_i^{2\top} \end{bmatrix} \\ \cdots \end{bmatrix}. \tag{3.3}$$

Besides $\mathbf{X}$, the true values of the scalars w_i are also unknown. Therefore w_i is given an initial estimate, e.g. $w_i = 1$, for $i = 1, \ldots, N$, and hence $\mathbf{X}$ can be solved using *SVD* or the normal equations (see Sect. 2.4.1). This process is repeated while w_i is re-estimated in each iteration by $(\mathbf{P}_i^{3\top}\tilde{\mathbf{X}})^{-1}$ using the new estimated $\mathbf{X}$.

Note that, the solution of the *Least-Squares Method* is equivalent to that of the *Iterative Least-Squares Method* in the first iteration with an initial estimate $w_i = 1$, for $i = 1, \ldots, N$. Moreover, the inversion of the weights in Eq. 3.3, $w_i^{-1} = \mathbf{P}_i^{3\top}\tilde{\mathbf{X}}$ is in fact the depth of the 3D point $\mathbf{X}$ in the coordinate system of view i (see Fig. 3.1). Therefore we can tell, when the depth of a space point does not vary significantly over the views compared with the value of the depth (i.e. $w_i \approx w_j$, when $i \neq j$), the Least-Squares Method (*LSM*) of 3D-point reconstruction can obtain similar accuracy with *ILSM*. This point is also shown by the experimental results in Sect. 3.5.

ILSM obtains more accurate results than *LSM*. But as an iterative method, *ILSM* is still relatively slow in computation.

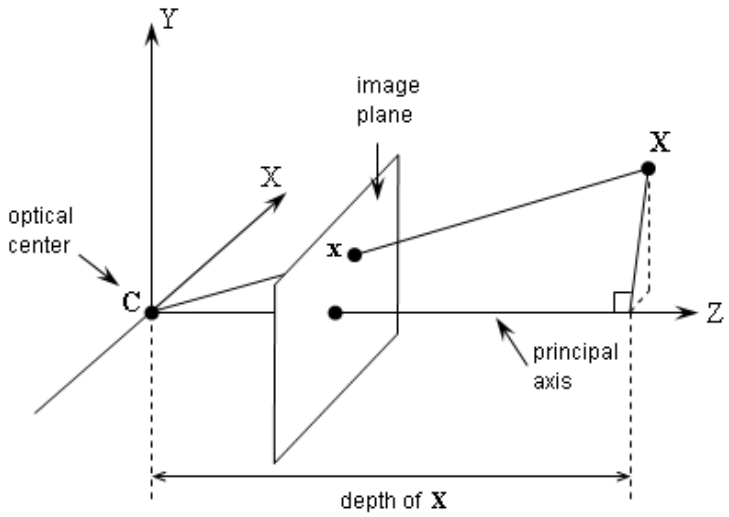


Figure 3.2: **Depth of a 3D-point in the coordinate system of a view.**

3.2.4 Generalized Iterative Least-Squares Method

In the presence of a general Gaussian noise distribution, $\| \mathbf{B}\tilde{\mathbf{X}} \|^2$ in the above section is not equal to the geometric reprojection error in Eq. 3.1. This section will propose another linear triangulation method. It is a *generalized ILSM* method and it works under the assumption of a general Gaussian noise distribution.

In this algorithm, function $\| \mathbf{MB}\tilde{\mathbf{X}} \|^2$ instead of $\| \mathbf{B}\tilde{\mathbf{X}} \|^2$ is minimized, where $\mathbf{M}^\top\mathbf{M} = \mathbf{\Sigma}^{-1}$ and $\mathbf{\Sigma}$ is the covariance matrix of the measurement errors of image points. It is easy to prove that $\| \mathbf{MB}\tilde{\mathbf{X}} \|^2$ is equal to the geometric reprojection error in terms of Mahalanobis distance.

Sect. 2.4.1 introduced two methods: *SVD* and the *normal equation*, to solve the least-squares problem of minimizing $\| \mathbf{A}\tilde{\mathbf{X}} \|$, $\| \mathbf{B}\tilde{\mathbf{X}} \|$ or $\| \mathbf{M}B\tilde{\mathbf{X}} \|$. However, the *SVD* solution is even variant to Euclidean transformation of the world coordinate. Therefore, the normal-equation solution is suggested. It is not only faster to compute than *SVD*, but also invariant to the affine transformation of the world coordinate system. In the following of this chapter, the normal-equation solution is always implied in default.

In general *ILSM* obtains more accurate results than *LSM*. But when the depth of the space point does not vary much across the views, *LSM* may achieve as good results as *ILSM*, since in such a case the weights w_i in matrix $\mathbf{B}$ are nearly identical and thus will not change much through the iterations of *ILSM*.

Moreover, due to the advanced research on numerical optimization during the re-

cent years, the iterative *ILSM* method does not run much faster than some numerical optimizers, such as the advance Levenberg-Marquardt optimizer [37].

These conclusions are confirmed by the experimental results.

3.2.5 Other Methods for 3D-Point Reconstruction from Two views

For two views, many methods have been proposed to reconstruct a space point, including the above three. An *optimal triangulation* method was proposed in [27], which reduces the problem of point reconstruction to one of finding the minimum of a sixth-order polynomial function with a single variable. When isotropic Gaussian noise distribution can be assumed, it can be proved that this method obtains the optimal solution. But when the number of views increases, it would be too difficult to create a corresponding polynomial function with only one variable.

Mid-point method is another commonly-known method for two-view triangulation. It is to find the mid-point of the common perpendicular to the two rays corresponding to the matched points. This method is easy to compute, but it is neither affine nor projective invariant. It behaves very poorly under projective and affine transformation, and has been seldom used in precise computation.

3.3 A Proposed Minimization Criterion for 3D-Point Reconstruction from Multiple Views

3.3.1 Representation of Intersection Constraint

Suppose that a point $\mathbf{X}$ in 3-space is projected in $N(\geq 2)$ views. Let $\mathbf{x}_i$ be the projection of the 3D-point $\mathbf{X}$ in view i $(i = 1, \ldots, N)$. The projection matrix $\mathbf{P}_i$ of each view is known, and hence the fundamental matrix $\mathbf{F}_{i,j}$ between each pair of views is known. In the absence of noise, the N back-projected rays corresponding to the N image points meet at a single point in space, which is the 3D-point $\mathbf{X}$, and each pair of matched image points $(\mathbf{x}_i, \mathbf{x}_j)$ must satisfy the epipolar constraint

$$\tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j} \tilde{\mathbf{x}}_i = 0. \tag{3.4}$$

In the case of two views, the intersection constraint can be expressed by the epipolar constraint between the two image points (see [27]). In other words, an epipolar constraint is equivalent to the intersection constraint between two rays in space. When $N \geq 3$ and not all the optical centers of the views are coplanar

with the 3D-point, the intersection constraint can also be expressed by the epipolar constraints between each pair of image points. There are a total of $\frac{N(N-1)}{2}$ pairs of image points (or views), and accordingly $\frac{N(N-1)}{2}$ pairwise epipolar constraints. However, only $2N-3$ of them [48] [78] are necessary and sufficient to be used as the intersection constraints upon the N rays.

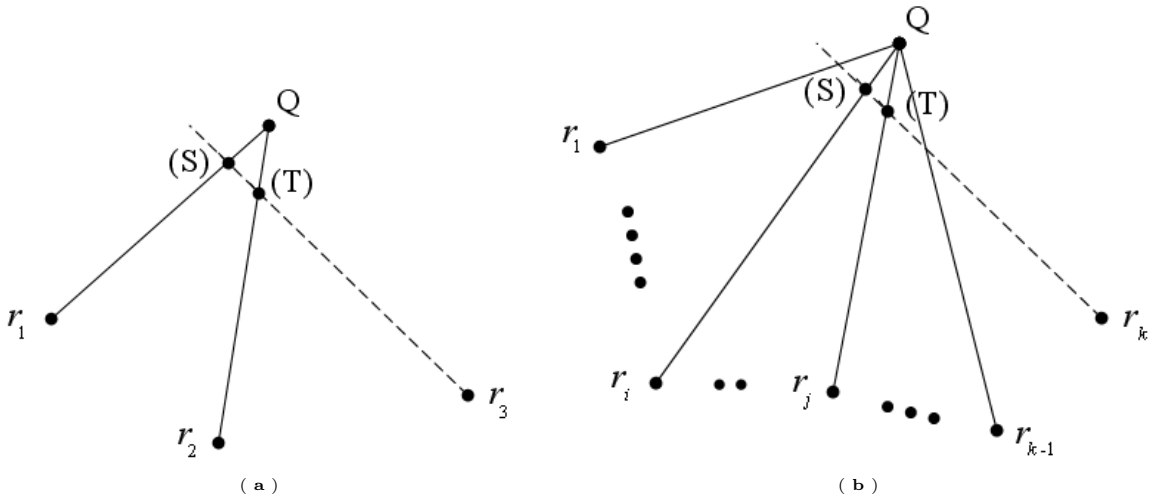


Figure 3.3: **Intersection of non-coplanar rays.** (a) *When the rays r_1, r_2 and r_3 are non-coplanar, the points **Q**, **S** and **T** are a same point.* (b) *Similarly when the rays r_i, r_j and r_k are non-coplanar, the points **S** and **T** are the same point with **Q**.*

Assume the three rays r_1, r_2 and r_3 are the related back-projected rays from three views and they are non-coplanar. First we use one epipolar constraint to define that r_1 and r_2 cross at a single 3D-point **Q**. When the third ray r_3 is added, we add another two epipolar constraints to define that r_3 intersects r_1 at a 3D-point **S**, and intersects r_2 at a 3D-point **T**. Since r_1, r_2 and r_3 are non-planar, **S** and **T** must be the same point as **Q** (see Fig. 3.3a). That is, the three rays intersect at a single point in space. Similarly, each time when a new ray r_k ($k > 3$) is added, there always exist two rays, say r_i and r_j among the first $k-1$ rays such that they are non-coplanar with r_k (see Fig. 3.3b). Therefore, we only need to add another two epipolar constraints to define that r_i intersects r_k and that r_j intersects r_k. Then r_k is sufficiently constrained to intersect the first k rays at the same point. Therefore, we can conclude, for N views only $1+(N-2)*2$ epipolar constraints are necessary and sufficient to enforce the intersection of all the N rays at a single 3D-point, as long as not all of them are coplanar.

When $N \geq 4$, the possible combinations of such $2N-3$ independent constraints are

not unique. The following set of epipolar constraints is one possibility to represent the intersection constraint of the N back-projected rays:

$$\begin{array}{ll} \tilde{\hat{\mathbf{x}}}_2^\top \mathbf{F}_{1,2} \tilde{\hat{\mathbf{x}}}_1 = 0, & \\ \tilde{\hat{\mathbf{x}}}_3^\top \mathbf{F}_{2,3} \tilde{\hat{\mathbf{x}}}_2 = 0, & \tilde{\hat{\mathbf{x}}}_3^\top \mathbf{F}_{1,3} \tilde{\hat{\mathbf{x}}}_1 = 0, \\ \tilde{\hat{\mathbf{x}}}_4^\top \mathbf{F}_{2,4} \tilde{\hat{\mathbf{x}}}_2 = 0, & \tilde{\hat{\mathbf{x}}}_4^\top \mathbf{F}_{1,4} \tilde{\hat{\mathbf{x}}}_1 = 0, \\ \quad \vdots & \quad \vdots \\ \tilde{\hat{\mathbf{x}}}_N^\top \mathbf{F}_{2,N} \tilde{\hat{\mathbf{x}}}_2 = 0, & \tilde{\hat{\mathbf{x}}}_N^\top \mathbf{F}_{1,N} \tilde{\hat{\mathbf{x}}}_1 = 0 \end{array} \tag{3.5}$$

where it is assumed that there exist no other optical centers of the views in the plane determined by the 3D-point and the two optical centers of view 1 and view 2.

3.3.2 The Proposed Minimization Criterion

It is assumed in this chapter that the projection matrices of those views as well as the fundamental matrices between the view pairs are known precisely, or at least with great accuracy compared with the measured image points. Thus, the *maximum likelihood estimate* of the 3D point depends on the assumption of the error model of image-point measurement; and as given in Sect. 3.2.1, the geometric cost function may be computed as the summed squared distance between the measured image point $\mathbf{x}_i$ and its estimation $\hat{\mathbf{x}}_i$

$$\mathcal{J} = \sum_{i=1}^{N} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2, \tag{3.6}$$

subject to the intersection constraint of the back-projected rays, e.g. the $2N-3$ equations in Eq. 3.6.

Therefore, seeking the estimated "true" image points $\hat{\mathbf{x}}_i$ $(i = 1, 2, \cdots, N)$ that minimize the objective function Eq. 3.6 subject to constraints in Eq. 3.5 can be used as a minimization criterion for reconstructing a 3D point in space from its projections in $N(\geq 2)$ views. As stated above, the constraints in Eq. 3.5 are possibly substituted by another set of $2N-3$ independent epipolar constraints, when $N \geq 4$.

In the presence of Gaussian noise, the estimated image points $\hat{\mathbf{x}}_i$ that meet the above minimization criterion in Eq. 3.5 are the *Maximum Likelihood Estimates* of the true values of the observed image points x_i. The intersection point of their back-projected rays is accordingly the MLE of the true 3D-point in space.

3.4 The Proposed Method of 3D-Point Reconstruction from Multiple Views

In this section, a linear and non-iterative method is proposed to reconstruct a 3D-point in space from its projections in multiple views. First, it estimates the true values of the image points through the first-order approximation to the geometric error, and then reconstructs the 3D-point using the estimated image points. Due to the first-order approximation, we may call this method *1st-order MLE*.

In the following, the first-order correction of image points in the presence of isotropic Gaussian noise is firstly deduced. Then it is shown that, with the corrected image points, the 3D point is possibly reconstructed using various linear triangulation methods. The according solution in the presence of a general Gaussian noise distribution is also proposed nextly. At last, several special topics on this first-order solution are discussed.

3.4.1 First-Order Geometric Correction of the Image Points

Assuming that the measurement of image points follows an isotropic Gaussian noise distribution, the objective function in Eq. 3.6 can then be rewritten as

$$\mathcal{J} = \sum_{i=1}^{N} \| \mathbf{x}_i - \hat{\mathbf{x}}_i \|^2 = \sum_{i=1}^{N} \Delta\mathbf{x}_i^\top \Delta\mathbf{x}_i = \Delta\mathbf{x}^\top \Delta\mathbf{x} \tag{3.7}$$

subject to the epipolar constraints given in Eq. 3.5, where $\| \cdot \|$ refers to Euclidean distance, and $2N$-vector $\Delta\mathbf{x} = [\Delta\mathbf{x}_1^\top, \Delta\mathbf{x}_2^\top, \ldots, \Delta\mathbf{x}_N^\top]^\top$.

By applying the technique of Lagrange multiplier, the constrained minimization problem can be converted into an unconstrained minimization problem with a new objective function given by

$$\mathcal{J}' = \mathcal{J} + \lambda_{1,2}\mathcal{F}_{1,2} + \sum_{j=3}^{N} (\lambda_{2,j}\mathcal{F}_{2,j} + \lambda_{1,j}\mathcal{F}_{1,j}) \tag{3.8}$$

where $\mathcal{F}_{i,j} = \tilde{\hat{\mathbf{x}}}_j^\top \mathbf{F}_{i,j} \tilde{\hat{\mathbf{x}}}_i$, and $\lambda_{i,j}$ is the Lagrange multiplier.

Let $(2N-3)$-vector $\lambda = [\lambda_{1,2}\ \lambda_{2,3}\ \lambda_{1,3}\ \ldots\ \lambda_{2,N}\ \lambda_{1,N}]^\top$, and $(2N-3)$-vector $\mathcal{F} = [\mathcal{F}_{1,2}\ \mathcal{F}_{2,3}\ \mathcal{F}_{1,3}\ \ldots\ \mathcal{F}_{2,N}\ \mathcal{F}_{1,N}]^\top$, then the objective function of the unconstrained minimization problem can be rewritten as

$$\mathcal{J}' = \mathcal{J} + \lambda^\top \mathcal{F}. \tag{3.9}$$

Now we may expand the individual $\mathcal{F}_{i,j}$ as follows:

$$\begin{aligned}\mathcal{F}_{i,j} &= (\tilde{\mathbf{x}}_j - \Delta\tilde{\mathbf{x}}_j)^\top \mathbf{F}_{i,j}(\tilde{\mathbf{x}}_i - \Delta\tilde{\mathbf{x}}_i) \\ &= \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\Delta\tilde{\mathbf{x}}_i - \Delta\tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i + \Delta\tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\Delta\tilde{\mathbf{x}}_i \\ &= 0.\end{aligned} \tag{3.10}$$

Neglect the second-order term, then we get

$$\begin{aligned}\mathcal{F}_{i,j} &\approx \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\Delta\tilde{\mathbf{x}}_i - \Delta\tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i \\ &= \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\mathbf{Z}\Delta\mathbf{x}_i - \tilde{\mathbf{x}}_i^\top \mathbf{F}_{j,i}\mathbf{Z}\Delta\mathbf{x}_j \\ &\approx 0\end{aligned} \tag{3.11}$$

where the 3×2-matrix $\mathbf{Z}$ [85] is given by

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^\top. \tag{3.12}$$

With it, we can convert easily between $\Delta\tilde{\mathbf{x}}$ and $\Delta\mathbf{x}$, for $\Delta\tilde{\mathbf{x}} = \mathbf{Z}\Delta\mathbf{x}$ and $\Delta\mathbf{x} = \mathbf{Z}^\top\Delta\tilde{\mathbf{x}}$.

We define a set of 2-vectors $\mathbf{h}_{i,j} = \mathbf{Z}^\top \mathbf{F}_{j,i}\tilde{\mathbf{x}}_j$ and a set of scalars $e_{i,j} = \tilde{\mathbf{x}}_j^\top \mathbf{F}_{i,j}\tilde{\mathbf{x}}_i$, in order to simplify the notation. Note that $e_{i,j} = e_{j,i}$, but $\mathbf{h}_{i,j} \neq \mathbf{h}_{j,i}$. Therefore, Eq. 3.11 can be rewritten as

$$\mathcal{F}_{i,j} \approx e_{i,j} - \begin{bmatrix} \mathbf{h}_{i,j}^\top & \mathbf{h}_{j,i}^\top \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_i \\ \Delta\mathbf{x}_j \end{bmatrix} \approx 0. \tag{3.13}$$

To minimize $\mathcal{J}'$, we may set its first-order derivatives to zero with respect to $\Delta\mathbf{x}_i$, $i = 1, 2, \ldots, N$, which yields

$$\begin{aligned}&\left(\frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{x}_1}\right)^\top = 2\Delta\mathbf{x}_1 - \textstyle\sum_{j=2}^{N} \lambda_{1,j}\mathbf{h}_{1,j} = 0, \\ &\left(\frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{x}_2}\right)^\top = 2\Delta\mathbf{x}_2 - \lambda_{1,2}\mathbf{h}_{2,1} - \textstyle\sum_{j=3}^{N} \lambda_{2,j}\mathbf{h}_{2,j} = 0, \\ &\text{and for } i = 3, \ldots N, \\ &\left(\frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{x}_i}\right)^\top = 2\Delta\mathbf{x}_i - \lambda_{2,i}\mathbf{h}_{i,2} - \lambda_{1,i}\mathbf{h}_{i,1} = 0.\end{aligned} \tag{3.14}$$

From Eq. 3.14 we obtain a total of $2N$ linear equations, which may be written in

the following form

$$\begin{bmatrix} \Delta\mathbf{x}_1 \\ \Delta\mathbf{x}_2 \\ \Delta\mathbf{x}_3 \\ \vdots \\ \Delta\mathbf{x}_{N-1} \\ \Delta\mathbf{x}_N \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{h}_{1,2} & & \mathbf{h}_{1,3} & & \mathbf{h}_{1,4} & \dots & & \mathbf{h}_{1,N} \\ \mathbf{h}_{2,1} & \mathbf{h}_{2,3} & & \mathbf{h}_{2,4} & & \dots & \mathbf{h}_{2,N} & \\ & \mathbf{h}_{3,2} & \mathbf{h}_{3,1} & & & & & \\ & & & \mathbf{h}_{4,2} & \mathbf{h}_{4,1} & & & \\ & & & & & \ddots & & \\ & & & & & & \mathbf{h}_{N,2} & \mathbf{h}_{N,1} \end{bmatrix} \begin{bmatrix} \lambda_{1,2} \\ \lambda_{2,3} \\ \lambda_{1,3} \\ \lambda_{2,4} \\ \lambda_{1,4} \\ \vdots \\ \lambda_{2,N} \\ \lambda_{1,N} \end{bmatrix}$$

$$\text{abbr.} \qquad \Delta\mathbf{x} = \tfrac{1}{2}\mathbf{H}_1\lambda \tag{3.15}$$

where $\mathbf{H}_1$ is a $(2N) \times (2N-3)$-matrix.

Additionally, from the epipolar constraints in Eq. 3.5, we can obtain a total of $2N-3$ linear equations as Eq. 3.13, which may be written as

$$\begin{bmatrix} e_{1,2} \\ e_{2,3} \\ e_{1,3} \\ \vdots \\ e_{2,N} \\ e_{1,N} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{1,2}^\top & \mathbf{h}_{2,1}^\top & & & & & \\ & \mathbf{h}_{2,3}^\top & \mathbf{h}_{3,2}^\top & & & & \\ \mathbf{h}_{1,3}^\top & & \mathbf{h}_{3,1}^\top & & & & \\ & \mathbf{h}_{2,4}^\top & & \mathbf{h}_{4,2}^\top & & & \\ \mathbf{h}_{1,4}^\top & & & \mathbf{h}_{4,1}^\top & & & \\ \vdots & \vdots & & & \ddots & & \\ & \mathbf{h}_{2,N}^\top & & & & \mathbf{h}_{N,2}^\top & \\ \mathbf{h}_{1,N}^\top & & & & & \mathbf{h}_{N,1}^\top & \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_1 \\ \Delta\mathbf{x}_2 \\ \Delta\mathbf{x}_3 \\ \Delta\mathbf{x}_4 \\ \vdots \\ \Delta\mathbf{x}_N \end{bmatrix}$$

$$\text{abbr.} \qquad \mathbf{e} = \mathbf{H}_2^\top \Delta\mathbf{x} \tag{3.16}$$

where $(2N-3)$-vector $\mathbf{e} = [e_{1,2}\ e_{2,3}\ e_{1,3}\ \dots\ e_{2,N}\ e_{1,N}]^\top$.

Comparing the two sets of linear equations in Eq. 3.15 and Eq. 3.16, we can see the two sparse matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are equal. We define matrix $\mathbf{H} = \mathbf{H}_1 = \mathbf{H}_2$. Then Eq. 3.15 and Eq. 3.16 may be rewritten as

$$\Delta\mathbf{x} = \frac{1}{2}\mathbf{H}\lambda, \tag{3.17}$$

$$\mathbf{e} = \mathbf{H}^\top \Delta\mathbf{x}. \tag{3.18}$$

In the above two sets of linear equations, $\mathbf{e}$ and $\mathbf{H}$ are functions of the observed image points and the known fundamental matrices and can be computed directly, whereas λ and $\Delta\mathbf{x}$ are still the unknowns and to be computed.

Substituting Eq. 3.17 into Eq. 3.18, we get

$$\mathbf{e} = \frac{1}{2}\mathbf{H}^\top\mathbf{H}\lambda. \tag{3.19}$$

Then λ can be computed as

$$\lambda = 2(\mathbf{H}^\top\mathbf{H})^-\mathbf{e} \tag{3.20}$$

where $(\mathbf{H}^\top\mathbf{H})^-$ is the generalized inverse (or pseudo inverse) of the $(2N-3)\times(2N-3)$-matrix $\mathbf{H}^\top\mathbf{H}$. When the preconditions in selecting the $2N-3$ epipolar constraints are satisfied, $\mathbf{H}^\top\mathbf{H}$ is of full rank and $(\mathbf{H}^\top\mathbf{H})^-$ is equal to $(\mathbf{H}^\top\mathbf{H})^{-1}$.

Substitute Eq. 3.20 into Eq. 3.17, then the solution to $\Delta\mathbf{x}$ is obtained:

$$\Delta\mathbf{x} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^-\mathbf{e} \tag{3.21}$$

and the residual of the objective function is then

$$\mathcal{J} = \| \Delta\mathbf{x} \|^2 = \Delta\mathbf{x}^\top\Delta\mathbf{x} = \mathbf{e}^\top(\mathbf{H}^\top\mathbf{H})^-\mathbf{e}. \tag{3.22}$$

Because of the first-order approximation in Eq. 3.11 during the whole deduction process, the obtained $\Delta\mathbf{x}_i, (i = 1, \ldots, N)$ is called the *first-order geometric correction* of the image points $\mathbf{x}_i$, and $\hat{\mathbf{x}}_i = \mathbf{x}_i - \Delta\mathbf{x}_i$ is the *first-order maximum likelihood estimate* (*1st-order MLE*) of the true value of the image points.

3.4.2 3D-Point Reconstruction Using the Estimated Image Points

Due to the first-order approximation, the epipolar constraints in Eq. 3.5 are not satisfied precisely. Accordingly, the back-projected rays of the estimated image points do not exactly but almost meet at a single point in space. Therefore, we may use the estimated image points to reconstruct the 3D-point.

Several methods can be used to conduct the reconstruction, since the reprojection error for the estimated image points is just subtle. One way is using all the estimated image points to reconstruct the 3D-point with the *LSM* method (or linear triangulation) as stated in Sect. 2.4.1. Because the estimated image points have been corrected to a great extent to meet the minimization criterion as given in Sect. 3.3, the *LSM* reconstruction using the estimates obtains more accurate results than using the original observed image points. The difference will be shown through the experimental results in Sect. 3.5.

Alternative methods to reconstruct the 3D-point use some of the estimated image points, e.g. $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$, instead of all the points. When N is large, this alternative

could reduce the computational cost significantly at the cost of a very slight loss of accuracy. It will be further discussed in Sect. 3.5.

Note that the reprojection error for the reconstructed 3D-point is not exactly equal to the residual obtained in Eq. 3.22 generally. If necessary, the reprojection error should be re-computed using the reconstructed point.

3.4.3 Solution for General Gaussian Noise Distribution

After the first-order approximation in Eq. 3.11, the problem solved in Sect. 3.4.1 is in fact converted into one of quadratic optimization constrained by linear equations. We can rewrite the problem in the following form

$$\mathcal{J}[\Delta\mathbf{x}] = \Delta\mathbf{x}^\top\boldsymbol{\Sigma}^{-1}\Delta\mathbf{x} \rightarrow \min \tag{3.23}$$

constrained by

$$\mathbf{H}^\top\Delta\mathbf{x} = \mathbf{e} \tag{3.24}$$

where matrix $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & & & \\ & \boldsymbol{\Sigma}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Sigma}_N \end{bmatrix}. \tag{3.25}$$

$\boldsymbol{\Sigma}_i$ is the covariance matrix of the measurement errors of the image point $\mathbf{x}_i$.

Solution to such linearly-constrained quadratic optimization has been given in [34] by

$$\Delta\mathbf{x} = \boldsymbol{\Sigma}\mathbf{H}(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{e} \tag{3.26}$$

and the residual is

$$\mathcal{J}[\Delta\mathbf{x}] = \mathbf{e}^\top(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{e}. \tag{3.27}$$

In the presence of an isotropic Gaussian noise distribution, Eq. 3.26 and Eq. 3.27 are equivalent to Eq. 3.21 and Eq. 3.22, respectively. In the presence of a general Gaussian noise distribution, the same solution as in Eq. 3.26 and Eq. 3.27 can be obtained as well through the same deduction as described in Sect. 3.4.1.

Additionally, when another set of $2N - 3$ independent epipolar constraints is used in the minimization criterion in Eq. 3.5, the minimization problem can still be converted into one of such linearly-constrained quadratic optimization only with a little difference in the matrix $\mathbf{H}$ and can be solved in the same way.

3.4.4 Arguments in the First-Order Geometric Solution

Equivalence with Sampson Approximation for a Two-View correspondence

Let us consider the case when a 3D point is only captured in two views, i.e. $N = 2$. When an isotropic Gaussian noise distribution can be assumed for the measurement error of image points, the image-point correction in Eq. (3.21) and the minimized residual in (3.22) are computed as

$$\Delta\mathbf{x} = \frac{e_{1,2}}{\mathbf{h}_{1,2}^\top \mathbf{h}_{1,2} + \mathbf{h}_{2,1}^\top \mathbf{h}_{2,1}} \begin{bmatrix} \mathbf{h}_{1,2} \\ \mathbf{h}_{2,1} \end{bmatrix} \tag{3.28}$$

and

$$\mathcal{J} = \frac{e_{1,2}^2}{\mathbf{h}_{1,2}^\top \mathbf{h}_{1,2} + \mathbf{h}_{2,1}^\top \mathbf{h}_{2,1}} \tag{3.29}$$

These two equations are exactly the same with the results obtained with *Sampson approximation* [28], i.e. the first-order geometric correction and cost function.

Independence from the Scales of Individual Fundamental Matrices

We see both matrix $\mathbf{H}$ and vector $\mathbf{e}$ in Eqs. (3.26) and (3.27) are composed of the homogeneous fundamental matrices $\mathbf{F}_{ij}$, so questions may be raised whether the results depend on the scales of the individual fundamental matrices.

Now let us use another set of fundamental matrices, $\mathbf{F}'_{ij} = w_{ij}\mathbf{F}_{ji}$, where w_{ij} is a scalar and $w_{ij} = w_{ji}$ to guarantee $\mathbf{F}_{ij} = \mathbf{F}_{ij}^\top$. Define a $(2N-3)\times(2N-3)$ diagonal matrix $\mathbf{W} = diag(w_{12}, w_{23}, w_{21}, w_{24}, w_{14}, \ldots, w_{2N}, w_{1N})$. Then the corresponding scaled $\mathbf{H}' = \mathbf{HW}$ and $\mathbf{e}' = \mathbf{We}$. Because $\mathbf{W}$ is a diagonal matrix, it can easily prove that $(\mathbf{W}^\top\mathbf{MW})^- = \mathbf{W}^-\mathbf{M}^-\mathbf{W}^-$ for an arbitrary square matrix $\mathbf{M}$ of the same size with $\mathbf{W}$. Then we have

$$\begin{aligned}
\Delta\mathbf{x}' &= \boldsymbol{\Sigma}\mathbf{H}'(\mathbf{H}'^\top\boldsymbol{\Sigma}\mathbf{H}')^-\mathbf{e}' \\
&= \boldsymbol{\Sigma}\mathbf{HW}(\mathbf{W}^\top\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{HW})^-\mathbf{We} \\
&= \boldsymbol{\Sigma}\mathbf{HWW}^-(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{W}^-\mathbf{We} \\
&= \boldsymbol{\Sigma}\mathbf{H}(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{e} \\
&= \Delta\mathbf{x}
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{J}[\Delta\mathbf{x}'] &= \mathbf{e}'^\top(\mathbf{H}'^\top\boldsymbol{\Sigma}\mathbf{H}')^-\mathbf{e}' \\
&= \mathbf{e}^\top\mathbf{W}^\top(\mathbf{W}^\top\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{HW})^-\mathbf{We} \\
&= \mathbf{e}^\top\mathbf{WW}^-(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{W}^-\mathbf{We} \\
&= \mathbf{e}^\top(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{We} \\
&= \mathcal{J}[\Delta\mathbf{x}].
\end{aligned}$$

The above deduction shows that both the image-point corrections in Eq. (3.26) and the residual in Eq. (3.27) are consistent with the scales of the fundamental matrices.

Avoid Biased-Estimation When Using Canonical Coordinates

Moreover, it is important to point out that, when image points in the canonical coordinates $\mathbf{x}'_i$ are used in the objective function $\mathcal{J}$, matrix $\mathbf{\Sigma}_i$ should be replaced as well by the covariance matrix $\mathbf{\Sigma}'_i$ with respect to $\Delta\mathbf{x}'_i$. The canonical image point $\mathbf{x}'_i$ is defined in such a way that $\tilde{\mathbf{x}}'_i = \mathbf{K}_i^{-1}\tilde{\mathbf{x}}_i$. $\mathbf{K}_i$ is the calibration matrix of the i-th view, which can be written as

$$\mathbf{K}_i = \begin{bmatrix} f_{xi} & \alpha_i & x_{0i} \\ 0 & f_{yi} & y_{0i} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.30}$$

Then the covariance matrix of $\Delta\mathbf{x}'_i$ is computed as

$$\mathbf{\Sigma}'_i = \begin{bmatrix} \frac{1}{f_{xi}} & -\frac{\alpha_i}{f_{xi}f_{yi}} \\ 0 & \frac{1}{f_{yi}} \end{bmatrix} \mathbf{\Sigma}_i \begin{bmatrix} \frac{1}{f_{xi}} & 0 \\ -\frac{\alpha_i}{f_{xi}f_{yi}} & \frac{1}{f_{yi}} \end{bmatrix}. \tag{3.31}$$

The problem of minimizing $\Delta\mathbf{x}'^{\top}\mathbf{\Sigma}^{-1}\Delta\mathbf{x}'$ is equivalent to that of minimizing $\Delta\mathbf{x}'^{\top}\mathbf{\Sigma}'^{-1}\Delta\mathbf{x}'$ or $\Delta\mathbf{x}^{\top}\mathbf{\Sigma}^{-1}\Delta\mathbf{x}$, if and only if the following conditions are satisfied:

- $\alpha_i = 0$ and $f_{xi} = f_{yi}$, for $i = 1, 2, \ldots, N$;
- $f_{x1} = f_{xi}$, for $i = 2, 3, \ldots, N$.

Otherwise, $\Delta\mathbf{x}'^{\top}\mathbf{\Sigma}'^{-1}\Delta\mathbf{x}'$ instead of $\Delta\mathbf{x}'^{\top}\mathbf{\Sigma}^{-1}\Delta\mathbf{x}'$ must be minimized. This difference is very essential, or else *biased* estimation will be conducted. However, this point has been ignored in some papers [48] [47] before.

3.5 Experiments and Discussions

In this section, the experimental results with both simulated and real data are presented. Several algorithms are evaluated, including:

LSM. Least-Squares Method as described in Sect. 3.2.2.

ILSM. Iterative Least-Squares Method as described in Sect. 3.2.3.

MLE1. the proposed First-Order Maximum Likelihood Estimation, with the linear triangulation using all the corrected image points.

MLE2. First-Order Maximum Likelihood Estimation, with the linear triangulation using the first two corrected image points.

LM. Levenberg-Marquardt optimizer [37] used to minimize the cost function in Eq. 3.1, with the result of *LSM* as the initial estimate of the point in space. The 3D position of the space point is the parameter vector of the optimization, and its goal vector is composed of the 2D positions of the observed image points.

The threshold t for the convergence of both *ILSM* and *LM* is set to 10^{-8}. That means, the iteration is terminated when the relative reduce of the reprojection error in one iteration is less than t.

In the following experiments, it is assumed that error occurs only in the measured image coordinates, but not in the camera matrices; and an isotropic Gaussian noise distribution is assumed for the measurement of image points.

3.5.1 Experiment on Simulated Data

In this section, we set up a simulated scene, in which twelve cameras are posed around facing down at the points in space (see Fig. 3.4). The calibration and exact pose of the cameras are known. We set the focal lengths of all the cameras to 1000, and every image size is 512×512. 10,000 3D points are chosen in front of the cameras. The image points are corrupted by an additive independent Gaussian noise [10] in each dimension, with mean 0 and standard deviation σ.

Outliers are filtered out with significance level $\alpha\% = 5\%$ (or confidence level 95%), if

- its total reprojection error in Eq. (3.7) falls into the rejection region $(\chi^2_{2N,\alpha}\sigma^2, \infty)$,
- or its reprojection error of any image point falls into the rejection region $(\chi^2_{2,\alpha}\sigma^2, \infty)$.

$\chi^2_{r,\alpha}$ could be looked up in [9].

Reconstruction accuracy.

First, we fix the number of views to 8, and vary the standard deviation σ of the noise in each dimension of the image points, to compared the average reprojection error obtained by the several reconstruction methods. In the experiment, the iterative optimizer *LM* always gets the minimal reprojection errors. In Fig. 3.5a we show the average reprojection error obtained by *LM*.

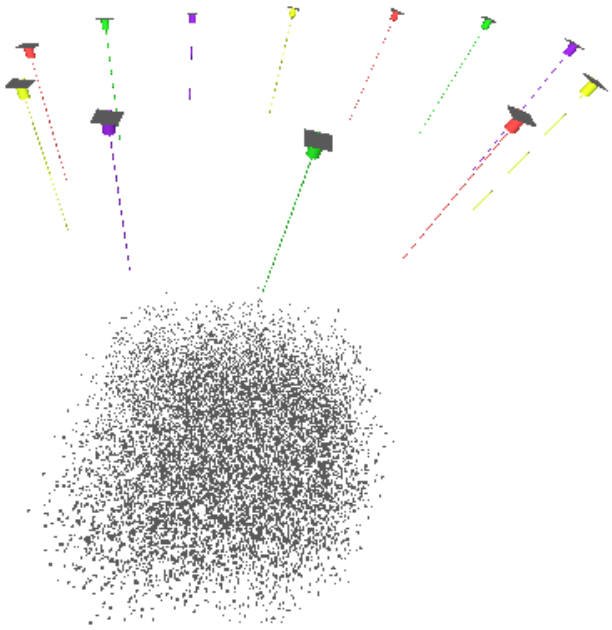


Figure 3.4: Poses of cameras in the simulated circumstance.

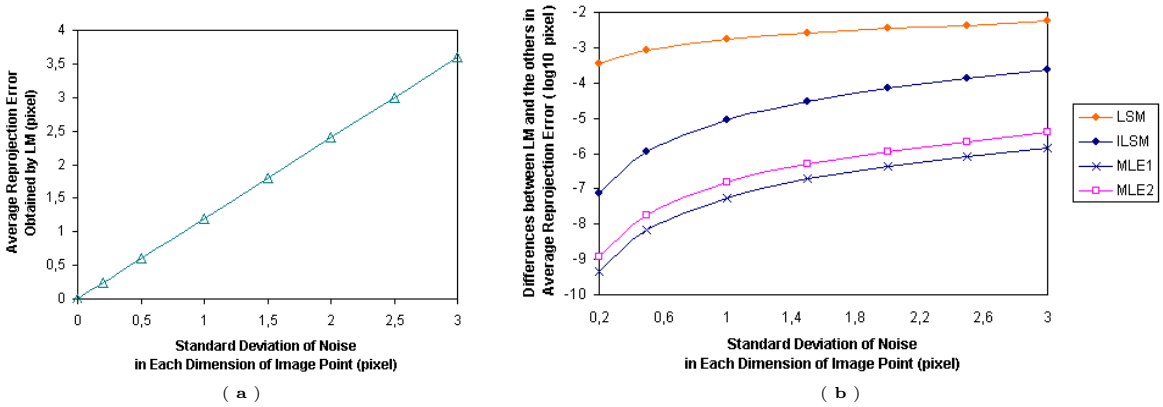


Figure 3.5: Average reprojection error per image point versus noise in each dimension of the image points, when the number of views is 8. (a) Average reprojection error obtained by *LM*; (b) Differences between *LM* and the linear methods in the average reprojection error ($\log_{10} pixel$).

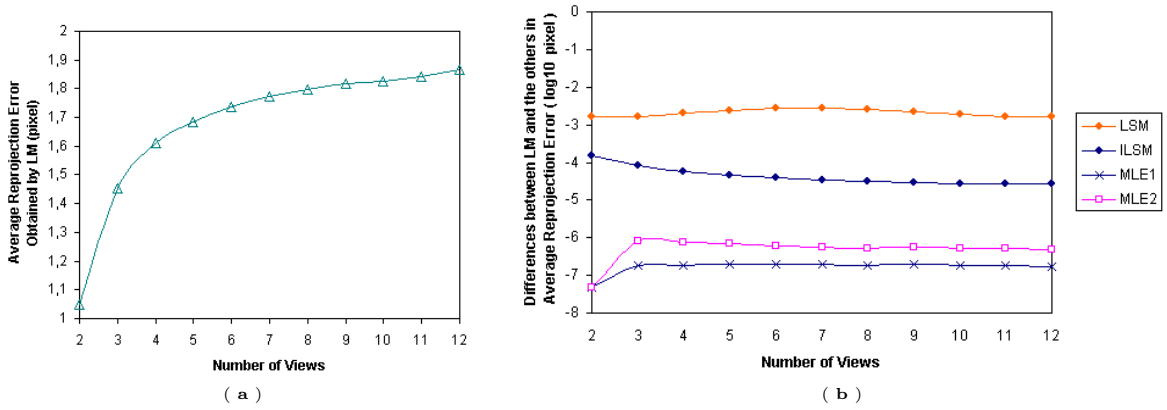


Figure 3.6: Average reprojection error per image point versus number of views, when $\sigma = 1.5$ pixels. (a) Average reprojection error obtained by *LM*; (b) Differences between *LM* and the linear methods in the average reprojection error ($\log_{10} pixel$).

It is easy to understand that the reprojection error increases linearly with the magnitude of the noise. The differences between *LM* and the other methods in the average reprojection error are shown in Fig. 3.5b. Difference between *LSM* and *LM* is the largest. In this experiment, it is more than 10^{-3} pixel when $\sigma > 0.5$ pixel. Then *ILSM* always obtains more accurate results than *LSM*, but still not so accurate as methods *MLE1* and *MLE2*. The results by these two methods are always very similar to each other, and the closest to those by *LM*. When $\sigma < 2.5$ pixels, relative difference between *MLE1* and *LM* is less than 10^{-6}.

Second, we fix the standard deviation to 1.5 pixels, while varying the number of views to compare the obtained average reprojection errors. The average reprojection error obtained by *LM* is shown in Fig. 3.6a, and the differences between it and the others are shown in Fig. 3.6b. Difference between *LM* and *LSM* is still the largest, around 10^{-3} pixel in this experiment. In the case of two views, difference between *LM* and *ILSM* is more than 10^{-4} pixel, and it decreases gradually with the number of the views. Results by *MLE1* and *MLE2* are still very similar to each other, and the closest to those by *LM*. *MLE1* is always a little more accurate than *MLE2*. In this experiment, the difference between *MLE1* and *LM* in the average reprojection error is around 10^{-7} pixel.

Since the ground-truth of 3D points is known for the simulated data, we can

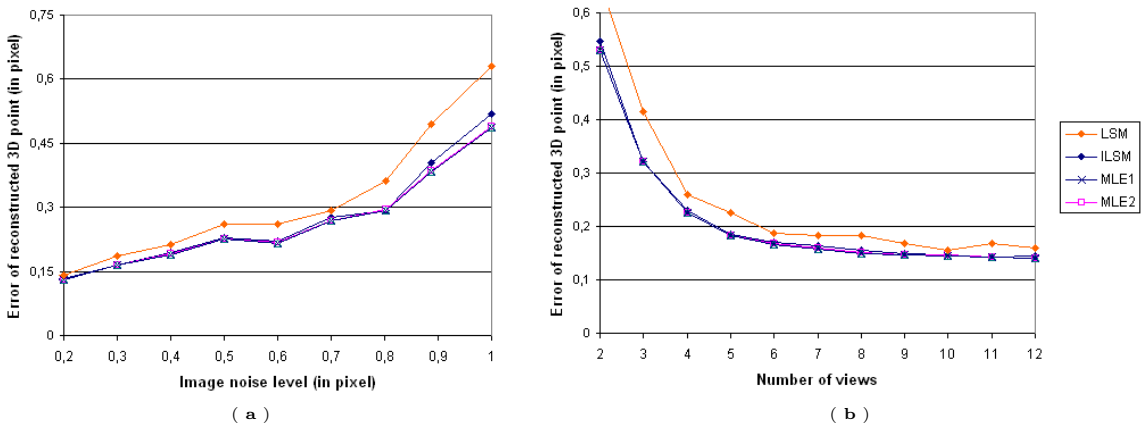


Figure 3.7: Accuracy comparison in term of reconstructed 3D points. (b) Error of reconstructed 3D point versus noise level of image points, in case of 4 views. (a) Error of reconstructed 3D point versus number of views, when $\sigma = 0.5$ pixels.

also compared the accuracy of the 3D points reconstructed with different methods. See Fig. 3.7. It is shown that, when the noise level of the measured image points is small, the four methods *LM*, *ILSM*, *MLE1* and *MLE2* obtain quite close accuracy in term of the reconstructed 3D points, which is significantly higher than that of *LSM*.

Computational cost.

Now we vary the number of views, to compare the average running time for these methods to reconstruct a single space point (see Fig. 3.8). We observe from the figure that the curves of *MLE1* and *MLE2* have relatively steeper gradient than the other methods. But when the number of views is small, these two methods are by far faster than the two iterative methods, *ILSM* and *LM*. From the experiments with real data in the following section, we will find image-point correspondences cross 3, 4 or 5 views are by far more frequent than those cross a large number of views (see Tab. 3.1).

3.5.2 Experiment on Real Data

In this section, two sets of real images are used to evaluate the proposed method *1st-order MLE* with the other methods. One is the well-known Dinosaur36 Sequence from the University of Hannover, and it consists of 36 images that are taken from an artificial dinosaur on a turntable. The other set is a 11-image sequence (Mouse11)

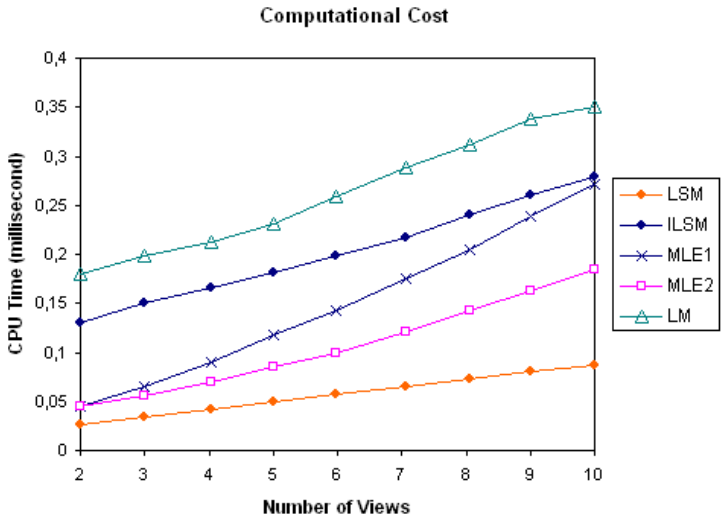


Figure 3.8: CPU time versus the number of views.

of a savings box in shape of two mouses. It was taken by a hand-held camera, and the camera motion is semi-translation.

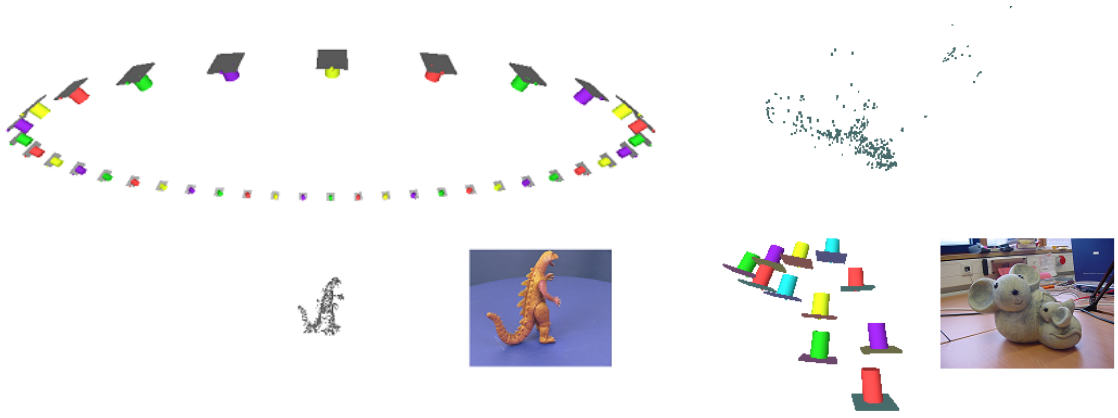
Number of Views	2	3	4	5	6	7	8	≥9
Dinosaur63 (with 1304 correspondences)	0	626	387	189	67	23	11	1
Mouse11 (with 512 correspondences)	9	352	100	39	9	3	0	0

Table 3.1: Counts of image-point correspondences across different number of views.

The configurations of the cameras and 3D points for the two real sequences of images are shown in Fig. 3.9. The feature points are detected and matched between the images using the standard KLT tracker [62], and the counts of image-point correspondences over different number of views are listed in Tab. 3.1. The pose and calibration of the cameras are computed a priori using the technique of bundle adjustment [28]. Assuming that noise occurs only in the measured image points, we reconstruct the 3D points for two sets of real images with the above five methods. The experimental results are listed in Tab. 3.2 and Tab. 3.3.

Generally, as with the experiments over the simulated data, the reprojection error obtained with the optimizer *LM* is still the smallest, then *MLE1*, *MLE2*, *ILSM*, and *LSM* sequentially. As for computational cost, *LSM* is still the fastest, then *MLE2* and *MLE1*. They are about twice as fast as the two iterative methods *LM* and *ILSM*.

Comparing the results of the two sets of real data, we can notice that difference between the five methods upon Dinosaur36 is by far less significant than that upon



(Dinosaur36) (Mouse11)

Figure 3.9: Settings of two real image sequences.

Mouse11 in term of the reprojection error. Especially, *LSM* performs much better for Dinosaur36. This phenomenon has been explained in Sect. 3.2.3. Because the depths of the 3D points on the "Dinosaur" are nearly identical from the optical centers of the views in Dinosaur36, *LSM* can achieve very close results to *ILSM*. For Mouse11, cameras are distributed far or near in front of the 3D scene (see Fig. 3.9), and in such a case the other four methods obtain much smaller residual than *LSM*.

	LSM	*ILSM*	*MLE1*	*MLE2*	*LM*
Residual error (pixel*pixel)	2793.8915	2793.7265	2793.7200	2793.7215	2793.7193
Relative difference from *LM*	$6.2 * 10^{-5}$	$2.6 * 10^{-6}$	$2.5 * 10^{-7}$	$7.9 * 10^{-7}$	0
Running time (milliseconds)	45	138	99	76	182

* Number of 3D points = 1304, number of outliers = 7, $\sigma = 1.5$ pixels, $\alpha\% = 5\%$, $t = 10^{-8}$.

Table 3.2: Experimental results on real Dinosaur images.

	LSM	*ILSM*	*MLE1*	*MLE2*	*LM*
Residual error (pixel*pixel)	771.24	769.45	769.23	769.55	769.06
Relative difference from *LM*	$2.8 * 10^{-3}$	$5.07 * 10^{-4}$	$2.2 * 10^{-4}$	$6.37 * 10^{-4}$	0
Running time (milliseconds)	26	78	53	43	111

* Number of 3D points = 512, number of outliers = 2, $\sigma = 1.5$ pixels, $\alpha\% = 5\%$, $t = 10^{-8}$.

Table 3.3: Experimental results on real Mouse images.

3.5.3 Analysis of the Experimental Results

According to the above experimental results, we can get the following conclusions:

- The iterative numerical optimizer *LM* converges consistently at a best solution, but it is relatively slow compared with the non-iterative methods.
- The direct linear *LSM* is the fastest method. But it has the lowest accuracy, especially when the depth of a space point varies significantly across the views, compared with the magnitude of the depth itself.
- *ILSM* obtains much higher accuracy than *LSM*, but it is still inferior to *MLE1* and *MLE2*, and it is significantly slower than the other two when processing correspondences cross a small number of views.
- *MLE1* and *MLE2* consistently obtain more accurate results than the other two linear methods.
- When the measurement errors of image points are small, the results obtained by *MLE1* and *MLE2* are comparable to those by *LM*, but they are much faster than *LM*.
- When the number of views increases, *MLE2* becomes much faster than *MLE1* with a little loss of accuracy.

Analysis of Reconstruction Accuracy.

When the measurement error of image points is small, the neglected second-order term in the epipolar constraints is trivial, and hence the corrected image points in the method *1st-order MLE* are very close to the true image points. Therefore, the results obtained with methods *MLE1* and *MLE2* are very close to those with *LM*, no matter how many views are taken into account. In the last step of *1st-order MLE*, the 3D point can be reconstructed by two or more corrected image points, and the accuracy increases slightly in direct proportion to the number of corrected image points in use, but the difference is generally insignificant.

ILSM obtains more accurate results than *LSM*, especially when the depths of the space point vary much over the views. It is because in such a case, the weights upon the linear equations in Eq. (3.3) change significantly in the iterations to approach the geometric reprojection error, while the algebraic solution by *LSM* is more inaccurate.

Computational Steps	Number of Multiplications	Number of Additions
$\mathbf{A}_{2N\times 3}, \mathbf{b}_{2N}$	$8N$	$8N$
$\mathbf{A}^\top\mathbf{A}$	$12N$	$12N-6$
$(\mathbf{A}^\top\mathbf{A})^{-1}$	18	8
$\mathbf{A}^\top\mathbf{b}$	$6N$	$6N-3$
$\mathbf{X}=(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$	9	6
SUM	$26N+27$	$26N+5$

Table 3.4: Floating-point operations in *LSM*. (See Sect. 3.2.2.)

Computational Steps		Number of Multiplications	Number of Additions
Initial estimate of $\mathbf{X}_0=(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$ using *LSM*		$26N+27$	$26N+5$
Initial residual		$14N$	$13N-1$
Each iteration (i – id of iteration)	$W_i=diag(w_1,w_1,\cdots w_N,w_N)$	$5N$	$3N$
	$\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i$	$6N$	0
	$\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i\mathbf{A}$	$12N$	$12N-12$
	$(\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i\mathbf{A})^{-1}$	18	8
	$\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i\mathbf{b}$	$6N$	$6N-3$
	$\mathbf{X}_i=(\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{W}_i^\top\mathbf{W}_i\mathbf{b}$	9	6
	reduce of relative residual	$14N+1$	$13N$
SUM (m – number of iterations)		$40N+27+m(43N+28)$	$39N+4+m(34N-1)$

Table 3.5: Floating-point operations of *ILSM*. (See Sect. 3.2.3.)

Analysis of Computational Cost.

In Tab. 3.4-3.7, the floating-point operations (flops) required to compute the five methods are listed out. Comparing the four tables, we see the time complexity of *MLE1* and *MLE2* is $\mathcal{O}(N^3)$, while the others are $\mathcal{O}(N^2)$. This explains why the computational costs of *MLE1* and *MLE2* increase greater with the number of views than the other methods. In Tab. 3.8, the number of floating-point operations is counted for each algorithm to reconstruct a 3D point from an image-point correspondence across 2–8 views, where the number of iterations for the two iterative methods is assumed to be 3. From this table we can see, when the number of views is less than 4, the number of flops required in the two *1st-order MLE* methods is much less than that required by the two iterative methods.

3.6 Conclusions

In this chapter, a new method is proposed to reconstruct a 3D point in space from its projections in multiple views that have known calibration and pose. The method is called *1st-order MLE*. It is linear and non-iterative. First, it converts the reconstruction problem into one of linearly-constrained quadratic optimization through approximating the error model to the first order. Then the first-order geometric correction of the image points is computed. Finally the 3D-point is reconstructed using the corrected image points through linear triangulation.

Computational Steps		Number of Multiplications	Number of Additions
Initial estimate of $\mathbf{X}_0$ using *LSM*		$26N + 27$	$26N + 5$
Initial residual and ϵ_0		$14N$	$13N - 1$
Each iteration (i – id of iteration, J – Jacobian matrix)	$\mathbf{J}_i$	$25N$	$15N$
	$\mathbf{J}_i^\top \mathbf{J}_i$	$12N$	$12N - 6$
	$\mathbf{J}_i^\top \epsilon_{t-1}$	$6N$	$6N - 3$
	adjusted $\mathbf{N}\mathbf{J}_i^\top \mathbf{J}_i$	4	1
	$\mathbf{X}_i$ from $\mathbf{N}\mathbf{J}_i^\top \mathbf{J}_i \mathbf{X}_i = \mathbf{J}_i^\top \epsilon_{t-1}$	22	10
	reduce of relative residual and ϵ_i	$14N + 1$	$13N$
	Note, different from Newton optimization, LM optimization need to conduct the last three steps for more than once in each iteration, say l_i times ($l_i \geq 1$).		
SUM (m – number of iterations)		Multiplications: $40N + 27 + \sum_{i=1}^{m} [43N + l_i(14N + 27)]$	
		Additions: $39N + 4 + \sum_{i=1}^{m} [33N - 4 + l_i(13N + 11)]$	

Table 3.6: Floating-point operations of *LM*.

Computational Steps		Number of Multiplications	Number of Additions
$\mathbf{H}_{2N\times(2N-3)}, \mathbf{e}_{2N-3}$		$16N - 24$	$8N - 12$
$\mathbf{H}^\top\mathbf{H}$		$2N^2 + 4N - 12$	$N^2 + 2N - 6$
$(\mathbf{H}^\top\mathbf{H})^{-1}$, $(k = 2N - 3)$		$(k^3 + k^2)/2$	$(k^3 - 2k^2 + 3k - k\log_2^k)/2$
$(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{e}$		$4N^2 - 12N + 9$	$4N^2 - 14N + 12$
$\Delta\mathbf{x} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{e}$		$8N - 12$	$6N - 12$
$\mathbf{X}$ from $\mathbf{x} - \Delta\mathbf{x}$ using *LSM*	*MLE1*	$26N + 27$	$28N + 5$
	MLE2	79	61
SUM	*MLE1*	$4N^3 - 10N^2 + 63N - 21$	$4N^3 - 17N^2 + 72N - 40 - (2N - 3)\log_2^{2N-3}/2$
	MLE2	$4N^3 - 10N^2 + 37N + 31$	$4N^3 - 17N^2 + 44N + 16 - (2N - 3)\log_2^{2N-3}/2$

Table 3.7: Floating-point operations in *1st-order MLE*.

N		*LSM*	*ILSM* ($m = 3$)	*LM* ($m = 3$, $l_i = 2$)	*MLE1*	*MLE2*
2	Multiplications	79	$107 + 114m = 449$	$107 + \sum_{i=1}^{m}(86 + 55l_i) = 695$	97	97
	Additions	57	$82 + 67m = 238$	$82 + \sum_{i=1}^{m}(62 + 37l_i) = 490$	68	68
3	Multiplications	105	$147 + 157m = 618$	$147 + \sum_{i=1}^{m}(129 + 69l_i) = 948$	186	160
	Additions	83	$121 + 101m = 424$	$121 + \sum_{i=1}^{m}(95 + 50l_i) = 706$	129	101
4	Multiplications	131	$187 + 200m = 787$	$187 + \sum_{i=1}^{m}(172 + 83l_i) = 1201$	327	275
	Additions	109	$160 + 135m = 565$	$160 + \sum_{i=1}^{m}(128 + 63l_i) = 922$	227	171
5	Multiplications	157	$227 + 243m = 956$	$227 + \sum_{i=1}^{m}(215 + 97l_i) = 1454$	544	466
	Additions	135	$199 + 169m = 706$	$199 + \sum_{i=1}^{m}(161 + 76l_i) = 1138$	386	302
6	Multiplications	183	$267 + 286m = 1125$	$267 + \sum_{i=1}^{m}(258 + 111l_i) = 1707$	861	757
	Additions	161	$238 + 203m = 847$	$238 + \sum_{i=1}^{m}(194 + 89l_i) = 1554$	630	518
7	Multiplications	209	$307 + 329m = 1294$	$307 + \sum_{i=1}^{m}(301 + 125l_i) = 1960$	1302	1172
	Additions	187	$277 + 237m = 988$	$277 + \sum_{i=1}^{m}(227 + 102l_i) = 1570$	984	844
8	Multiplications	235	$347 + 372m = 1463$	$347 + \sum_{i=1}^{m}(344 + 139l_i) = 2213$	1891	1735
	Additions	213	$316 + 271m = 1129$	$316 + \sum_{i=1}^{m}(260 + 115l_i) = 1786$	1472	1304

Table 3.8: Comparison of floating-point operations in *1st-order MLE*.

A series of experiments have been presented both with simulated data and with real data, which show the proposed method obtains consistently higher accuracy than the currently widely-used Least-Squares Method (*LSM*) and Iterative Least-Squares Method (*ILSM*). When the measurement errors of the image points are relatively small, the results obtained by *1st-order MLE* are comparable to those of the Levenberg-Marquardt optimizer. However, it is much faster than the numerical optimizer. The solution in the presence of a general Gaussian noise distribution is also provided by the proposed *1st-order MLE*, which has never been solved with other linear methods before. When the number of views is small, the proposed method is by far more efficient than the Newton-type optimizers and the Iterative Least-Squares Method. This is a very practical advantage of the *1st-order MLE* method, because in real data image-point correspondences cross 3 views, 4 views, or sometimes 5 views as well are the commonest. Furthermore, solution in the presence of a general Gaussian noise distribution is also provided by *1st-order MLE*.

The *1st-order MLE* method is valid under two assumptions: (1) not all optical centers of the views are coplanar with the 3D point, when the number of views ≥ 3; (2) the measurement error of the image points is relatively small. Fortunately, the first assumption is usually satisfied in the real data; and the outlier threshold could help to diminish the deviation of the proposed method from the *LM* optimizer.

Meanwhile, a generalized iterative least-squares method is proposed as well in this chapter. Unlike the previous *ILSM*, this generalized method works as well in the presence of a general Gaussian noise distribution. But this method only provides another theoretically-provable possibility to solve the reconstruction problem.

As a basic 3D-reconstruction tool, the proposed method *1st-order MLE* is also very useful in the field of motion and structure estimate, such as the incremental multiple-view reconstruction or other applications where numerous 3D points need to be reconstructed for many times.

4 Linear Iterative Least-Squares Method for Estimating the Fundamental Matrix

This Chapter deals with the problem of estimating the fundamental matrix. In Sect. 2.2, we have learnt that the fundamental matrix encapsulates the epipolar geometry between two un-calibrated views. It is independent of the scene structure, and can be computed from image-point matches without a priori knowledge of the internal parameters of the views. Due to these characteristics, the estimation of the fundamental matrix is usually the first and important step in multiple-view reconstruction.

In the last decade a lot of researches have been done on the fundamental matrix [25] [84] [12] [70]. Specific surveys of the estimation criterions and algorithms of the fundamental matrix were given in paper [44] and [81]. The most commonly-used method for estimating the fundamental matrix from a set of image-point matches is the normalized 8-point method as described in Sect 2.2.5. But as an algebraic method, this method has no geometric meaning [25] [28]. Usually a Newton-type optimizer is applied further to improve its result through minimizing the geometric error.

In this chapter, a linear and iterative least-squares method is proposed for estimating the fundamental matrix. It preserves the noise model of the observed image points, e.g. a Gaussian noise distribution. When the noise in the measurement of image points is small, the accuracy of this method is comparable to that of the non-linear Newton-type optimizers. However, the proposed method is much faster than the optimizers.

Moreover, all the previous proposed linear methods for estimating the fundamental matrix are only applied to the situation when the x and y positional errors of the image points are uncorrelated and identically distributed. However, this is rarely the case in the real data. The method proposed in this chapter is covariance-weighted, and can deal with noisy feature correspondences with high degree of directional uncertainty.

This chapter is organized as follows. In Sect. 4.1 several minimization criterions are first reviewed for estimating the fundamental matrix in the presence of an isotropic Gaussian noise distribution, and then a generalized criterion is proposed in the presence of a general Gaussian noise distribution in Sect. 4.2. The proposed iterative least-squares method is presented in Sect. 4.3. Several algorithms are compared in Sect. 4.4 through the experiments on the real image pairs. Sect. 4.5 gives the conclusions of this work.

4.1 Previous Minimization Criterions

As described in Sect. 2.2.2, the fundamental matrix $\mathbf{F}$ has 7 DOF, and each point correspondence $\{\mathbf{x} \leftrightarrow \mathbf{x}'\}$ provides a linear constraint on it by $\tilde{\mathbf{x}}'\mathbf{F}\tilde{\mathbf{x}} = 0$. Therefore, as long as 7 non-collinear point correspondences are available, the fundamental matrix can be computed. When there are more image correspondences available and with noise in their measurement, the goal of the computation is hence to find a rank-2 matrix that best fits a given criterion.

4.1.1 Algebraic Error

We learnt from Sect. 2.2.2 that each noise-free point correspondence $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ satisfies $\tilde{\mathbf{x}}'_i\mathbf{F}\tilde{\mathbf{x}}_i = 0$. Hence, the most direct idea for computing the fundamental matrix is to find the matrix $\mathbf{F}$ that minimizes the algebraic error:

$$\min_{\mathbf{F}} \sum_i (\tilde{\mathbf{x}}'_i\mathbf{F}\tilde{\mathbf{x}}_i)^2.$$

This minimization problem could be solve with a linear non-iterative algorithm, as described in Sect. 2.2.5. Usually, normalization is necessary to be conducted upon the image points in the two images respectively before the computation (see the step 1 in algorithm 2.2).

The advantage of the linear criterion is that it may be solved with a linear non-iterative method, and its disadvantages are (1) that the rank-2 constraint on $\mathbf{F}$ is not taken into consideration during the minimization, (2) and that the algebraic minimization criterion has no geometric meaning.

4.1.2 Symmetric Epipolar Distance

As described in Sect. 2.2.1, a point in one image should lie on the epipolar line of its corresponding point in the other image. However, this is usually not the case for the measured image points due to noise. The distance of a point from the epipolar line of its corresponding image point is termed as the *epipolar distance*. Another criterion is proposed to estimate the fundamental matrix $\mathbf{F}$ through minimizing the epipolar distance in both images for all the point correspondences:

$$\min_{\mathbf{F}} \sum_i d(\mathbf{x}'_i, \mathbf{F}\tilde{\mathbf{x}}_i)^2 + d(\mathbf{x}_i, \mathbf{F}^\top \tilde{\mathbf{x}}'_i)^2 \tag{4.1}$$

where $d(\mathbf{x}'_i, \mathbf{F}\tilde{\mathbf{x}}_i)$ is the distance of an image point $\mathbf{x}'_i$ from the epipolar line $\mathbf{F}\tilde{\mathbf{x}}_i$, and similar is $d(\mathbf{x}_i, \mathbf{F}^\top \tilde{\mathbf{x}}'_i)$. It is provable that the cost function in Eq. 4.2 is equal to

$$\sum_i (\tilde{\mathbf{x}}'_i \mathbf{F} \tilde{\mathbf{x}}_i)^2 \left(\frac{1}{(\mathbf{F}\tilde{\mathbf{x}}_i)_1^2 + (\mathbf{F}\tilde{\mathbf{x}}_i)_2^2} + \frac{1}{(\mathbf{F}^\top \tilde{\mathbf{x}}'_i)_1^2 + (\mathbf{F}^\top \tilde{\mathbf{x}}'_i)_2^2} \right) \tag{4.2}$$

where $(\mathbf{F}\tilde{\mathbf{x}}_i)_k$ refers to the k-th entry of the 3-vector $\mathbf{F}\tilde{\mathbf{x}}_i$, and similar is $(\mathbf{F}^\top \tilde{\mathbf{x}}'_i)_k$.

The symmetric epipolar distance above provides a kind of geometric error for the estimated $\mathbf{F}$, however, it does not really follow the noise model of the measured image points. Moreover, it works only under the assumption of an isotropic Gaussian noise distribution.

4.1.3 Standard Geometric Error (Reprojection Error)

In Sect. 2.2.5, we reviewed the gold-standard algorithm 2.2 (*p* 22) for estimating the fundamental matrix $\mathbf{F}$. In the algorithm, $\mathbf{F}$ is estimated through minimizing the reprojection error of the matched image points

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \tag{4.3}$$

subject to the epipolar constraints $\hat{\tilde{\mathbf{x}}}_i'^\top \mathbf{F} \hat{\tilde{\mathbf{x}}}_i = 0$ for each pair of matched image points $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$. This function is called the *standard geometric error* or *standard geometric cost function*, since it follows the noise model of the image-point measurement.

In the presence of a Gaussian noise distribution, the estimated image points $\hat{\mathbf{x}}_i$ that meet the above minimization criterion are the *Maximum Likelihood Estimates* for the true values of the observed image points x_i; and the matrix $\mathbf{F}$ is the MLE of the true fundamental matrix.

However, as described in the algorithm 2.2, the above non-linear minimization is usually conducted over $3n + 12$ variables: $3n$ for the n 3D points $\mathbf{X}_i$ and 12 for

the second camera matrix $\mathbf{P}'$. This is a large-scaled sparse optimization problem, compared with the minimizations in Eq. 4.2 and Eq. 4.4 with only 7 parameters of the 7-DOF fundamental matrix $\mathbf{F}$.

4.1.4 First-Order Geometric Error (Sampson Distance)

When an isotropic Gaussian noise distribution can be assumed for the measurement of the image points, the distance $d(*,*)$ in Eq. 4.3 represents the Euclidean distance. Neglecting the second-order term in the epipolar constraint $\tilde{\hat{\mathbf{x}}}_i'^{\top}\mathbf{F}\tilde{\hat{\mathbf{x}}}_i = 0$, the geometric error in Eq. 4.3 is computed as

$$\sum_i e_i^2 ({\mathbf{h}_i}^{\top}\mathbf{h}_i)^{-1}, \tag{4.4}$$

where $\mathbf{h}_i$ is the 4-vector $[(\mathbf{F}\tilde{\mathbf{x}}_i)_1, (\mathbf{F}\tilde{\mathbf{x}}_i)_2, (\mathbf{F}^{\top}\tilde{\mathbf{x}}_i')_1, (\mathbf{F}^{\top}\tilde{\mathbf{x}}_i')_2]^{\top}$, and e_i is the scalar $\tilde{\mathbf{x}}_i'^{\top}\mathbf{F}\tilde{\mathbf{x}}_i$. The detailed deduction can be found in Sect. 3.4.1. This first-order geometric error is called *Sampson distance* in book [28]. In paper [44], the same minimization criterion is deduced as the *non-linear gradient criterion* for estimating the fundamental matrix.

Compared with the cost function in Eq. 4.2, the minimization criterion in Eq. 4.4 fits the noise model of the observed image points, and hence provides better results than the former [81].

However, the 1st-order approximation to the geometric error in Eq. 4.4 is correct only when the x and y positional errors of the image points are uncorrelated and identically distributed, i.e. an isotropic Gaussian noise distribution is assumed.

4.2 Proposed Minimization Criterion — Generalized First-Order Geometric Error

In Sect. 3.4.3 we learnt, in the presence of a general Gaussian noise distribution, the first-order geometric correction to a pair of matched image points $\{\mathbf{x} \leftrightarrow \mathbf{x}'\}$ in two views with known fundamental matrix $\mathbf{F}$ may be computed as

$$[\Delta\mathbf{x}^{\top}, \Delta\mathbf{x}'^{\top}]^{\top} = e(\mathbf{h}^{\top}\boldsymbol{\Sigma}\mathbf{h})^{-1}\mathbf{h}, \tag{4.5}$$

and the reprojection error is

$$e^2(\mathbf{h}^{\top}\boldsymbol{\Sigma}\mathbf{h})^{-1}, \tag{4.6}$$

where $\boldsymbol{\Sigma}$ is the 4×4 covariance matrix of the measurement noise for the image points in the two 2D views.

When the fundamental matrix must be estimated, the sum of the reprojection error in Eq. 4.6 for n (≥ 8) point matches can be used as the cost function as well. In other words, we seek a rank-2 3×3-matrix $\mathbf{F}$ that minimizes the total first-order geometric error

$$\mathcal{J} = \sum_{i=1}^{n} e_i^2 (\mathbf{h}_i^\top \boldsymbol{\Sigma} \mathbf{h}_i)^{-1}. \tag{4.7}$$

This cost function is the general form of Eq. 4.4, in the presence of a general Gaussian noise distribution. The cost function in Eq. 4.4 is in fact a special case of Eq. 4.6, which is valid only in the presence of an isotropic Gaussian noise distribution, i.e. the covariance matrix $\boldsymbol{\Sigma}$ is an identity matrix.

Because of the first-order approximation, the minimization criterions in Eq. 4.4 and 4.7 are valid under the assumption that the measurement error of the image points is small. Experiments in Sect. 3.5.1 show, when the standard deviation of the noise is around 2 pixels, the average reprojection error obtained with the Sampson approximation is different from that obtained with maximum likelihood estimation by around 10^{-8} pixel. Therefore, the minimization criterion in Eq. 4.7 can provide us reliable estimate of the fundamental matrix.

The method proposed in the next section for estimating the fundamental matrix is based on the minimization criterion given in Eq. 4.7.

4.3 Proposed Linear Iterative Least-Squares Method

Instead of using the traditional non-linear Newton-type optimizers, such as Levenberg-Marquardt optimizer, this section proposes an iterative but linear method to estimate the fundamental matrix that meets the minimization criterion given in Eq. 4.7. It converts the problem into one of linear least-squares, and then solves the least-squares problem iteratively.

4.3.1 Least-Squares Expression for the Minimization Criterion

From the commonly-used 8-point algorithm [25], [28], we know the function $\tilde{\mathbf{x}}_i'^\top \mathbf{F} \tilde{\mathbf{x}}_i$ can be written as $\mathbf{a}_i^\top \mathbf{f}$, where the 9-vector $\mathbf{a}_i = [x_i' x_i,\ x_i' y_i, x_i',\ y_i' x_i, y_i' y_i, y_i',\ x_i, y_i, 1]^\top$ and $\mathbf{f}$ is the 9-vector consisting of the entries of $\mathbf{F}$ in the row-major order. Note that $\mathbf{x}_i = [x_i, y_i]^\top$ and $\mathbf{x}_i' = [x_i', y_i']^\top$. Through this expression we can rewrite the cost function in Eq. 4.7 by

$$\mathcal{J} = \sum_{i=1}^{n} (w_i \mathbf{a}_i^\top \mathbf{f})^2 \tag{4.8}$$

where $w_i = (\mathbf{h}_i^\top \Sigma \mathbf{h}_i)^{-\frac{1}{2}}$. We may define an $(n{\times}9)$-matrix $\mathbf{B} = [w_1\mathbf{a}_1, w_2\mathbf{a}_2, \cdots, w_n\mathbf{a}_n]^\top$, and substitute it into Eq. 4.8, which yields

$$\mathcal{J} = \|\mathbf{B}\mathbf{f}\|^2 \tag{4.9}$$

where the $\|\cdot\|$ represents the 2-norm of a vector. Now the minimization problem of Eq. 4.7 is converted into a least-squares problem.

4.3.2 Iterative Solution to the Least-Squares Problem

Now the problem in Eq. 4.9 is that the weights w_i in the rows of matrix $\mathbf{B}$ are unknown. Iterative least-squares method is usually used to solve such problems [7], [27]. The proposed method in this section also belongs to this category.

First, w_i is given an initial estimate, e.g. $w_i = 1$, for $i = 1, 2, \ldots, n$, and then $\mathbf{f}$ can be computed as the unit singular vector corresponding to the smallest singular value of the matrix $\mathbf{B}$ using *Singular Value Decomposition* (*SVD*) [19]. This process is iterated while w_i is re-estimated in each iteration by $w_i = (\mathbf{h}_i^\top \boldsymbol{\Sigma} \mathbf{h}_i)^{-\frac{1}{2}}$ using the new estimated $\mathbf{f}$.

Additionally, because $\mathbf{F}$ is a rank-2 matrix, the obtained $\mathbf{F}$ in each iteration must be replaced by the closest singular matrix to $\mathbf{F}$ using *SVD* (see Sect. 2.2.5).

4.3.3 Normalized Linear Iterative Least-Squares Method

As with the *8-point algorithm*, it is also necessary for the proposed linear iterative least-squares method to normalize the image points in both images, respectively. The normalized linear iterative least-squares algorithm is proposed as follows.

(i) **Normalization:** Transform the image coordinates according to $\tilde{\mathbf{x}}_{Ni} = \mathbf{T}\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_{Ni} = \mathbf{T}'\tilde{\mathbf{x}}'_i$, where 3×3-matrices $\mathbf{T}$ and $\mathbf{T}'$ are normalizing transformations that translate the origin of the coordinates to the centroid of the reference points and scale the root-mean-square (*RMS*) distance of the points from the origin to be $\sqrt{2}$.

(ii) **Initialize $\mathbf{B}_N$:** As defined in Sect. 4.3.1, $\mathbf{B}_N$ is computed using the matches $\{\mathbf{x}_{Ni} \leftrightarrow \mathbf{x}'_{Ni}\}$, with $w_i = 1$, for $i = 1, 2, \ldots, n$.

(iv) **Iterative estimation of the fundamental matrix $\mathbf{F}_N$** corresponding to the matches $\{\mathbf{x}_{Ni} \leftrightarrow \mathbf{x}'_{Ni}\}$

  (a) **Linear least-squares solution:** Compute $\mathbf{F}'_N$ by the singular vector corresponding to the smallest singular value of $\mathbf{B}_N$.

(b) **Enforce rank constraint:** Replace $\mathbf{F}'_N$ by the closest rank-2 matrix $\mathbf{F}_N$ using *SVD*.

(c) **Update $\mathbf{B}_N$:** The weights w_i in $\mathbf{B}_N$ are re-estimated by $(\mathbf{h}_{Ni}^\top \boldsymbol{\Sigma} \mathbf{h}_{Ni})^{-\frac{1}{2}}$ using the new estimated $\mathbf{F}_N$, where

$$\mathbf{h}_{Ni} = \begin{pmatrix} (\mathbf{T}'^\top \mathbf{F}_N \tilde{\mathbf{x}}_{Ni})_1 \\ (\mathbf{T}'^\top \mathbf{F}_N \tilde{\mathbf{x}}_{Ni})_2 \\ (\mathbf{T}^\top \mathbf{F}_N^\top \tilde{\mathbf{x}}'_{Ni})_1 \\ (\mathbf{T}^\top \mathbf{F}_N^\top \tilde{\mathbf{x}}'_{Ni})_2 \end{pmatrix} \tag{4.10}$$

(d) **Compute residual:** The residual is computed as $\|\mathbf{B}_N \mathbf{f}_N\|^2$, where $\mathbf{f}_N$ is the 9-vector made up of the entries of $\mathbf{F}_N$ in the row-major order.

(e) Repeat (a)-(d) till the convergency in the residual.

(iv) **Denormalization:** Set $\mathbf{F} = \mathbf{T}'^\top \mathbf{F}_N \mathbf{T}$. Matrix $\mathbf{F}$ is the estimated fundamental matrix corresponding to the original matches $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$.

4.4 Experiments

In this section, four algorithms of computing fundamental matrices are compared. The algorithms are

LSM. Least-Squares Method or the normalized 8-point algorithm.

ILSM. The normalized Iterative Least-Squares Method as proposed in Sect. 4.3.3.

LM. An advanced Levenberg-Marquardt optimizer [37], in which the 1st-order geometric cost function in Eq. 4.7 is minimized. The fundamental matrix is parameterized with 7 variables. The initial estimate of the fundamental matrix is provided by method *LSM*.

SparseLM. Sparse Levenberg-Marquardt optimizer or the golden standard algorithm (see algorithm 2.2), in which the geometric cost function as given in Eq. 4.3 is minimized. There are totally $3n + 12$ variables involved in the minimization. The initial estimate is also provided by the normalized 8-point algorithm.

The algorithms are tested with two different pairs of images, as shown in Fig. 4.1. 249 point matches are tracked for the pair of table images, and 696 matches for the image pair of the corridor scene.

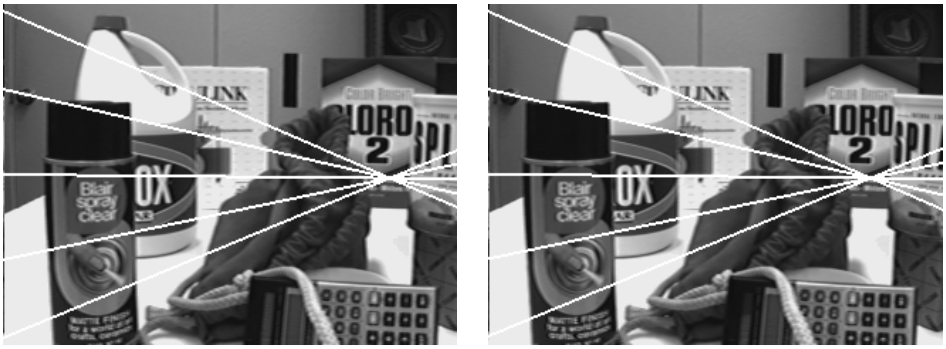


Table images

Corridor scene

Figure 4.1: **Two real image pairs are used to evaluate the algorithms.** *The lines in the images are the corresponding epipolar lines.*

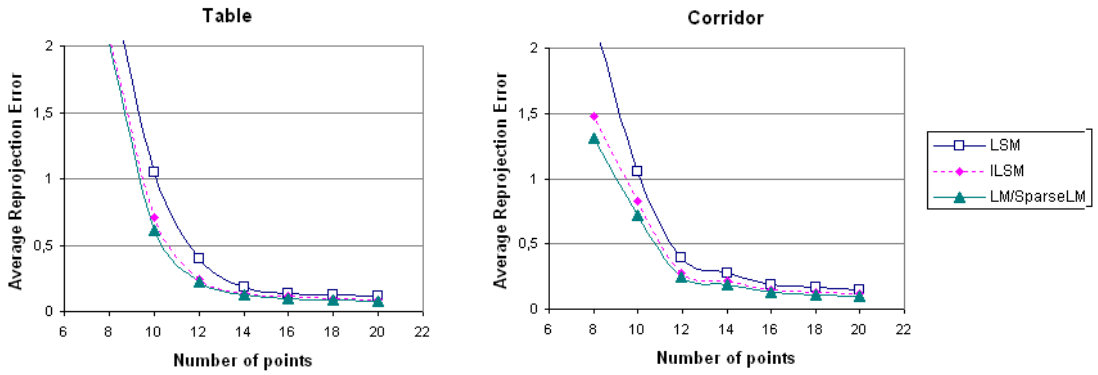


Figure 4.2: **Average reprojection error versus the number of image-point matches.** *The four algorithms of estimating the fundamental matrix are compared in this graph with the two pairs of images as shown in Fig. 4.1. The two optimizers* LM *and* SparseLM *obtain very close results in the experiment and hence are represented by a same curve in the graph. It is shown that in most cases the results of* ILSM *are almost indistinguishable from those of* LM*. Both of them are noticeably better than the non-iterative normalized 8-point algorithm (*LSM*).*

The experimental procedure is as follows. For each pair of images, a number n of matched image points are chosen randomly from the tracked matches and used to estimate the fundamental matrix. The average reprojection error (see below) is computed and compared. The experiment were repeated 20 times for each pair of images and each value of n. This gives an idea of how the different algorithms behave as the number of points is increased. The *average reprojection error* in this section is defined as

$$\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$$

where $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ are the estimated image-point matches subject to the epipolar constraints $\tilde{\hat{\mathbf{x}}}_i'^{\top}\mathbf{F}\tilde{\hat{\mathbf{x}}}_i = 0$.

The results of this experiment are shown and explained in Fig. 4.2. It is obvious that the difference between Levenberg-Marquardt optimizer (*LM*) and sparse Levenberg-Marquardt optimizer (*SparseLM*) is very slight. In this experiment, the difference between the two optimizers in terms of the average reprojection error is around 10^{-6}. This experiment shows that the proposed linear iterative least-squares method (*ILSM*) gives essentially indistinguishable results from the numerical optimizers.

	Table (249 matches)			Corridor (696 matches)		
	Average error	CPU time (ms)	Iterations	Average error	CPU time (ms)	Iterations
LSM	0,0618	19	-	0,1046	7	-
ILSM	0,0602	71	8	0,0914	18	5
LM	0,0598	369	5	0,0904	3379	165
SparseLM	0,0598	713	4	0,0904	1062	5

Table 4.1: **Experimental results using all the detected matches.** The fundamental matrices for the two image pairs are estimated with the four algorithms respectively. In this experiment method *LSM* is the fastest, but obtains much larger reprojection error than the other three methods, especially for the image pair of the corridor scene. Relatively, the results of method *ILSM* is very similar to those of methods *LM* and *SparseLM*, but it is by far faster than the two optimizers.

However, the proposed iterative least-squares method is much faster than the Levenberg-Marquardt optimizers. Tab. 4.1 lists out the experimental results obtained with the four algorithms using all the tracked image-point matches. The epipoles and the epipolar lines obtained with *ILSM* are shown in Fig. 4.1.

4.5 Conclusions

This chapter proposed a linear iterative least-squares method (*ILSM*) for estimating the fundamental matrix.

First, a minimization criterion for estimating the fundamental matrix was proposed, in the presence of a general Gaussian noise distribution. It makes use of the first-order approximation to the geometric error, to convert the original geometric cost function (i.e. reprojection error) into one of linear least-squares minimization $\|Bf\|^2$, where f is composed of the entries of the fundamental matrix in the row-major order. Then f can be computed through singular value decomposition [28]. The weights on the rows of B are re-estimated in each iteration using the new estimated f, such that the weighted least-squares approaches the geometric error adaptively.

The proposed *ILSM* preserves the noise model of the observed image points, and can deal with noisy feature correspondences with high degree of directional uncertainty. When the noise is small compared with the measurement, its accuracy is

comparable to that of the non-linear Newton-type optimizers. However, because of its linearity, *ILSM* is by far faster than these non-linear iterative optimizers.

Additionally, it should be noted that, as with the 8-point algorithm, normalization is also important and necessary for the proposed *ILSM*. Moreover, *ILSM* performs well especially for a pair of un-calibrated cameras; but when the pair of cameras are calibrated, *ILSM* usually obtains no better results than the normalized 8-point algorithm, because there are more constraints upon the fundamental matrix. When the cameras are calibrated, in fact it is the essential matrix to be computed, which represents the relative rotation and the translation between the two cameras. Besides the rank-2 constraint, two of the non-zero singular values of the essential matrix must be equal. With the two non-linear constraints, a good convergency is usually difficult to be achieved through the proposed iterative least-squares method.

5 Accelerated Bundle Adjustment

This chapter discusses a central topic in computer vision, *bundle adjustment*. It is one of the main tools used to optimize the estimates of 3D structure and multiple view parameters, e.g. camera pose or/and calibration parameters (See Sect. 2.4.2). Various techniques have been proposed for bundle adjustment. Mostly they are performed using non-linear Newton-type optimizers which are usually slow when handling a large number of points or views. Readers are referred to books [8] [64] for more detailed introduction to bundle adjustment, or [75] for a latest comprehensive survey of the topic.

In this chapter two bundle-adjustment algorithms are proposed. The algorithms are not only tolerant of missing data, but also allow the assignment of individual covariance to each image measurement. Experiments are conducted on both synthetic data and real data to compare the proposed bundle adjustment techniques with other techniques. It is shown that results of the proposed algorithms are consistently as accurate as those obtained with traditional Newton-type optimizers, and they are much faster in computation.

The remainder of this chapter is organized as follows. First, Sect. 5.1 gives an overview to the problem of bundle adjustment. Then, Sect. 5.2 reviews the previous techniques of bundle adjustment. Sect. 5.3 proposes a simplified cost function for multiple-view reconstruction, and solves it using two of the bundle adjustment techniques. Experimental results on both synthetic data and real data are given in Sect. 5.4, to compare the proposed bundle adjustment techniques with other techniques. Sect. 5.5 gives the conclusions of this chapter.

5.1 Problem Statement

Consider the situation where a set of 3D points $\mathbf{X}_j$ is viewed by a set of cameras. Given the observed image points x_j^i (the projection of the j-th 3D point in the i-th camera image) and the initial estimates of the camera matrices $\mathbf{P}^i$ and the 3D points $\mathbf{X}_j$, *bundle adjustment* is to refine the estimates of $\mathbf{P}^i$ and $\mathbf{X}_j$ such that $\mathbf{P}^i\tilde{\mathbf{X}}_j \sim \tilde{\mathbf{x}}_j^i$ for all available $\mathbf{x}_j^i$. In this chapter, the initial estimation of the cameras and the 3D structure is not discussed.

Because the observed image points are usually noisy, the relationships $\mathbf{P}^i\tilde{\mathbf{X}}_j \sim \tilde{\mathbf{x}}_j^i$ will not be satisfied exactly. Therefore, we seek $\mathbf{P}^i$ and $\mathbf{X}_j$ that minimize the objective function, i.e. the total reprojection error

$$\mathcal{J} = \sum_{ij} d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)^2 \tag{5.1}$$

where $\hat{\mathbf{x}}_j^i$ are the estimated image points and $\tilde{\hat{\mathbf{x}}}_j^i \sim \mathbf{P}^i\tilde{\mathbf{X}}_j$. In the presence of a Gaussian noise distribution, $d(\mathbf{x}_i, \hat{\mathbf{x}}_i)$ refers to the Mahalanobis distance between two image points. $\mathbf{P}^i$ and $\mathbf{X}_j$ meeting the above minimization criterion are the *MLE* of the views and the 3D structure, and the according points $\hat{\mathbf{x}}_j^i$ are the *MLE* of the true image points.

As with other optimization problems, a good initialization is required for bundle adjustment. The initialization has been generally covered in Sect. 2.4.2.

5.2 State of the Art

According to the way of minimizing the cost function, I propose to classify the bundle-adjustment techniques into four categories: the joint, the partitioned, the interleaved, and the embedded techniques.

5.2.1 Joint Bundle Adjustment

Joint bundle adjustment optimizes the 3D structure and the view parameters simultaneously. That is, the objective function given in Eq. 5.1 is minimized by varying both the 3D points $\mathbf{X}_j$ and the camera matrices $\mathbf{P}^i$ at the same time.

Sparse Levenberg-Marquardt optimizer is an efficient method to solve the problem of bundle adjustment jointly [28], which takes advantage of the sparse and regular structure of the Jacobian matrix and the Hessian matrix of the objective function. But as with the factorization method reviewed in Sect. 2.4.2, the sparse Levenberg-

Marquardt optimizer is not tolerant of missing data. In other words, it requires each 3D-point to be visible in all the views.

However, in a real image sequence, a 3D point is usually visible in some arbitrary subset of the available views. In such cases, the joint bundle adjustment has to be solved with a classical Newton-type optimizer, such as Levenberg-Marquardt, over $3n + k * m$ parameters (or variables), where n is the number of 3D points, m is the number of the views, and k is the number of the unknown parameters for each view. In the projective reconstruction, each camera has 11 DOF, i.e. $k = 11$ [6]. Actually entities are often over-parameterized to simplify the coding. For example, 12 parameters are usually used to represent a homogeneous camera matrix, i.e. $k = 12$. As m and n increase, this optimization becomes an extremely large-parameterized optimization problem, and the computation is extremely costly and eventually impossible.

5.2.2 Interleaved Bundle Adjustment

Since each point is estimated independently given fixed cameras, and similarly each camera is estimated independently from fixed points, an interleaved technique was proposed to solve the problem of bundle adjustment: *interleaved bundle adjustment*. The *interleaved bundle adjustment* is also called *resection-intersection* [79] [11] [46] [7], which alternates between the two steps of *resection* and *intersection*. *Resection* refers to optimizing each view independently with fixed 3D points $\mathbf{X}_j$, i.e.

$$\min_{P^i} \sum_j d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)^2 \tag{5.2}$$

for $i = 1, \ldots, n$, where $\tilde{\mathbf{x}}_j^i \sim \mathbf{P}^i \tilde{\mathbf{X}}_j$. The other way round, *intersection* is to optimize each 3D-point independently with fixed views $\mathbf{P}^i$, i.e.

$$\min_{\mathbf{X}_j} \sum_i d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)^2 \tag{5.3}$$

for $j = 1, \ldots, m$.

In both of the steps, the same objective function as that in Eq. 5.1 is minimized. According to the reports in [11] [46] [80], this algorithm performs as well as directly optimizing over all the parameters, i.e. the joint bundle adjustment in terms of convergence accuracy. But it should be noted that such an interleaved solution is only an approximation, but not equivalent to the original optimization problem in the full meaning [52]. Additionally, it usually takes more iterations for the interleaved bundle adjustment to converge than the joint bundle adjustment [74], as is also shown through the experiments in Sect. 5.4.

The main advantage of this interleaved algorithm is that, there are by far fewer parameters involved in the individual minimizations (Eq. 5.2 and Eq. 5.3).

5.2.3 Embedded Bundle Adjustment

As with the sparse Levenberg-Marquardt optimizer, *embedded bundle adjustment* also takes advantage of the sparse structure of the Jacobian matrix of the objective function (Eq. 5.1). But it embeds the optimization of the 3D structure in the optimization of the cameras [85], since the unknown 3D points are independent from each other. The problem may be written mathematically as

$$\min_{\mathbf{P}^i} \left(\sum_j \min_{\mathbf{X}_j} \sum_i d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)^2 \right). \tag{5.4}$$

Therefore, a problem of minimization over $3n + k * m$ dimensional space becomes a problem of minimization over $k * m$, and each iteration contains n independent optimizations over 3 structure parameters. The minimization that has usually the computational complexity of n^3 in the number of parameters n is thus considerably reduced by optimization embedding.

Note that the embedded optimization is totally equivalent to the optimization in Eq. 5.1; there is no approximation. This is different from the approximate algorithm of resection-intersection.

5.2.4 Partitioned Bundle Adjustment

Instead of optimizing over all the views or all the points, *partitioned bundle adjustment* divides the data into several sets, bundle adjusts each set separately, and then merge them by resection or triangulation. This technique is similar to the hierarchical reconstruction (see Sect. 2.4.2 *p* 33). Compared with the other three techniques, this method is sub-optimal. When higher accuracy is required, an overall bundle adjustment still has to be conducted after the merging step.

5.3 Proposed Techniques of Bundle Adjustment

In this section, the 1st-order MLE method of 3D-point reconstruction, as proposed in Chapter 3, is applied to speed up both the embedded and the interleaved techniques of bundle adjustment.

First let us review the situation where the 1st-order MLE method of 3D-point reconstruction applies. Consider the situation in which the camera matrices $\mathbf{P}^i$

$(i = 1, \ldots, m)$ are given fixed, and the image points $\mathbf{x}^i$ corresponding to a single point in space are identified in the images. The problem is to seek the optimal 3D point $\mathbf{X}$ that minimizes the cost function $\mathcal{J} = \sum_i d(\mathbf{x}^i, \hat{\mathbf{x}}^i)^2$ where $\tilde{\hat{\mathbf{x}}}^i \sim \mathbf{P}^i\tilde{\mathbf{X}}$, i.e. the back-projected rays of the estimated image points $\hat{\mathbf{x}}^i$ intersect at a single point $\mathbf{X}$ in space.

The above problem may be solved by the first-order MLE method as described in Sect. 3.4. The first-order MLE of the true image points are computed by

$$\hat{\mathbf{x}} = \mathbf{x} - \boldsymbol{\Sigma}\mathbf{H}(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{e} \tag{5.5}$$

where $\mathbf{x} = ({\mathbf{x}^1}^\top, \cdots, {\mathbf{x}^m}^\top)^\top$, $\hat{\mathbf{x}} = ({\hat{\mathbf{x}}^1}^\top, \cdots, {\hat{\mathbf{x}}^m}^\top)^\top$, and $\boldsymbol{\Sigma}$ is the covariance matrix of the measurement error of the image points. The $(2m) \times (2m-3)$ matrix $\mathbf{H}$ and the $2N-3$ vector $\mathbf{e}$ may be computed by the image points $\mathbf{x}^i$ and the fundamental matrices between pairs of the views (see *p* 46-47). The residual of the cost function is

$$\mathcal{J} = \| \Delta\mathbf{x} \|^2_{\boldsymbol{\Sigma}} = \mathbf{e}^\top(\mathbf{H}^\top\boldsymbol{\Sigma}\mathbf{H})^-\mathbf{e}. \tag{5.6}$$

The 3D point $\mathbf{X}$ may be computed with the least-squares method using the corrected image points $\hat{\mathbf{x}}^i$.

The difference in the results between the first-order MLE and the MLE is very slight, which has been compared in Sect. 3.5.

5.3.1 Accelerating the Embedded Bundle Adjustment

In the inner minimization of the *embedded bundle adjustment*, the reprojection error for each 3D point is minimized separately with fixed view parameters. The first-order MLE method applies exactly to such a situation. The result of the inner minimization can be computed as the residual in Eq. 5.6, and the embedded bundle adjustment is converted into the problem of

$$\min_{\mathbf{P}^i} \left(\sum_j \mathbf{e}_j^\top (\mathbf{H}_j^\top \boldsymbol{\Sigma} \mathbf{H}_j)^- \mathbf{e_j} \right). \tag{5.7}$$

When Levenberg-Marquardt optimizer is used to perform the outer minimization, the image-point corrections in Eq. 5.5 may be used as the measurement vector, and the parameter vector is composed of the parameters of the camera matrices.

5.3.2 Accelerating the Interleaved Bundle Adjustment

In the intersection step of the *interleaved bundle adjustment*, the 3D points need to be reconstructed as well with fixed view parameters. As above, the true image

points may be estimated firstly using Eq. 5.5, and then be used to reconstruct the 3D points $\mathbf{X}_j$ with the least-squares method, for $j = 1, \ldots, n$, which are further used as the input of the following resection step.

5.4 Experiments

In this section, experiments with both synthetic and real data are conducted. Five bundle-adjustment algorithms are evaluated, including:

Interleaved LM-LM. Resection-intersection with Levenberg-Marquardt optimization in both steps.

Interleaved LM-MLE. Resection-intersection with Levenberg-Marquardt optimization in the resection step and the proposed 1st-order MLE method in the intersection step.

Embedded LM-LM. Embedded bundle adjustment with Levenberg-Marquardt optimization in both the outer and the inner minimization.

Embedded LM-MLE. Embedded bundle adjustment with Levenberg-Marquardt optimization in the resection step and the proposed 1st-order MLE method in the intersection step.

Sparse-LM. Sparse Levenberg-Marquardt optimizer.

The convergency threshold for all the above methods is set to 10^{-6}. Note that, **Embedded LM-MLE** and **Interleaved LM-LM** are the two algorithms proposed in Sect. 5.3 and Sect. 5.3.2.

5.4.1 Experiments on Synthetic Data

In this section, a virtual environment is set up, in which twelve cameras are posed around facing down at an artificial scene. Each image's size is 512×512 $pixel^2$. 100 random feature points are tracked by all the 12 cameras, i.e. there is no missing data in the sequence of images. The projected image points are corrupted by an additive independent Gaussian noise, with zero mean and standard deviation of $\sigma = 0.5$ pixel. An initial estimate of the twelve projective matrices is achieved with an incremental reconstruction method as a priori knowledge. The five algorithms listed above are used to conduct projective bundle adjustment over the initial estimate.

In this experiment, bundle adjustment is conducted over different numbers of views. Each method is run 20 times, and then the average reprojection error per image point and the average running time by the five bundle-adjustment methods are compared in Fig. 5.1.

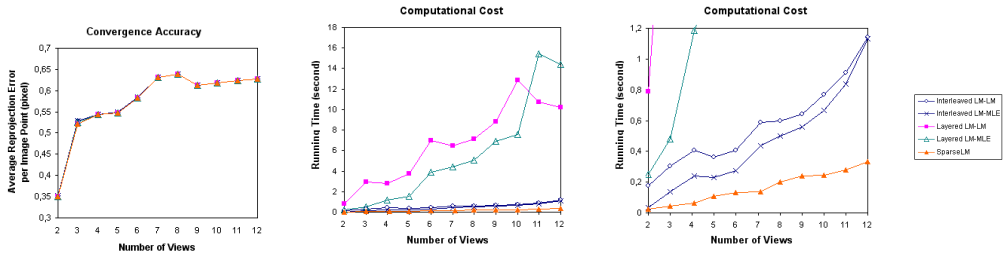
The difference of the five methods in terms of the reprojection error is too subtle to be identified in the Fig. 5.1a. Generally, *embedded LM-LM* and *Sparse LM* achieved the smallest reprojection error. *embedded LM-MLE* obtained a slightly higher reprojection error by less than 10^{-5} pixel, and then *interleaved LM-LM* and *embedded LM-MLE* by around 10^{-3} pixel. The difference between *interleaved LM-LM* and *embedded LM-MLE* is about 10^{-4} pixel in terms of the reprojection error.

The two interleaved bundle-adjustment methods are relatively slower than the other three, because they used more iterations to get converged. See Fig. 5.1bc. *Sparse LM* is the fastest method among the five, but as mentioned before, it is only useful for the no-missing data. Generally *interleaved LM-MLE* is faster than *interleaved LM-LM*, and *embedded LM-MLE* is faster than *embedded LM-LM*, especially when the number of views is small. The experiments show, when dealing with such global-bundle-adjustment problems, the two methods proposed in this chapter (i.e. *embedded LM-MLE* and *interleaved LM-MLE*) are not significantly faster than the two traditional techniques (i.e. *embedded LM-LM* and *interleaved LM-LM*). One reason is that, whenever the parameters of the camera matrices are changed, $n(n-1)/2$ fundamental matrices between each pair of cameras have to be re-computed from the n camera matrices for the proposed methods. However, this disadvantage diminishes when the methods are applied to the *local bundle adjustment* in the incremental multiple-view reconstruction. In that *local bundle adjustment* the estimate of only one view is refined, and accordingly only $n-1$ fundamental matrices are necessarily updated. It will be shown in Chapter 6.

5.4.2 Experiments on Real Data

In this section, bundle adjustment using different methods is conducted upon the well-known Dinosaur Sequence provided by the University of Hannover. There are 37 images in the sequence. They are taken from an artificial dinosaur on a turntable. The inter-frame rotation axis is fixed and the rotation angle is controlled to be 10 degrees. It is a closed sequence, meaning that the first image is the same as the last. 2224 feature points are tracked from the sequence of images, but they are not always visible in all the views.

The intrinsic parameters of the cameras are given a priori, and an initial estimate of camera motions is obtained using an incremental reconstruction method. Then



(a) (b) (c)

Figure 5.1: **Experimental results on the synthetic data ($\sigma = 0.5$).** (a) *average reprojection error versus the number of views;* (b)(c) *running time versus the number of views. In this experiment, the accuracy and the computational cost of the five bundle-adjustment methods are compared with various number of views. The average reprojection errors obtained with the five methods are very close to one another. The* embedded LM-LM *method obtains the smallest reprojection error, then* embedded LM-MLE, interleaved LM-LM *and* interleaved LM-MLE *sequentially. Graphs (b) and (c) represent the same set of data with different scales in the vertical direction, since the two interleaved methods are by far slower than the other three methods.*

the four methods *Embedded LM-LM*, *Embedded LM-MLE*, *Interleaved LM-LM* and *Interleaved LM-MLE* are used to conduct the global metric bundle adjustment over the 37 views. The reconstructed camera motions by the four methods is shown in Fig. 5.2, and the statistical evaluation of the experimental results is listed in Tab. 5.1. The meaning of the statistical items is listed in table. 5.2. The residual, the running time and the number of iterations of the four bundle-adjustment are given in Tab. 5.3.

It is shown through this experiment that the four bundle-adjustment methods got similar results both in terms of the reconstructed 3D structure and motion and in terms of the total reprojection error. Relatively the two embedded methods obtain even more similar results to each other, and so do the two interleaved methods. It also can be seen that it takes much more iterations for the interleaved methods to converge than the embedded methods. Additionally, compared with the experiment with the synthetic data, in this experiment the two proposed approaches *Embedded LM-MLE* and *interleaved LM-MLE* are much faster than the other two approaches, respectively. It is because much more feature points are tracked in the real sequence, i.e. 2224 features versus 100 features in the synthetic sequence. The two proposed approaches gain efficiency especially through the 3D-point reconstruction,

and therefore the more feature points are taken into account, the more efficient are the proposed methods in computation.

	$\overline{\|\mathbf{t}\|}$	$\delta_{\|\mathbf{t}\|}$	$\overline{\mathbf{t}}$	$\delta_{\mathbf{t}}$
Initial Estimate	1.0259	0.0207	1.0227, 0.0112, -0.0784	0.0206, 0.0164, 0.0090
Embedded LM-LM	1.0104	0.0183	1.0072, 0.0110, -0.0769	0.0181, 0.0160, 0.0090
Embedded LM-MLE	1.0104	0.0184	1.0072, 0.0110, -0.0769	0.0181, 0.0160, 0.0090
Interleaved LM-LM	1.0211	0.0189	1.0180, 0.0108, -0.0776	0.0185, 0.0164, 0.0093
Interleaved LM-MLE	1.0212	0.0189	1.0181, 0.0108, -0.0777	0.0186, 0.0164, 0.0093
Standard	1.0000	0.0000	0.9969, 0.0152, -0.0773	0.0000, 0.0000, 0.0000
	$\overline{\alpha}$	δ_{α}	$\overline{\mathbf{r}}$	$\delta_{\mathbf{r}}$
Initial Estimate	10.0365	0.1771	0.0218, 0.9108, 0.4118	0.0147, 0.0040, 0.0090
Embedded LM-LM	10.0177	0.1422	0.0222, 0.9103, 0.4129	0.0144, 0.0047, 0.0104
Embedded LM-MLE	10.0177	0.1423	0.0222, 0.9103, 0.4129	0.0144, 0.0047, 0.0104
Interleaved LM-LM	10.0150	0.1532	0.0225, 0.9103, 0.4129	0.0146, 0.0046, 0.0102
Interleaved LM-MLE	10.0151	0.1532	0.0226, 0.9103, 0.4129	0.0147, 0.0045, 0.0101
Standard	10.0000	0.0000	0.0184, 0.9087, 0.4170	0.0000, 0.0000, 0.0000
	$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
Initial Estimate	0.2948	0.3054	-0.0003, 0.0000	0.2083, 0.3699
Embedded LM-LM	0.2937	0.3034	0.0000, -0.0000	0.2060, 0.3686
Embedded LM-MLE	0.2937	0.3034	0.0000, -0.0000	0.2060, 0.3686
Interleaved LM-LM	0.2940	0.3032	-0.0000, -0.0000	0.2063, 0.3686
Interleaved LM-MLE	0.2941	0.3032	-0.0000, -0.0000	0.2064, 0.3686
Standard	0.2999	0.3249	0.0041, -0.0002	0.2094, 0.3893
	$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
Initial Estimate	0.5465	0.0144	-0.0146, 0.0033, 0.5456	0.0181, 0.0209, 0.0143
Embedded LM-LM	0.2114	0.0104	-0.0036, 0.0032, 0.2108	0.0120, 0.0104, 0.0102
Embedded LM-MLE	0.2114	0.0105	-0.0036, 0.0032, 0.2108	0.0120, 0.0105, 0.0103
Interleaved LM-LM	0.4969	0.0141	-0.0134, 0.0037, 0.4962	0.0174, 0.0189, 0.0142
Interleaved LM-MLE	0.4972	0.0142	-0.0135, 0.0037, 0.4966	0.0173, 0.0191, 0.0141
Standard	0.0000	0.0000	0.0000, 0.0000, 0.0000	0.0000, 0.0000, 0.0000

Table 5.1: **Statistical evaluation of the global bundle-adjustment methods through experiments on the Dinosaur sequence.** *The four bundle-adjustment methods obtains very similar results to the standard configuration of the Dinosaur sequence in terms of the average translation and rotation between the inter-frames. Relatively, the two embedded methods obtain closer results to each other, and so do the two interleave methods. In terms of error in the reconstructed 3D points, the embedded methods are more accurate or much closer to the standard configuration than the interleave methods. Additionally, all the four bundle-adjustment methods obtains smaller reprojection error than the standard configuration. This is because of the noise in the mechanical measurement of the camera motions for the standard configuration.*

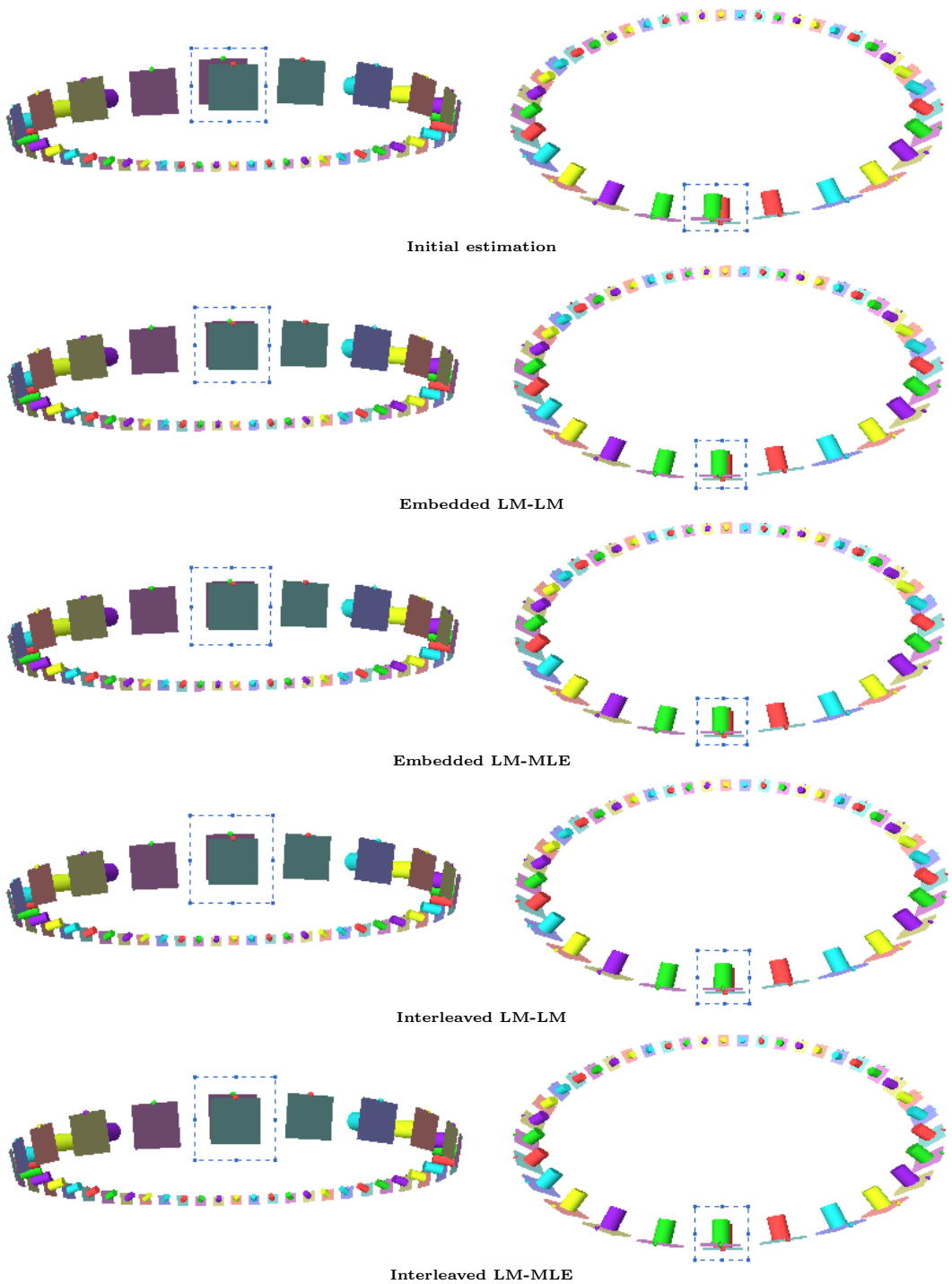


Figure 5.2: **Reconstructed camera motions through global bundle adjustment.** *The above graphs show the frontal (left) and planform (right) of the reconstructed camera motions with respect to the first camera motion. The red and the green cameras inside the dashed rectangles represent the first and the last motions. Actually the two motions are identical, but due to noise they do not coincide through the computation. The top set of graphs is the result of an initial estimation; the other four sets are the refined estimation using the four different bundle-adjustment algorithms. The improvement from the initial estimation to the refined estimation is obvious. But the four refined results are almost indistinguishable from one another. The related statistical evaluation of this experiment is given in Tab. 5.1.*

$\overline{\|\mathbf{t}\|}$	the mean inter-frame translation magnitude
$\delta_{\|\mathbf{t}\|}$	deviation of $\|\mathbf{t}\|$
$\overline{\mathbf{t}}$	the mean inter-frame translation vector
$\delta_{\mathbf{t}}$	deviation of $\mathbf{t}$ in each dimension
$\overline{\alpha}$	the mean inter-frame rotation angle
δ_{α}	deviation of α
$\overline{\mathbf{r}}$	the mean inter-frame rotation axis
$\delta_{\mathbf{r}}$	deviation of $\mathbf{r}$ in each dimension
$\overline{\|\Delta\mathbf{x}\|}$	the mean magnitude of the reprojection error per image point
$\delta_{\|\Delta\mathbf{x}\|}$	deviation of $\|\Delta\mathbf{x}\|$
$\overline{\Delta\mathbf{x}}$	the mean vector of reprojection error per image point
$\delta_{\Delta\mathbf{x}}$	deviation of $\Delta\mathbf{x}$ in each dimension
$\overline{\|\Delta\mathbf{X}\|}$	the mean magnitude of the reconstructed 3D-point error
$\delta_{\|\Delta\mathbf{X}\|}$	deviation of $\|\Delta\mathbf{X}\|$
$\overline{\Delta\mathbf{X}}$	the mean magnitude of the reconstructed 3D-point error
$\delta_{\Delta\mathbf{X}}$	deviation of $\Delta\mathbf{X}$ in each dimension

Table 5.2: **Meaning of the statistical items**

	Residual	Running Time	Number of Iterations
Initial Estimate	1417.0179	-	-
Embedded LM-LM	1402.3779	799	4
Embedded LM-MLE	1402.3780	541	4
Interleaved LM-LM	1403.5408	2559	357
Interleaved LM-MLE	1403.5611	1724	353
Standard	1537.1310	-	-

Table 5.3: **Experimental results of the Dinosaur sequence with different global bundle-adjustment methods.** *In this table, the final residual as computed in Eq. 5.1 and the running time are compared for the four bundle-adjustment methods. The residuals obtained with the two embedded methods are very close to each other and the* Embedded LM-MLE *method proposed in this chapter is much faster than the* Embedded LM-LM *method. Similar rules may be drawn from the table for the two interleaved methods. Additionally, the interleaved methods converge much slower than the embedded methods and also obtain larger residuals. As explained in Tab. 5.1, due to the measurement noise of the camera motions the residual of the standard configuration is larger than those obtained by the reconstruction methods.*

5.5 Conclusions

This chapter proposed two bundle-adjustment approaches. In the approaches, the *1st-order MLE* method of 3D-point reconstruction as proposed in Chapter 3 is applied to simplify the cost function for multiple-view reconstruction, and therefore the computation of the interleaved and the embedded bundle adjustment techniques is partly linearized, whether the cameras are calibrated or un-calibrated. Experiments show the proposed approaches are much faster than the previous bundle-adjustment approaches, at a slight cost of accuracy. Additionally, the proposed bundle adjustment algorithms are not only tolerant of missing data, but also allow the assignment of individual covariance to each image measurement.

6 Integration in Incremental Multiple-View Reconstruction

In Chapter 2, we have reviewed the problem of 3D-scene reconstruction using a set of point matches across a sequence of images, and reviewed the existing techniques used to solve the problem. In this chapter a new incremental multiple-view reconstruction method is proposed, dealing with the problem of 3D reconstruction from long sequences of camera images. It is assumed that feature correspondences (or image-point matches) across the images have been established, and each point feature does not necessarily appear in all the images. The length of a point track (i.e. the path of consecutive images that track a same feature point in space) can be any value equal or larger than 2.

The main idea of the incremental reconstruction is to firstly set up a reference 3D frame using the first two/three images, and then the camera poses for the other views are determined one by one in this frame.

The incremental reconstruction method proposed in this chapter integrates the techniques presented in the previous three chapters. The *ILSM* algorithm as proposed in Chapter 4 is used to estimate the fundamental matrix between the first two images, when the cameras used to take the images are un-calibrated. The *1st-order MLE* method as proposed in Chapter 3 is used to reconstruct the 3D points which are used to initially estimate an additional image using the 3D-2D point correspondences. The embedded bundle-adjustment method as proposed in Chapter 5 is used to conduct the local bundle adjustment, in order to refine the estimate of the additional image. These three linearized or partly-linearized techniques help to improve the efficiency of multiple-view reconstruction. Due to efficiency, the algorithm may take both two-view and multiple-view point matches into consideration.

This is unlike the previous approaches, which rely on point matches only across two or three views. Accordingly, more accurate results can be obtained with the proposed incremental construction technique. Experiments with both simulated data and real data are conducted as a main part of this chapter, to compare the proposed technique with other techniques.

In the the following of this chapter, the proposed reconstruction algorithm is first given in Sect. 6.1. Then experimental results are presented and discussed in Sect. 6.2. At last conclusions are given in Sect. 6.3.

6.1 The Proposed Incremental Reconstruction Algorithm

Given a sequence of images $\{I_i|i = 1, \ldots, N\}$, points of interest are extracted from each image and matched between successive images with the standard feature tracker [62]. The rotations and translations of the images, $\{\mathbf{R}_i, \mathbf{t}_i|i = 1, \ldots, N\}$ are to be estimated, when the calibration matrices of all the images $\{\mathbf{K}_i|i = 1, \ldots, N\}$ are known; or their projection matrices $\{\mathbf{P}_i|i = 1, \ldots, N\}$ are to be estimated, when the images are un-calibrated.

The algorithm of the proposed reconstruction method is given as follows:

1. **Set the world coordinate system.** Choose the camera coordinate system associated with the first image I_1 as the world coordinate system, i.e. $\mathbf{R}_1 = \mathbf{I}_{3\times3}$ and $\mathbf{t}_1 = \mathbf{0}_3$; and $\mathbf{K}_1 = \mathbf{I}_{3\times3}$ if the cameras are un-calibrated.

2. **Estimation of I_2.** Estimate the second image I_2, using a two-view reconstruction technique.

   - **For un-calibrated images.**

     (a) **Initial estimation of $\mathbf{P}_2$.**

     i. **Estimation of $\mathbf{F}_{12}$.** Use the ILSM method as proposed in Chapter 4 to estimate the fundamental matrix $\mathbf{F}_{12}$ between images I_1 and I_2.

     ii. **Computation of $\mathbf{P}_2$ from $\mathbf{F}_{12}$.** Compute the projection matrix $\mathbf{P}_2$ for the second image from $\mathbf{F}_{12}$, with $\mathbf{P}_1 = [\mathbf{I}|\mathbf{0}]$ (See Sect. 2.2.3).

     (b) **Refine the estimate.** Refine the estimate of $\mathbf{P}_2$ through minimizing the reprojection error of the image point matches between

the two images, using the embedded bundle adjustment technique as proposed in Chapter 5, i.e.

$$\min_{\mathbf{P}_2} \left(\sum_{j}^{n_{12}} \mathbf{e}_j^\top (\mathbf{H}_j^\top \mathbf{\Sigma} \mathbf{H}_j)^- \mathbf{e_j} \right),$$

where n_{12} is the number of the point matches between I_1 and I_2. The value of $\mathbf{e}_j^\top (\mathbf{H}_j^\top \mathbf{\Sigma} \mathbf{H}_j)^- \mathbf{e_j}$ is the first-order approximation to the reprojection error of one point match. More explanation of the above function may be found in Sect. 3.4 and Sect. 5.3.

– **For calibrated images.**

(a) **Initial estimation of $\mathbf{R}_2, \mathbf{t}_2$.**

i. **Initial estimation of $\mathbf{F}_{12}$.** The normalized 8-point method as mentioned in Sect. 2.2.5 is used to initially estimate $\mathbf{F}_{12}$.

ii. **Computation of $\mathbf{E}_{12}$ from $\mathbf{F}_{12}$.** From the fundamental matrix $\mathbf{F}_{12}$ and the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, the essential matrix $\mathbf{E}_{12}$ can be computed directly (see Eq. 2.19). Then the singular value constraint upon a valid essential matrix (i.e. two of its three singular values are equal non-zero and one is zero, or see Sect. 2.2.4) needs to be enforced on the matrix $\mathbf{E}_{12}$ using the SVD technique.

iii. **Computation of $\mathbf{R}_2$ and $\mathbf{t}_2$ from $\mathbf{E}_{12}$.** As mentioned in Sect. 2.2.4, four possible motions of the second image can be computed from the essential matrix $\mathbf{E}_{12}$. But only one of them is the correct and reasonable result, which can be found out by some samples of the point matches between the two images.

(b) **Refine the estimate.** As in the above step 2(b) for un-calibrated cameras, refine the initial estimates of $\mathbf{R}_2$ and $\mathbf{t}_2$ using the embedded bundle adjustment technique as proposed in Chapter 5:

$$\min_{\mathbf{R}_2, \mathbf{t}_2} \left(\sum_{j}^{n_{12}} \mathbf{e}_j^\top (\mathbf{H}_j^\top \mathbf{\Sigma} \mathbf{H}_j)^- \mathbf{e_j} \right).$$

3. **3D-point reconstruction for images I_1 and I_2.** Reconstruct the 3D points for the point matches between the first two images I_1 and I_2, using the 1st-order MLE method as proposed in Chapter 3, i.e. Sampson approximation for two views.

4. **Estimation of image I_i for $i \geq 3$.**

(a) **Initial estimation.**

i. **Estimation of $\mathbf{P}_i$.** Estimate the projection matrix $\mathbf{P}_i$ of image I_i using the 3D-2D point matches between the reconstructed 3D points in space and the 2D points in image I_i, using the Gold-Standard algorithm 2.1.

ii. **Computation of $\mathbf{R}_i, \mathbf{t}_i$ for calibrated images.** As in step 2(a)iii, $\mathbf{R}_i$ and $\mathbf{t}_i$ can be computed directly from $\mathbf{P}_i$ and $\mathbf{K}_i$, i.e.

$$(\mathbf{R}_i|\mathbf{t}_i) = \mathbf{K}_i^{-1}\mathbf{P}_i.$$

(b) **Refine the estimate.** Refine the estimate of $\mathbf{P}_i$, or $\mathbf{R}_i$ and $\mathbf{t}_i$, through bundle adjusting the group of images $\{I_1, I_2, \ldots, I_i\}$, using all the point matches shared by image I_i and one or more of the previous images. As above, this local bundle adjustment is conducted using the algorithm proposed in Chapter 5, i.e.

$$\min_{\mathbf{R}_i, \mathbf{t}_i} \left(\sum_j \mathbf{e}_j^\top (\mathbf{H}_j^\top \boldsymbol{\Sigma} \mathbf{H}_j)^- \mathbf{e_j} \right).$$

(c) **3D-point reconstruction.** Reconstruct the 3D points that are tracked by image I_i and one or more of the previous images using the 1st-order MLE method as proposed in Chapter 3.

5. Repeat step 4 while I_i is not the last image in the sequence.

In step 4b, a maximum number of views (or images), say $m \geq 3$, may be set for the local incremental bundle adjustment, since the point matches over a large number of views are usually few. That means, only the image-point matches across m or less views are taken into account for the local bundle adjustment to refine the estimate of an incremental view. Such an incremental reconstruction method is termed as *m-view incremental reconstruction*. Obviously, the computation is faster when m is small. It should be noted that the *m-view incremental reconstruction* is different from *m-view reconstruction*. In the m-view reconstruction there are altogether m views (or images/frames), whereas in the m-view incremental reconstruction there could be more views.

Additionally, when the motion between I_1 and I_2 is small, the estimate of I_2 may not be very accurate, and one can obtain better results through bundle adjusting the first three views, over $\mathbf{P}_2$ and $\mathbf{P}_3$ for un-calibrated images, or over $\mathbf{R}_2$, $\mathbf{t}_2$, $\mathbf{R}_3$ and $\mathbf{t}_3$ for calibrated images.

6.2 Experiments

In order to provide visible and statistical evaluation of the results, the experiments in this section are conducted with calibrated cameras, i.e. the metric reconstruction is performed.

The following three algorithms are tested in the experiments:

Method I. The incremental-reconstruction method proposed in Sect. 6.1.

Method E. A reconstruction method similar to Method I, except that in steps 2b, 4b and 4c, Levenberg-Marquardt optimizers are used to conduct the (inner) optimizations.

Method G. Global bundle adjustment which gives the statistically optimal solution. The embedding optimization technique proposed in Chapter 5 is used to conduct the bundle adjustment, and the result obtained with Method I (5-view incremental reconstruction) is used to initialize the optimization.

The above algorithms have been implemented in C++, and all experiments reported in this section were conducted on a Pentium IV 2.53GHz machine.

6.2.1 The Data

Real data

Two real sequences named Dinosaur37 and Mouse11 were used in the experiments. Dinosaur37 is a sequence of 37 images taken from an artificial dinosaur on a turntable. The inter-frame rotation angle is accurately controlled to be 10 degrees, and accordingly the inter-frame translation is fixed as well. Dinosaur37 is a closed-loop sequence, meaning that the 37th image coincides with the first image (i.e. the motion between them is zero).

Mouse11 is a 11-image sequence of a savings box in shape of two mouses (see Fig. 6.1). The camera motions in this sequence are semi-translation. *Semi-translation* means that the rotation between two consecutive frames is much less significant compared with the translation (see Fig. 6.2). The camera intrinsic parameters for the two image sequences were calibrated a priori, and those for Mouse11 are self-calibrated and thus less accurate. Fig. 6.1 shows the first four images of the two image sequences respectively.

Figure 6.1: **The respective first four images of the image sequences Dinosaur37 and Mouse11.**

Synthetic data

The synthetic data used in the experiments were generated from the above two real sequences of Dinosaur37 and Mouse11. First, 3D points were reconstructed using the real image points and camera motions; then the original image points were replaced by the re-projected 2D points of the reconstructed 3D points. After this replacement, the camera motions and the 3D points in the real data became the *ground truth* for the synthetic data. The re-projected 2D points are further corrupted with different noise levels [10] and used as the synthetic data for the experiments.

6.2.2 Experiments on Synthetic Data

In this section, **Method I** and **Method E** are compared using synthetic data. The two methods are used to conduct the incremental reconstructions, and their results are compared against those of the global bundle adjustment (i.e. **Method G**) and the standard configuration. The *standard configuration* refers to the ground-truth camera motions with the synthetic points. The 3D points and the relative camera motions with respect to the ground truth are used as the standard error metrics for the estimated results. At each image noise level, each method is run 20 times, and the mean differences between computed results and the ground truth were recorded. Average running time was recorded in the same way.

It should be noted that results of the global bundle adjustment are different from

the standard configuration. Global bundle adjustment seeks the 3D reconstruction with the given image-point matches such that the cost function (as given in Eq. 2.37), i.e. the total reprojection error, is minimized. Therefore, the results of the global bundle adjustment are influenced by the random selection of the noise that are added to the ground-truth image point and thus they do not always coincide with the standard configuration.

Additionally, for Dinosaur37 the ground-truth camera motions are measured mechanically through the movement of the turn-table, and the ground-truth 3D points are reconstructed with the real image points and the ground-truth camera motions. Therefore, the ground-truth 3D reconstruction of Dinosaur37 does not necessarily minimize the total reprojection error both for the real data and the synthetic data. For Mouse11 the ground-truth camera motions and 3D points are reconstructed using the real image points through self-calibration, and thus in the experiments with the real Mouse11 the ground-truth reconstruction is the same as that of the global bundle adjustment; but for the synthetic Mouse11, the two reconstructions are different.

Synthetic Dinosaur37

There are totally 2224 feature points tracked from the image sequence Dinosaur37. They are not visible in all the images. The image size is 720×576. Several facts are known for sure, within the control accuracy offered by the turntable. First, the camera movement is on a circle; second, the relative rotation angle and relative translation vector between two consecutive images are the same throughout the sequence, i.e. 10 degrees, and the rotation axis is fixed; third, the camera position of the first image is overlapped with the last one. As described in Sect. 6.2.1, these facts are the ground-truth camera motions for the synthetic data, and the 3D points reconstructed with these motions and the real image points are the ground-truth 3D structures. The real image points are then replaced with the re-projected 2D points of the ground-truth 3D points. Note that the visibility of the features in the images is unchanged after the replacement. At last, seven synthetic Dinosaur37 sequences are generated respectively from the re-projected 2D points corrupted with a Gaussian noise [10] with zero mean and standard deviation $\sigma = 0.1, 0.2, \cdots, 0.7$.

1) Graphical comparisons under different noise levels. Incremental reconstructions are conducted for the synthetic Dinosaur37 sequences with *Method I* and *Method E*, using point correspondences across 3–6 views respectively. Fig. 6.3–6.6 give the graphical comparisons of the experimental results under different noise levels

of image points. The results are compared against those of the bundle adjustment and the standard configuration in different aspects, including the average translation magnitude and rotation angle between every pair of consecutive frames, the average reprojection error per image point, the reconstructed 3D points. The computational time is compared as well.

Generally, the reconstruction errors increase with the level of image noise, especially the reprojection error. In comparison, the curves for the reconstructed camera motions and 3D points are much more irregular. The regularity of the reprojection-error curves and the irregularity of the other curves may be explained with two facts. First the random noise is added upon the image points, but not upon the 3D points or the camera motions; second, the reconstruction is conducted through minimizing the reprojection error, since the errors of 3D points and camera motions are unavailable for real image sequences.

In all the four figures, *Method I* consistently gives almost indistinguishable results from *Method E*. In this experiment, their difference is around 10^{-6} pixel in terms of the average reprojection error per image point, when the noise level is less than 1 pixel. However, *Method I* is nearly 5 times as fast as *Method E*, as shown in Fig. 6.3(f)–6.6(f).

In addition, comparing the four sets of graphs in Fig. 6.3–6.6, the curves are not much different from one another, except those in graphs(d). This is because of the missing data in Dinosaur37. The 2224 tracked feature points are not visible in all the images. The numbers of correspondences across different numbers of views in Dinosaur37 are listed in Tab. 6.1. In this sequence of images there are by far more features across 3 views than those across more views. Therefore, the 3 to 6-view incremental reconstructions give no significantly different results from one another, including the running time. Comparatively, the experiment with the synthetic Mouse11 sequences gives more noticeable comparison (see Fig. 6.10–6.13), since no missing data is assumed for those sequences.

Number of views	2	3	4	5	6	7	8	9	10	>=11
Number of correspondences	0	1482	454	183	66	22	15	1	1	0

Table 6.1: The numbers of correspondences across different numbers of views in Dinosaur37 sequences.

2) Visual and statistical comparisons of 5-view incremental reconstructions. The visual and statistical comparisons between the two incremental-reconstruction methods, *Method I* and *Method E*, are given in Fig. 6.7 and Tab. 6.2. Incremental

reconstructions are conducted for the synthetic Dinosaur37 sequence at noise level of 0.4 pixel, with *Method I* and *Method E* respectively, using the point correspondences across 5 views. The results are compared against those of the bundle adjustment and the standard configuration.

According to the ground truth mentioned earlier, the frames should form a closed circle if the camera motions are computed correctly. This conforms quite good with the results of the two incremental reconstructions and the bundle adjustment, as can be seen from Fig. 6.7. Note that, all the reconstructions conducted in this section for the Dinosaur37 sequence did not use the knowledge that the last frame is the same as the first frame.

The similar performances of *Method I* and *Method E* are shown both through the visual comparison of the reconstructed camera motions in Fig. 6.7 and through the statistical measurements in Tab. 6.2. Readers are referred to Tab. 5.2 for the meaning of the statistical items.

In this experiment, the global bundle adjustment in this experiment does not achieve better results than the incremental-reconstruction methods in terms of the reconstructed camera motions and 3D points, but only the reprojection error is much reduced through the global minimization. It may be explained as in the last experiment. Random noise is added upon the image points, but not upon the 3D points or the camera motions; and the total reprojection error of the image point is aimed to be minimized in the process of reconstruction, since it is impossible to measure the error of 3D points or camera motions directly for a real image sequence. In the experiment with the real Dinosaur37, the global bundle adjustment achieves much better results than the incremental reconstructions (see Fig. 6.17 or Fig. 6.16).

3) Graphical, visual and statistical comparisons of (3–6)-view incremental reconstructions. Comparing the four sets of graphs in Fig. 6.3–6.6, we can observe that the errors of the two incremental reconstructions generally decrease when point correspondences across more views are taken into account. Take for example the synthetic Dinosaur sequence at noise level of 0.4 pixel. The reconstruction error and the computational time for this synthetic sequence are compared in Fig. 6.8. In every respect, the reconstruction accuracy increases with the number of views, though the improvement of the reconstructed 3D points is relatively insignificant; and it is obvious that the running time increases with the number of views for both *Method I* and *Method E*. Still, it is observed that the two incremental-reconstruction methods obtain similar results at very different computational costs.

Fig. 6.9 shows the visual comparison of the estimated camera motions for the synthetic Dinosaur37 sequence at noise level of 0.4 pixel. The motions are reconstructed

with *Method I* using point correspondences across different numbers of views. The slight improvement with the number of views may be observed from the figures. The related statistical evaluation of this experiment is given in Tab. 6.3. It can be seen from the table that, the accuracy of the incremental reconstructions increases with the number of views, both in terms of the reprojection error and in terms of the reconstructed motions and structures. Additionally, as with for results in Fig. 6.7 and Tab. 6.2, the bundle adjustment in this experiment achieves no better results than the incremental-reconstruction methods in terms of camera motions and 3D structure, except that the reprojection error is minimized.

Synthetic Mouse11 (with no missing data)

The ground truth of 3D points and camera motions for the Mouse11 sequence is shown in Fig. 6.2. The image size is 1280×960. Totally 512 feature points are tracked from the sequence. They are not visible in all the images in the real data. But in the synthetic data, they were easily made visible in all the images. That means, there is no missing data in the synthetic Mouse11 sequences. Nine synthetic Mouse11 sequences are generated respectively with a Gaussian noise [10] of zero mean and standard deviation $\sigma = 0.1, 0.2, \cdots, 0.9$.

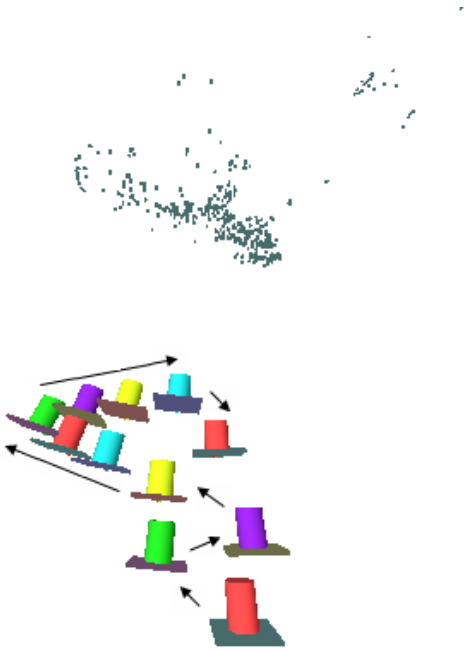


Figure 6.2: **The ground-truth camera motions and 3D points of image sequence Mouse11.**

The ground-truth camera parameters for Mouse11 were calibrated through self-calibration. Metric reconstructions with known intrinsic camera parameters are conducted for the synthetic Mouse11 sequences in the following experiments.

1) Graphical comparisons under different noise levels. In this experiment, reconstructions for the nine synthetic Mouse11 sequences are conducted and the results are compared under different noise levels in Fig. 6.10–6.10. The two incremental-reconstruction methods using point correspondences across different numbers of views are compared respectively with the bundle adjustment and the standard configuration (i.e. the ground-truth camera motions with the synthetic points). In the graphs, the reprojection error, the error of the reconstruction 3D points and the computational cost are compared while the noise level of image points is varied.

Compared with the graphs in Fig. 6.3–6.6, the reconstruction errors in this experiment increase with the level of image noise as well. But the difference between *Method I* and *Method E* in terms of the average reprojection error is relatively larger, around 10^{-3} pixel in this experiment compared with 10^{-6} pixel in the last. Sometimes, *Method I* obtains even smaller errors than *Method E*, e.g. for the synthetic Mouse11 sequence at noise level of 0.2 pixel. However, *Method I* requires only half the running time of *Method E* in general.

2) Graphical comparisons of (3–6)-view incremental reconstructions. In Fig. 6.14 we take the synthetic Mouse11 at noise level of 0.5 pixels for example, to compare the incremental reconstructions using point correspondences across different numbers of views. From the graphs we observe that the reprojection error for the incremental reconstructions drops significantly with the number of views, whereas the error of the reconstructed 3D points does not change much. *Method E* obtains a little better results than *Method I*.

6.2.3 Experiments on Real Data

In this section, the two incremental reconstruction methods, *Method I* and *Method E* are tested with the two real image sequences of Dinosaur37 and Mouse11. As with the synthetic data, (3–6)-view incremental reconstructions are conducted with the two methods and compared with the global bundle adjustment and the standard configuration (for Dinosaur37). Each method is run 20 times with the real image sequences.

Real Dinosaur37

1) Graphical, visual and statistical comparisons of (3–6)-view incremental reconstructions. Fig. 6.15 shows the experimental results for the real Dinosaur37 sequence using the two incremental reconstruction methods and the global bundle adjustment. Compared with the curves in the Fig. 6.8 for the synthetic Dinosaur37, the bundle adjustment for the real data obtained much better results than the incremental reconstructions both in terms of the reprojection error and in terms of the reconstructed 3D structure and camera motions (which can also be seen from the following visual and statistical comparisons). **Method I** still gets nearly the same results as **Method E**, and their reconstruction error decrease with the number of views used for the incremental reconstruction. **Method I** is about 4 times as fast as **Method E**, and more than 20 times faster than the global bundle adjustment. Additionally, compared with the experiment on the synthetic data, all the three reconstruction methods achieved by far smaller reprojection error than the standard configuration. It may be explained by the noise in the mechanical measurement of camera motions for the standard configuration.

Fig. 6.16 shows the camera motions reconstructed with *Method I* using point correspondences across different numbers of views. Compared with Fig. 6.8, the improvement in accuracy with the number of views are relatively more significant in Fig. 6.16. The related statistical evaluation of the results are listed in Tab. 6.4.

2) Visual and statistical comparisons of two 5-view incremental reconstructions. Fig. 6.17 and Tab. 6.5 show respectively the visual and statistical comparisons of the experimental results for the real Dinosaur37, using the two 5-view-incremental-reconstruction methods. We observe again that the difference between **Method I** and **Method E** is very slight, both in terms of reconstructed motions and structures and in terms of the reprojection error (or the residual of the cost function).

Real Mouse11

In this section, **Method I**, **Method E** and **Method G** are compared using the real Mouse11 sequence. As mentioned earlier, the ground-truth configuration for the real Mouse11 sequence coincides with that of the bundle adjustment, since the intrinsic and extrinsic camera parameters are self-calibrated.

Fig. 6.18 and Tab. 6.6 show the experimental results of the three reconstruction methods for the real Mouse11. Compared with the experiments on the synthetic Mouse11 sequence in Fig. 6.14, **Method I** and **Method E** achieve even closer results for the real data. The difference between them is around 10^{-5} pixel in terms

of the average reprojection error per image point. Additionally, **Method I** is about twice as fast as **Method E**, and more than 5 times faster than the global bundle adjustment, i.e. **Method G**.

6.3 Conclusions

An incremental reconstruction algorithm for long sequences of images is proposed in this chapter. It deals with both the projective reconstruction for un-calibrated cameras and the metric reconstruction for calibrated cameras, given point matches across the images. The algorithm integrates the three techniques proposed in the previous chapters.

With the three techniques, the process of the incremental reconstruction is accelerated, and it allows the computation over point matches across more views. This is unlike the previously-proposed reconstruction approaches [3] [18] [85], which rely on point matches only across two or three views. Experiments both with simulated data and with real data show that the reconstruction accuracy is significantly improved when point matches across more views are taken into account and that the proposed method consistently obtains as accurate results as the classical incremental-reconstruction methods, however the computational cost is much reduced.

	$\overline{\|\mathbf{t}\|}$	$\delta_{\|\mathbf{t}\|}$	$\overline{\mathbf{t}}$	$\delta_{\mathbf{t}}$
Method I	1.0080	0.0185	1.0049, 0.0145, -0.0772	0.0185, 0.0091, 0.0050
Method E	1.0080	0.0186	1.0049, 0.0145, -0.0772	0.0185, 0.0091, 0.0050
Method G	1.0075	0.0098	1.0043, 0.0165, -0.0781	0.0096, 0.0068, 0.0041
Standard	1.0000	0.0000	0.9969, 0.0152, -0.0773	0.0000, 0.0000, 0.0000
	$\overline{\alpha}$	δ_{α}	$\overline{\mathbf{r}}$	$\delta_{\mathbf{r}}$
Method I	10.0113	0.1443	0.0191, 0.9099, 0.4143	0.0084, 0.0025, 0.0056
Method E	10.0112	0.1444	0.0191, 0.9099, 0.4143	0.0084, 0.0025, 0.0056
Method G	9.9855	0.0841	0.0173, 0.9088, 0.4168	0.0061, 0.0017, 0.0037
Standard	10.0000	0.0000	0.0184, 0.9087, 0.4170	0.0000, 0.0000, 0.0000
	$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
Method I	0.3659	0.2083	-0.0001, -0.0000	0.2624, 0.3292
Method E	0.3659	0.2083	-0.0001, -0.0000	0.2624, 0.3292
Method G	0.3641	0.2072	-0.0000, -0.0000	0.2603, 0.3282
Standard	0.3686	0.2097	-0.0003, -0.0001	0.2638, 0.3320
	$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
Method I	0.2270	0.0127	0.0018, -0.0031, 0.2264	0.0127, 0.0101, 0.0127
Method E	0.2273	0.0127	0.0018, -0.0031, 0.2267	0.0127, 0.0101, 0.0127
Method G	0.2474	0.0125	0.0044, -0.0110, 0.2464	0.0145, 0.0128, 0.0123
Standard	0.0108	0.0082	-0.0002, 0.0000, 0.0002	0.0086, 0.0065, 0.0082

Table 6.2: **Statistical evaluation of 5-view incremental reconstructions for synthetic Dinosaur37 (at noise level of 0.4 pixel) with *Method I* and *Method E*.** *Incremental reconstructions are conducted for the synthetic Dinosaur37 with* Method I *and* Method E *respectively using point correspondences 5 views. The results are statistically compared with those of the global bundle adjustment (*Method G*) and the standard configuration in the above table. The related camera motions reconstructed in this experiment is compared in Fig. 6.9. It can be seen from the table that the accuracy of* Method I *and* Method E *are very close to each other in all the statistical respects, and the reprojection error of the two incremental reconstructions is also very close to that obtained with the global bundle adjustment. The global bundle adjustment in this experiment does not achieve better results than the incremental-reconstruction methods in terms of the reconstructed camera motions and 3D points, but the reprojection error is much reduced through it in any way (see the text for the explanation).*

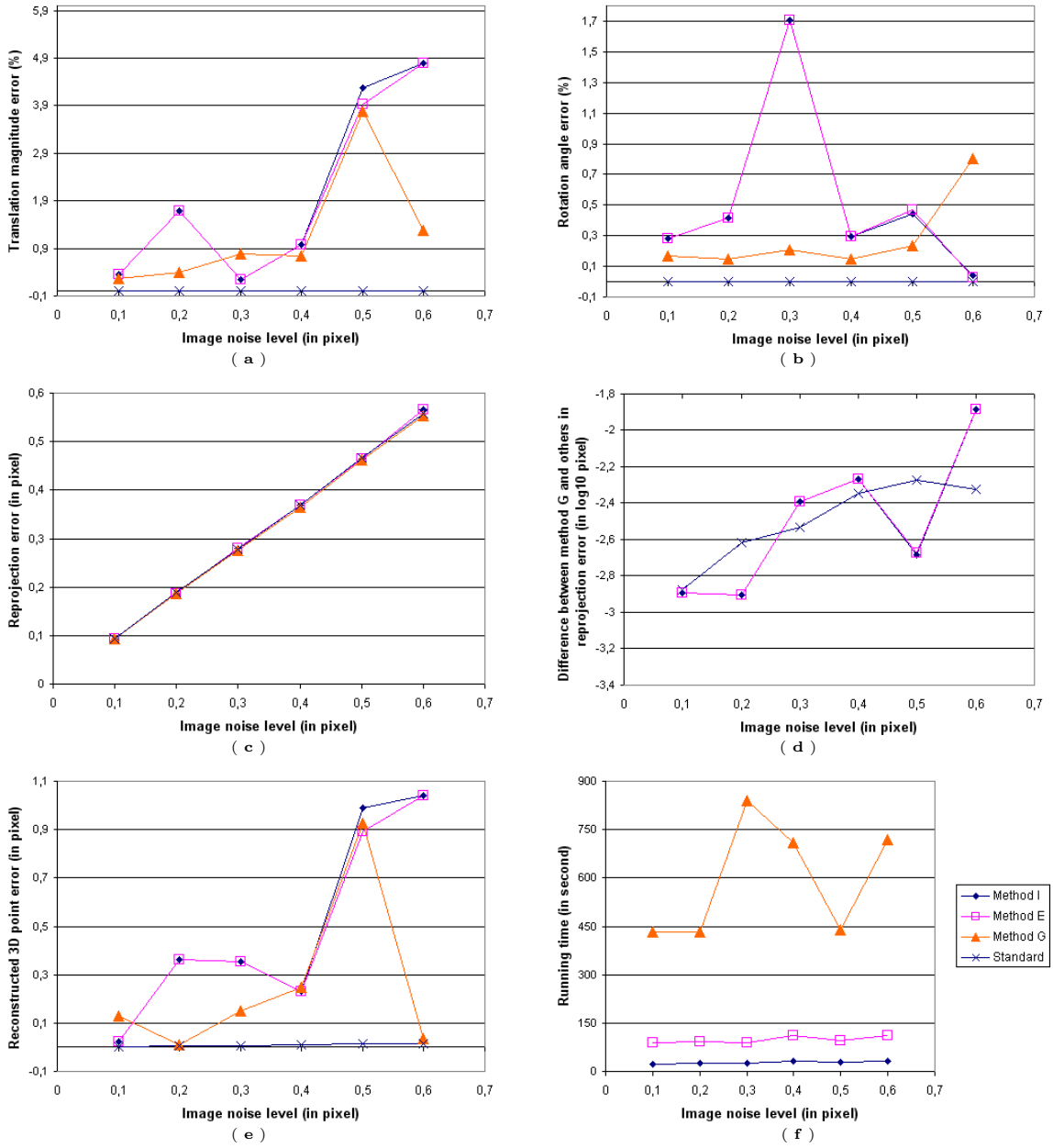


Figure 6.3: **3-view incremental reconstruction for synthetic Dinosaur37.** (a) *Error of the translation magnitude vs. image noise level;* (b) *Error of the rotation angle vs. image noise level;* (c) *Reprojection error vs. image noise level;* (d) *Difference in reprojection error vs. image noise level;* (e) *Error of reconstructed 3D-points vs. image noise level;* (f) *Running time vs. image noise level. Incremental reconstructions are conducted for the synthetic sequences of Dinosaur37 with* Method I *and* Method E *using point correspondences across 3 views. Their results are compared against those of the global bundle adjustment and the standard configuration in the above graphs. We see from the graphs that the reprojection error increases linearly with the level of noise for all the reconstruction methods. In comparison, the curves for the reconstructed camera motions and 3D points are much more irregular, but a general tendency to increase with the noise level still can be observed from the graphs.* Method I *and* Method E *obtain very similar results, but the former is by far faster than the later. Similar conclusions can be drawn from the following graphs in Fig. 6.4, 6.5 and 6.6.*

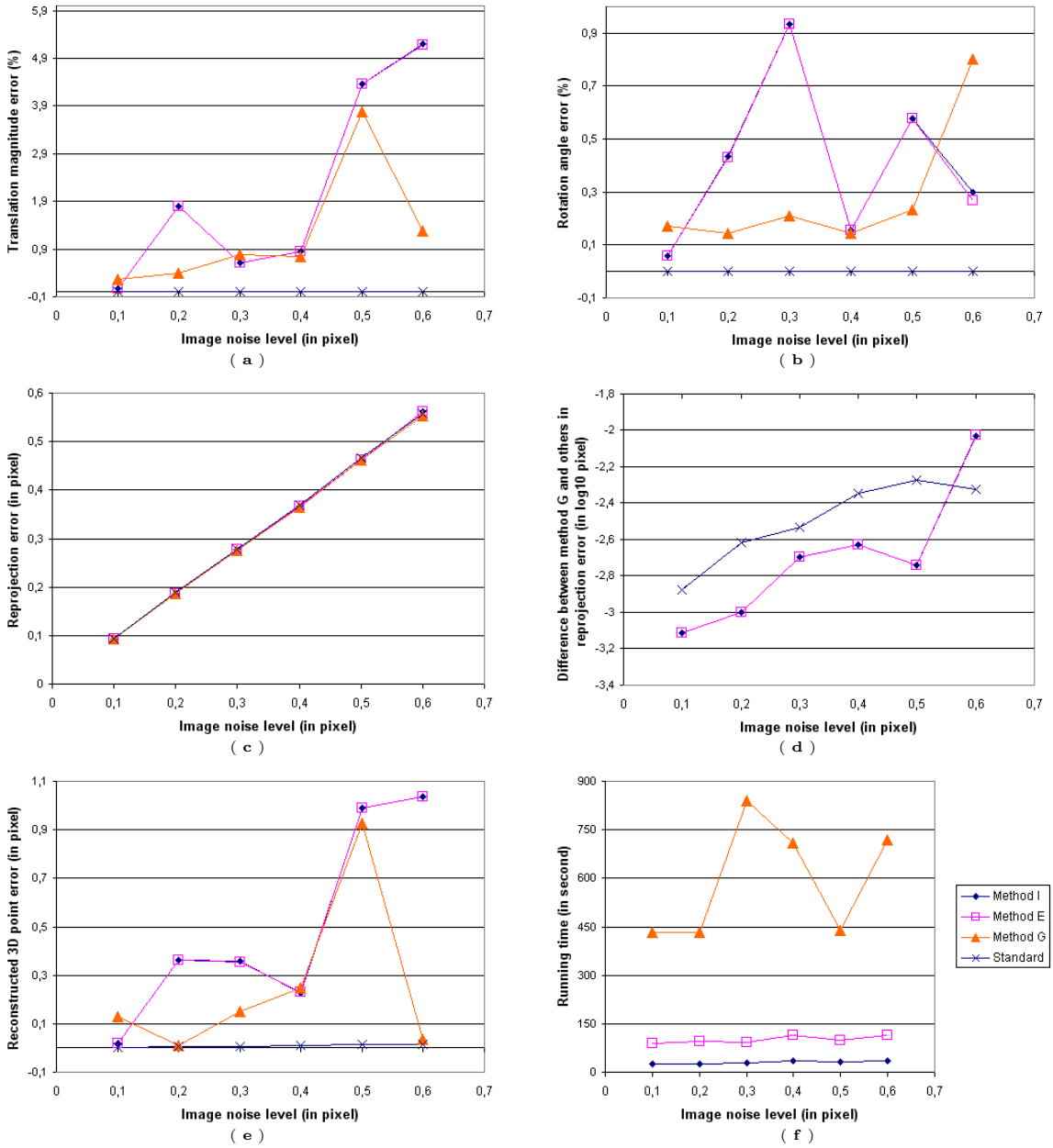


Figure 6.4: **4-view incremental reconstruction for synthetic Dinosaur37.** (a) *Error of the translation magnitude vs. image noise level;* (b) *Error of the rotation angle vs. image noise level;* (c) *Reprojection error vs. image noise level;* (d) *Difference in reprojection error vs. image noise level;* (e) *Error of reconstructed 3D-points vs. image noise level;* (f) *Running time vs. image noise level.*

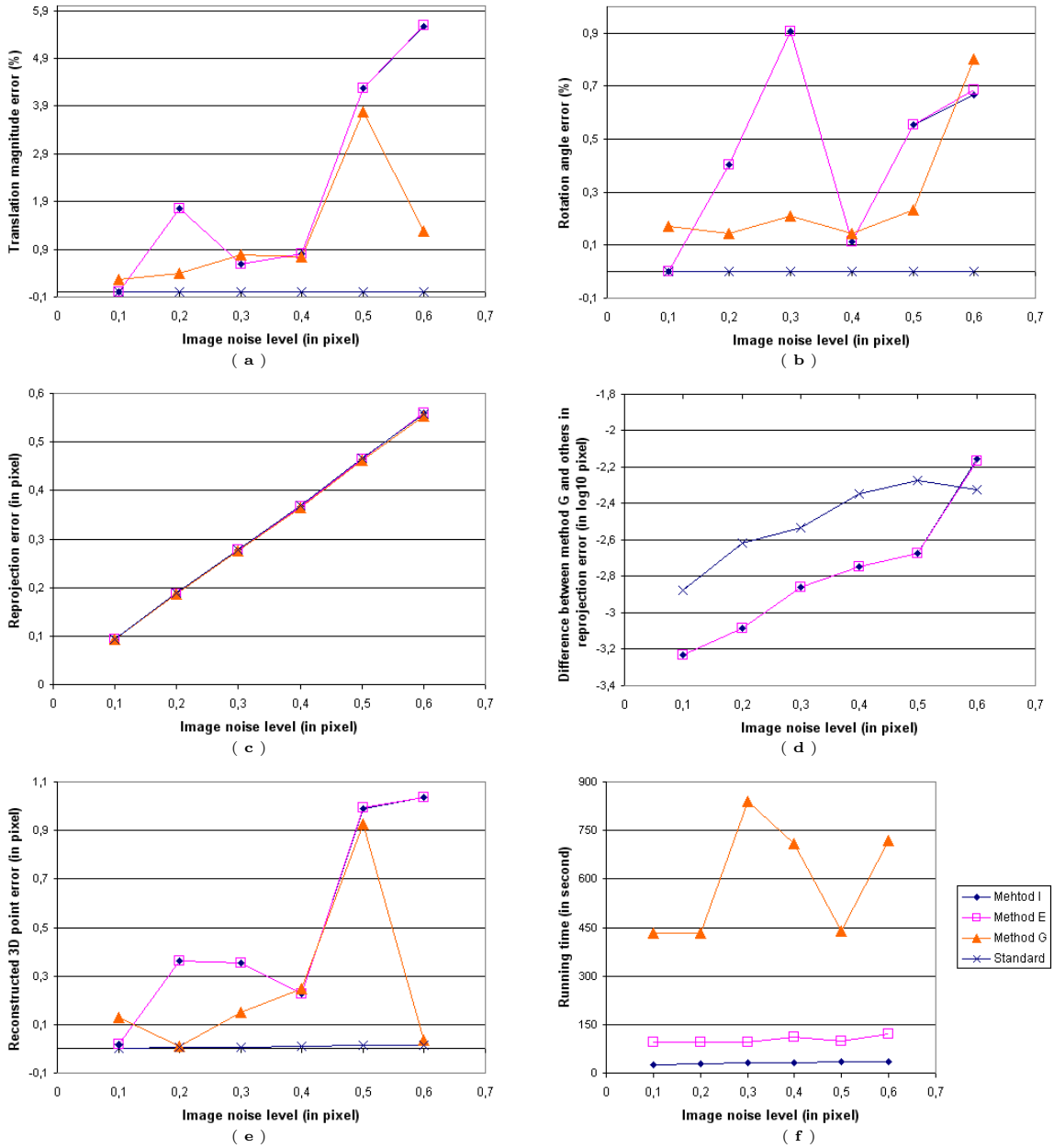


Figure 6.5: **5-view incremental reconstruction for synthetic Dinosaur37.** (a) *Error of the translation magnitude vs. image noise level;* (b) *Error of the rotation angle vs. image noise level;* (c) *Reprojection error vs. image noise level;* (d) *Difference in reprojection error vs. image noise level;* (e) *Error of reconstructed 3D-points vs. image noise level;* (f) *Running time vs. image noise level.*

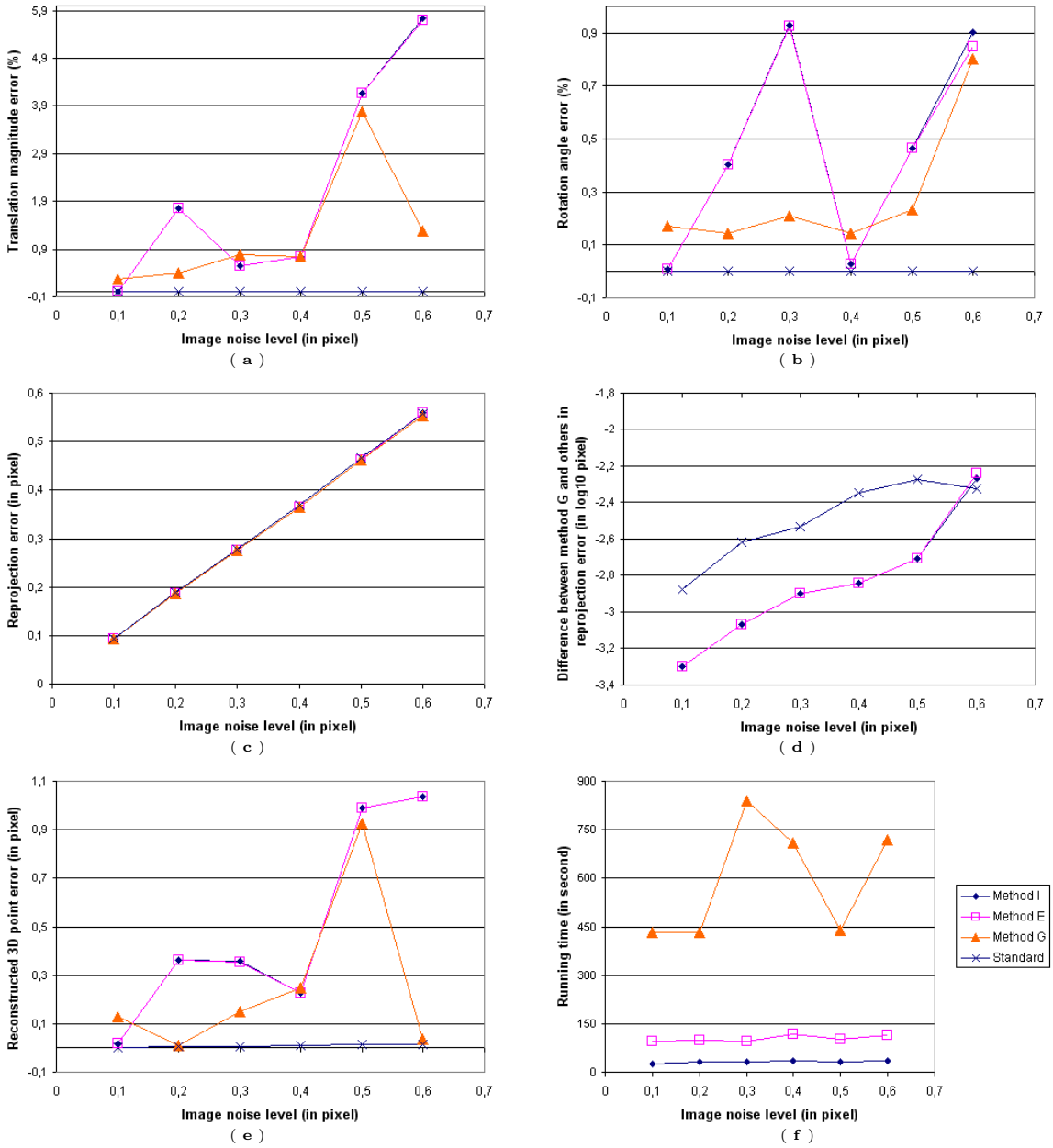


Figure 6.6: **6-view incremental reconstruction for synthetic Dinosaur37.** (a) *Error of the translation magnitude vs. image noise level;* (b) *Error of the rotation angle vs. image noise level;* (c) *Reprojection error vs. image noise level;* (d) *Difference in reprojection error vs. image noise level;* (e) *Error of reconstructed 3D-points vs. image noise level;* (f) *Running time vs. image noise level.*

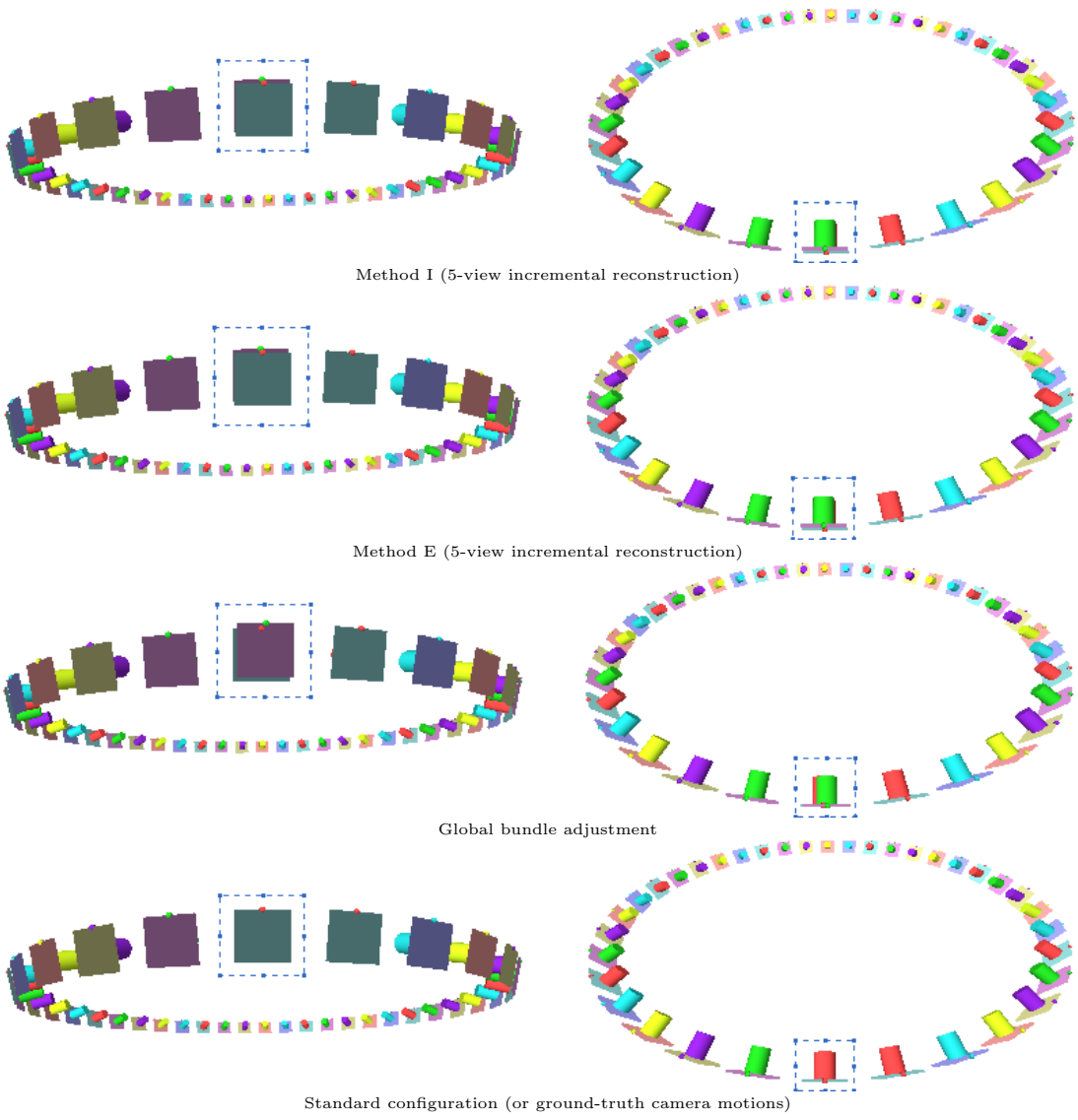


Figure 6.7: **Reconstructed camera motions for the synthetic Dinosaur37 (at noise level of 0.4 pixel) using two 5-view-incremental-reconstruction methods (*Method I* and *Method E*) and the global bundle adjustment.** *Here display the camera motions for a synthetic Dinosaur37 sequence, reconstructed with* Method I *and* Method E *respectively using image-point correspondences across 5 views. The results are compared against those of the global bundle adjustment and the ground-truth camera motions. The left graphs are the frontal profiles of the camera motions with respect to the first motion, and the right are the planforms. The red and the green cameras inside the dashed rectangles represent the first and the last camera motions respectively. In the ground-truth configuration, the camera position of the first frame coincides with that of the last. From the graphs above, the results of* Method I *and* Method E *are too close to be distinguished visually from each other. The related statistical evaluation of this experiment is given in Tab. 6.2. In this experiment, the global bundle adjustment achieves no better results than the incremental-reconstruction methods in terms of camera motions, but its final reprojection error (or the residual of the cost function) is much reduced (see Tab. 6.2).*

	$\overline{\|\mathbf{t}\|}$	$\delta_{\|\mathbf{t}\|}$	$\overline{\mathbf{t}}$	$\delta_{\mathbf{t}}$
3-view	1.0097	0.0327	1.0066, 0.0138, -0.0773	0.0327, 0.0104, 0.0062
4-view	1.0084	0.0213	1.0053, 0.0142, -0.0772	0.0212, 0.0091, 0.0052
5-view	1.0080	0.0185	1.0049, 0.0145, -0.0772	0.0185, 0.0091, 0.0050
6-view	1.0075	0.0166	1.0044, 0.0150, -0.0774	0.0166, 0.0086, 0.0046
Method G	1.0075	0.0098	1.0043, 0.0165, -0.0781	0.0096, 0.0068, 0.0041
Standard	1.0000	0.0000	0.9969, 0.0152, -0.0773	0.0000, 0.0000, 0.0000
	$\overline{\alpha}$	δ_{α}	$\overline{\mathbf{r}}$	$\delta_{\mathbf{r}}$
3-view	10.0295	0.2575	0.0196, 0.9102, 0.4134	0.0196, 0.9102, 0.4134
4-view	10.0157	0.1648	0.0193, 0.9100, 0.4141	0.0084, 0.0029, 0.0066
5-view	10.0113	0.1443	0.0191, 0.9099, 0.4143	0.0084, 0.0025, 0.0056
6-view	10.0030	0.1329	0.0186, 0.9097, 0.4148	0.0079, 0.0023, 0.0051
Method G	9.9855	0.0841	0.0173, 0.9088, 0.4168	0.0061, 0.0017, 0.0037
Standard	10.0000	0.0000	0.0184, 0.9087, 0.4170	0.0000, 0.0000, 0.0000
	$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
3-view	0.3695	0.2107	-0.0003, -0.0000	0.2685, 0.3299
4-view	0.3664	0.2087	-0.0002, -0.0000	0.2633, 0.3294
5-view	0.3659	0.2083	-0.0001, -0.0000	0.2624, 0.3292
6-view	0.3655	0.2082	-0.0001, -0.0000	0.2621, 0.3290
Method G	0.3641	0.2072	-0.0000, -0.0000	0.2603, 0.3282
Standard	0.3686	0.2097	-0.0003, -0.0001	0.2638, 0.3320
	$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
3-view	0.2314	0.0206	-0.0014, -0.0021, 0.2299	0.0221, 0.0145, 0.0209
4-view	0.2286	0.0142	-0.0009, -0.0022, 0.2279	0.0141, 0.0107, 0.0142
5-view	0.2270	0.0127	0.0018, -0.0031, 0.2264	0.0127, 0.0101, 0.0127
6-view	0.2263	0.0118	0.0032, -0.0045, 0.2257	0.0119, 0.0100, 0.0117
Method G	0.2474	0.0125	0.0044, -0.0110, 0.2464	0.0145, 0.0128, 0.0123
Standard	0.0108	0.0082	-0.0002, 0.0000, 0.0002	0.0086, 0.0065, 0.0082

Table 6.3: **Statistical evaluation of (3–6)-view incremental reconstructions for the synthetic Dinosaur37 (at noise level of 0.4 pixel) with *Method I.*** *Incremental reconstructions are conducted for the synthetic Dinosaur37 with* Method I *using point correspondences across different numbers of views (3 to 6 views) respectively. In this table their results are compared statistically with those of the global bundle adjustment (*Method G*) and the standard configuration. The related camera motions reconstructed in this experiment is compared in Fig. 6.9. It can be seen from the table that, the reconstruction accuracy increases with the number of views in the incremental reconstructions both in terms of the reprojection error and in terms of the reconstructed motions and structures.*

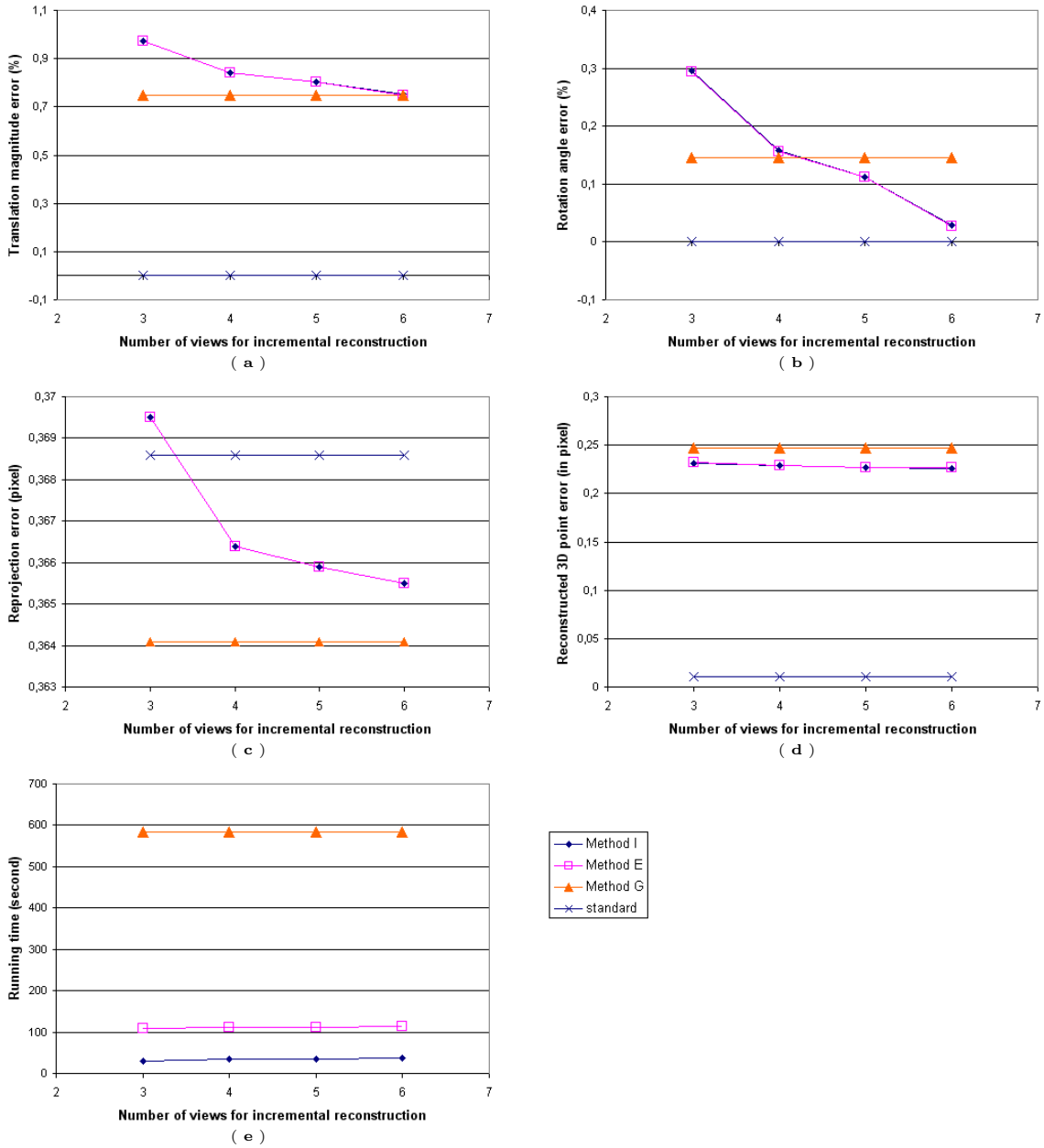


Figure 6.8: **(3–6)-view incremental reconstruction for synthetic Dinosaur37 (at noise level of 0.4 pixel).** (a) *Error of the translation magnitude vs. the number of views;* (b) *Error of the rotation angle vs. the number of views;* (c) *Reprojection error vs. the number of views;* (d) *Error of reconstructed 3D-points vs. the number of views;* (e) *Running time vs. the number of views. Incremental reconstructions are conducted for the synthetic Dinosaur37 (at noise level of 0.4 pixel) with* Method I *and* Method E *using point correspondences across different numbers of views. The results are compared against those of the global bundle adjustment and the standard configuration in the above graphs. Very similar results are obtained with the two incremental-reconstruction methods, except that* Method I *is by far faster than* Method E. *In every respect, the reconstruction accuracy increases with the number of views, though the improvement for the reconstructed 3D points is relatively insignificant. Obviously the running time increases with the number of views.*

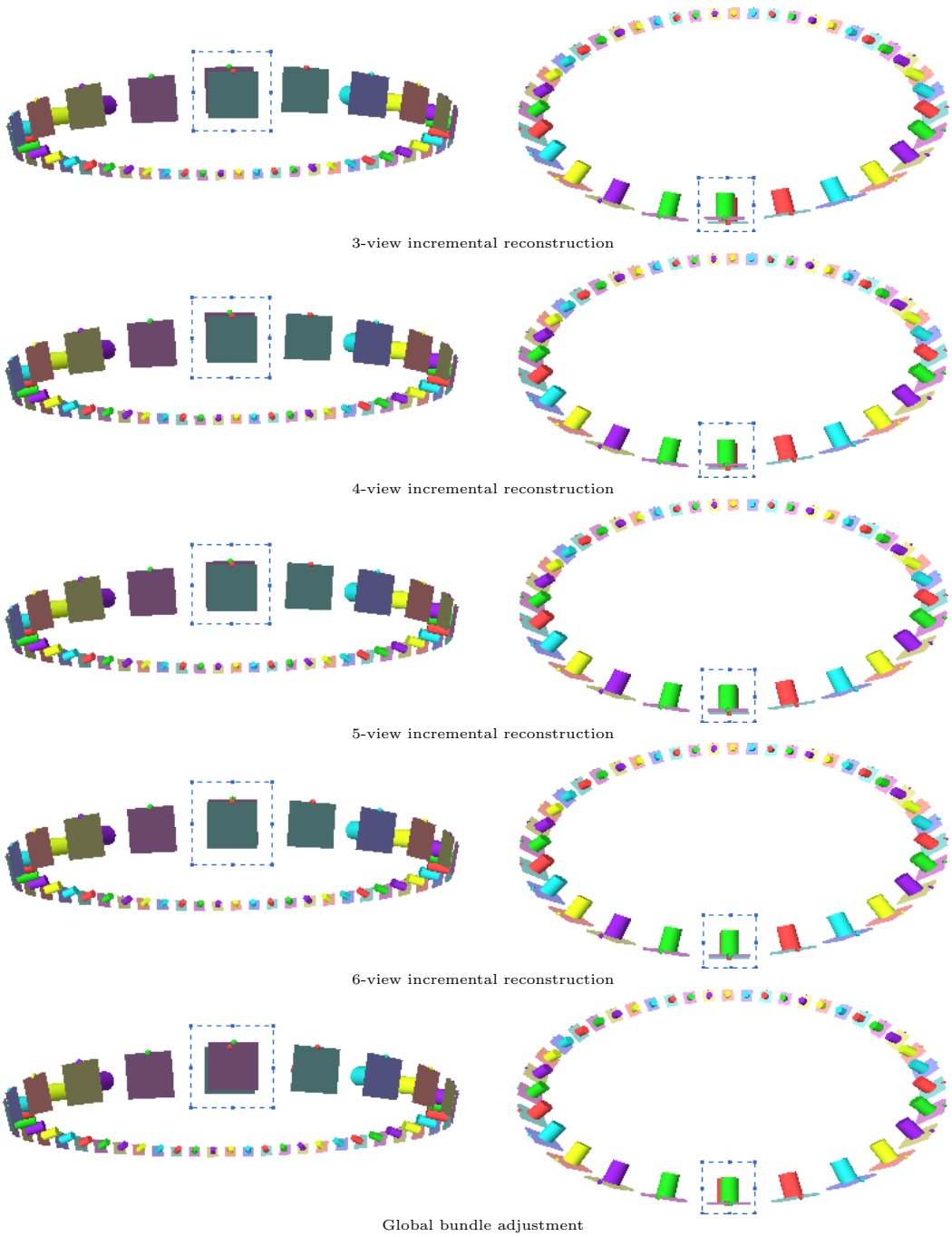


Figure 6.9: **Reconstructed camera motions for synthetic Dinosaur37 (at noise level of 0.4 pixel) with *Method I* and the global bundle adjustment.** *Here display the camera motions for a synthetic sequence of Dinosaur37 reconstructed with* Method I *using image-point correspondences across different numbers of views. The results are compared against those obtained with the global bundle adjustment. The slight improvement may be observed from the figures, when correspondences across more number of views are taken into account. The related statistical evaluation of this experiment is given in Tab. 6.3. As with the results in Fig. 6.7, the global bundle adjustment achieves no better results than the incremental-reconstruction methods in terms of camera motions.*

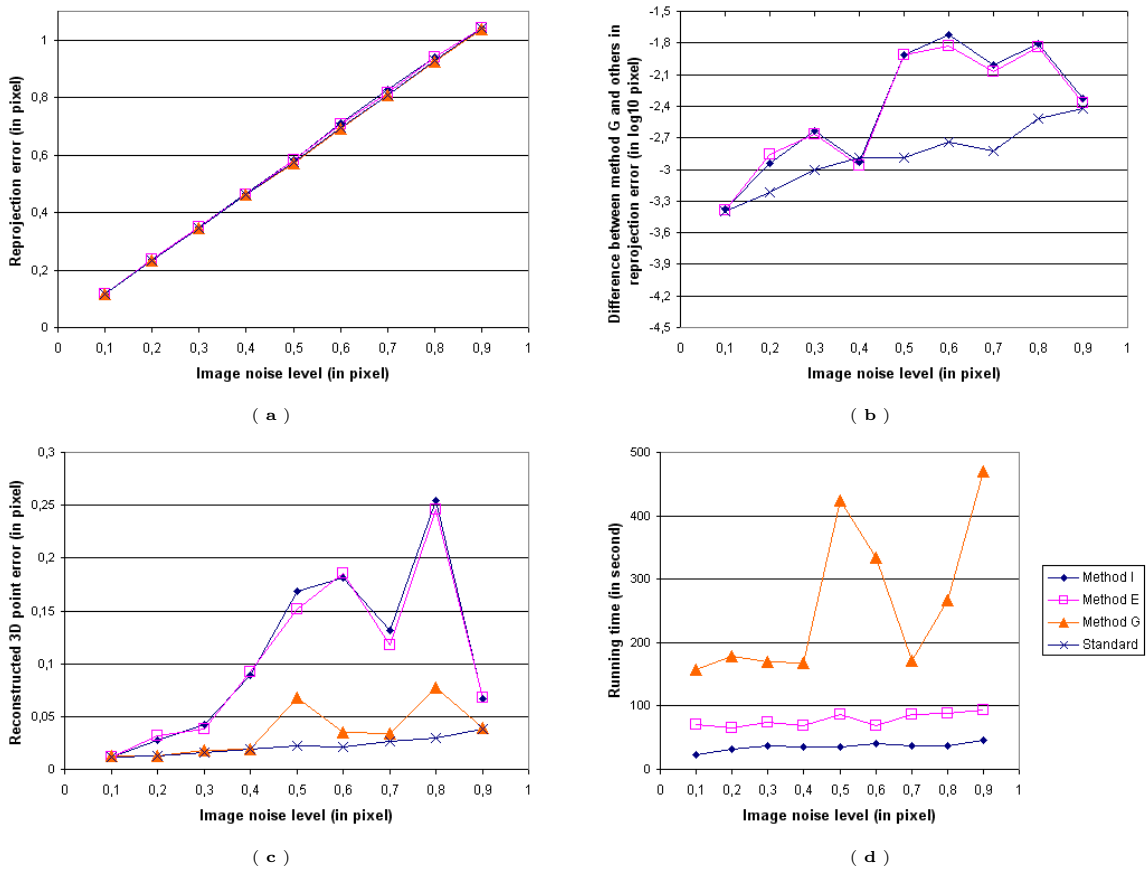


Figure 6.10: **3-view incremental reconstruction for synthetic Mouse11:** (a) *Reprojection error vs. image noise level;* (b) *Difference in reprojection error vs. image noise level;* (c) *Error of reconstructed 3D-points vs. image noise level;* (d) *Running time vs. image noise level.*

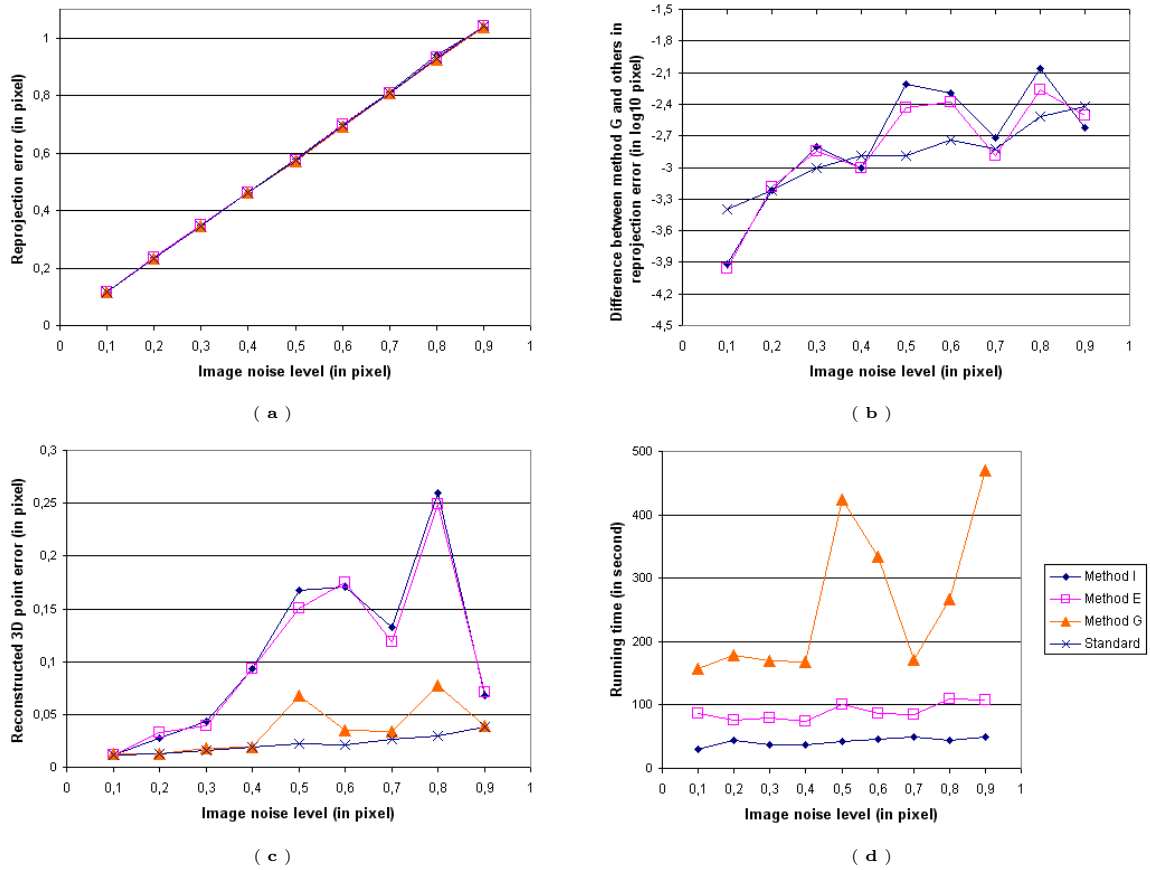


Figure 6.11: **4-view incremental reconstruction for synthetic Mouse11:** (a) *Reprojection error vs. image noise level;* (b) *Difference in reprojection error vs. image noise level;* (c) *Error of reconstructed 3D-points vs. image noise level;* (d) *Running time vs. image noise level.*

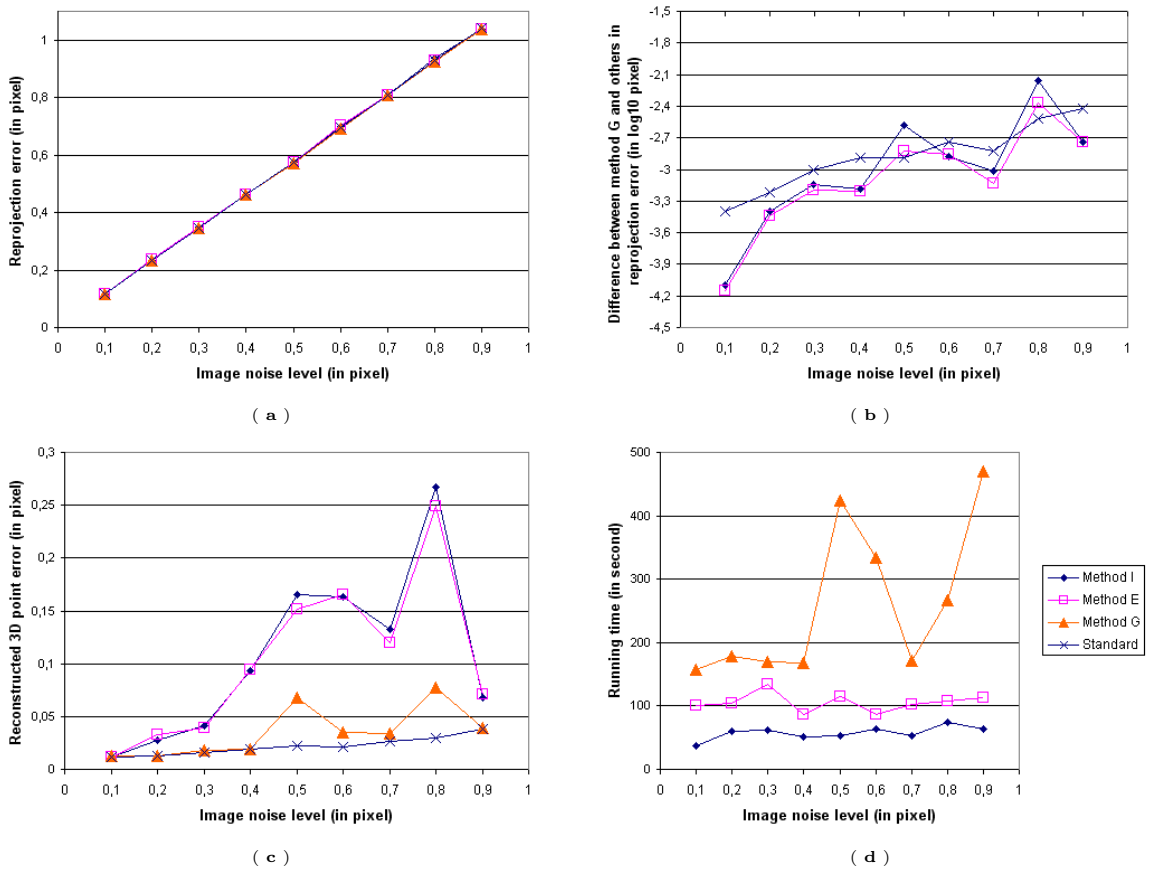


Figure 6.12: **5-view incremental reconstruction for synthetic Mouse11:** (a) *Reprojection error vs. image noise level;* (b) *Difference in reprojection error vs. image noise level;* (c) *Error of reconstructed 3D-points vs. image noise level;* (d) *Running time vs. image noise level.*

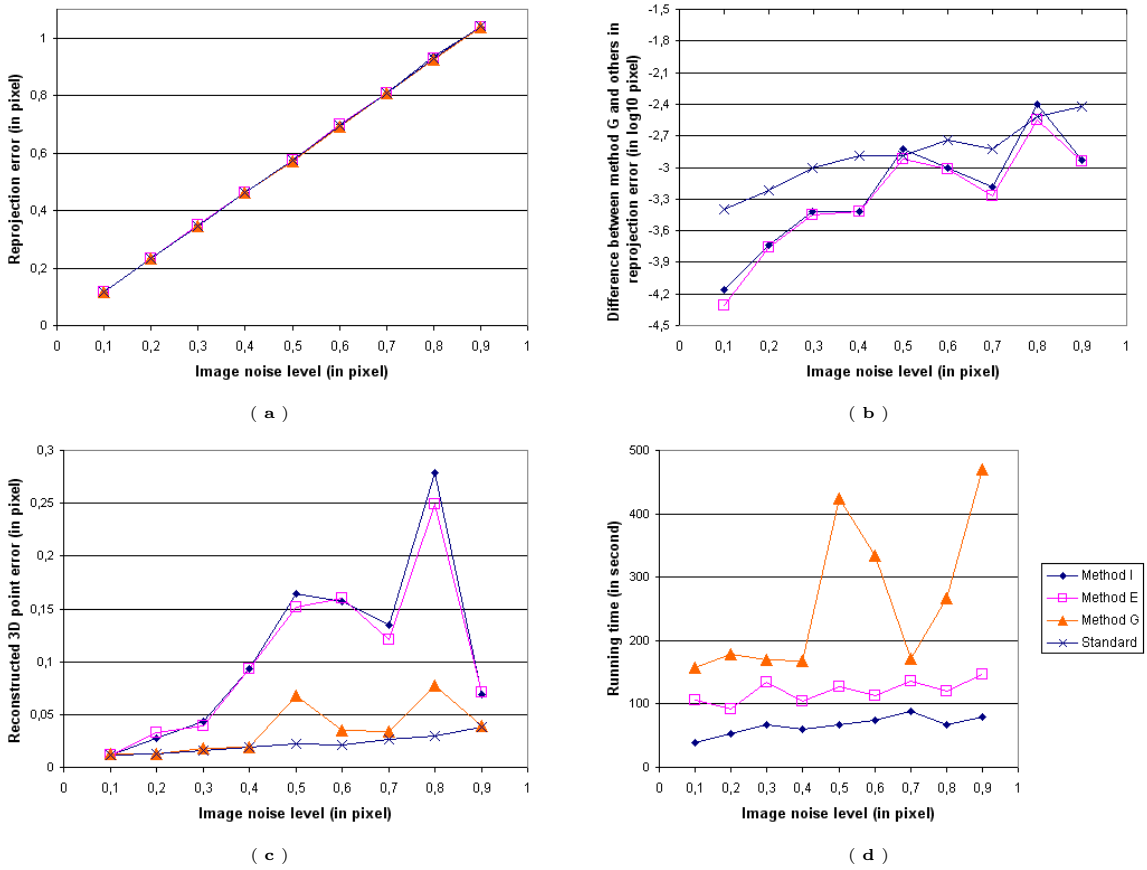


Figure 6.13: **6-view incremental reconstruction for synthetic Mouse11:** (a) *Reprojection error vs. image noise level;* (b) *Difference in reprojection error vs. image noise level;* (c) *Error of reconstructed 3D-points vs. image noise level;* (d) *Running time vs. image noise level.*

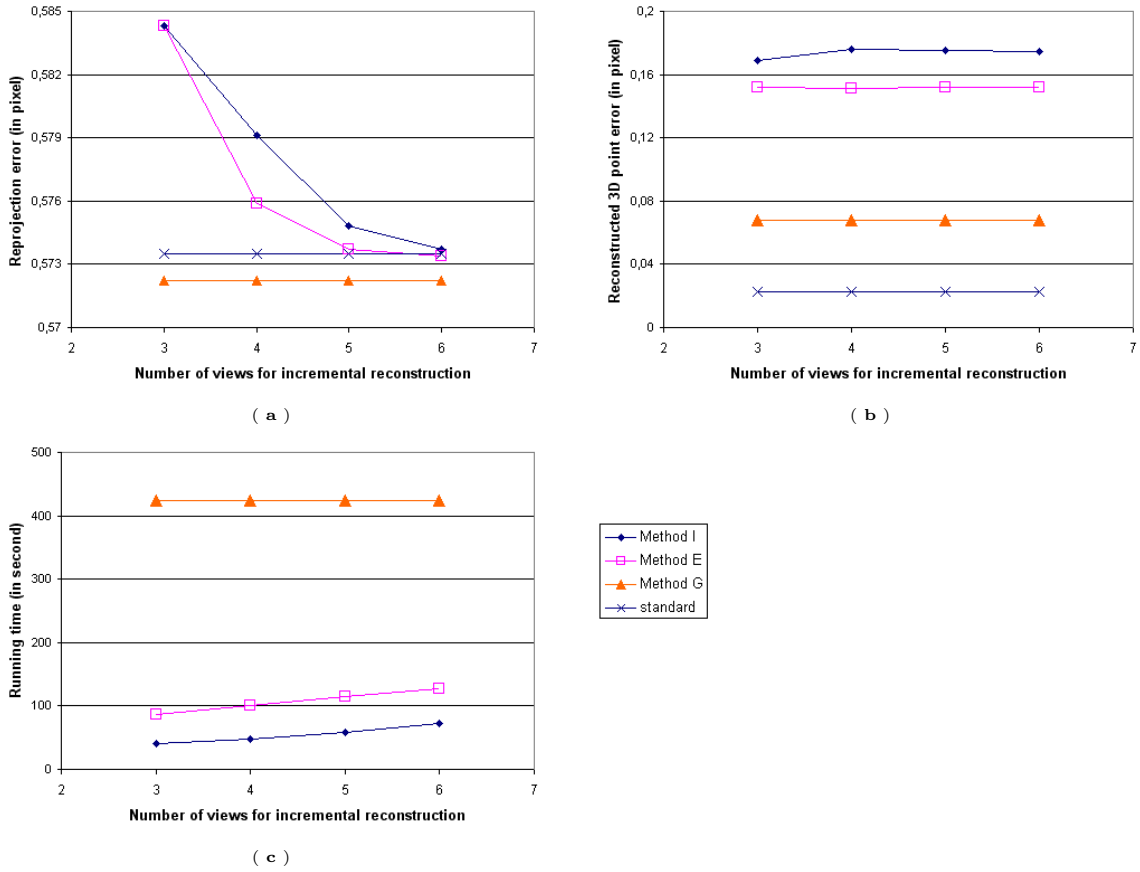


Figure 6.14: **(3–6)-view incremental reconstruction for synthetic Mouse11 (at noise level of 0.5 pixel):** (a) *Reprojection error vs. the number of views;* (b) *Error of reconstructed 3D-points vs. the number of views;* (c) *Running time vs. the number of views.*

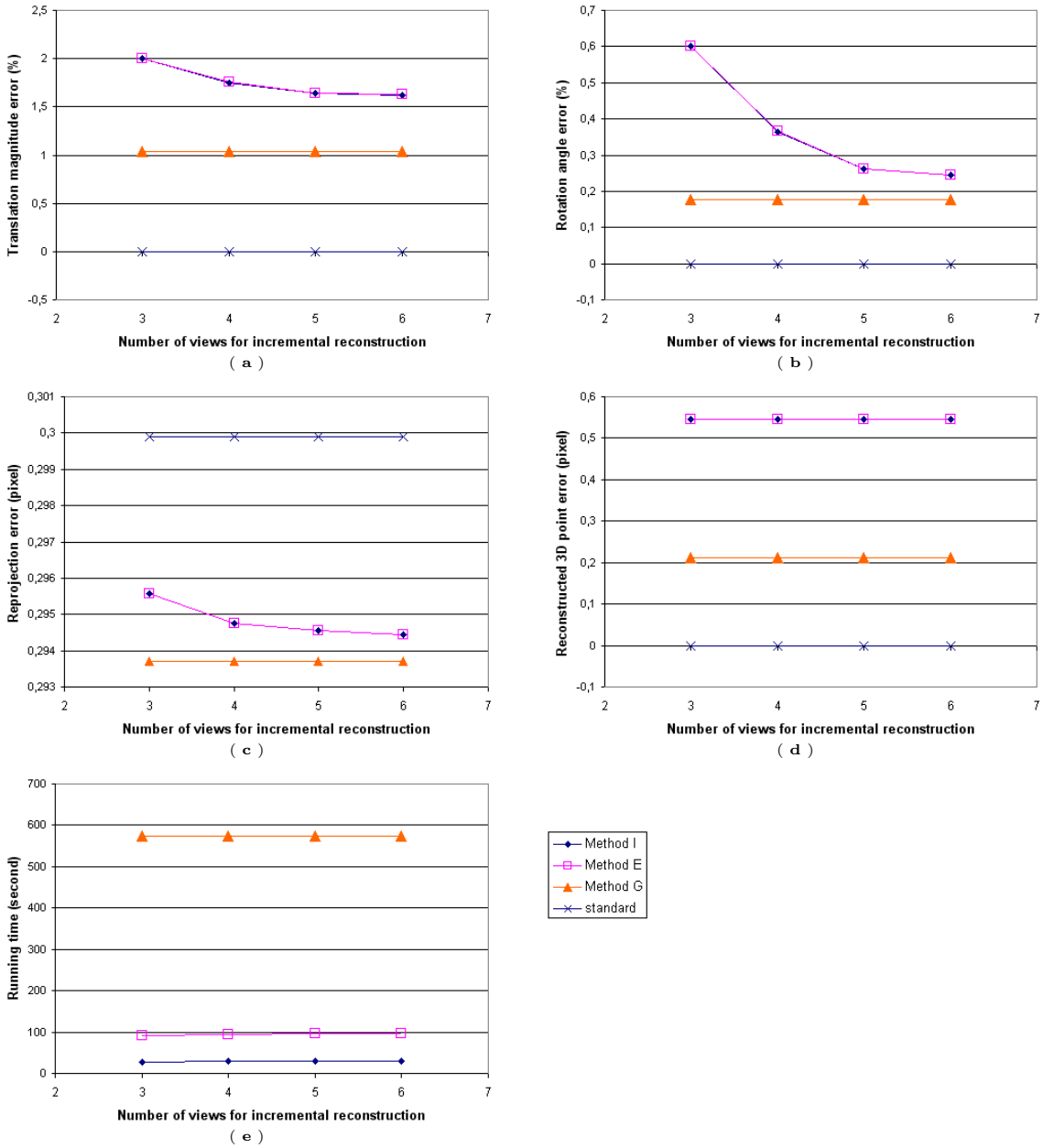


Figure 6.15: **(3–6)-view incremental reconstruction for real Dinosaur37** *(compare with Fig. 6.8).* (a) *Error of the translation magnitude vs. the number of views used for incremental reconstruction;* (b) *Error of the rotation angle vs. the number of views;* (c) *Reprojection error vs. the number of views;* (d) *Error of reconstructed 3D-points vs. the number of views;* (e) *Running time vs. the number of views.*

	$\overline{\|\mathbf{t}\|}$	$\delta_{\|\mathbf{t}\|}$	$\overline{\mathbf{t}}$	$\delta_{\mathbf{t}}$
3-view	1.0284	0.0199	1.0252, 0.0107, -0.0786	0.0197, 0.0172, 0.0093
4-view	1.0259	0.0207	1.0227, 0.0112, -0.0784	0.0206, 0.0164, 0.0090
5-view	1.0248	0.0201	1.0216, 0.0114, -0.0783	0.0199, 0.0162, 0.0089
6-view	1.0246	0.0203	1.0214, 0.0115, -0.0783	0.0201, 0.0161, 0.0090
Method G	1.0104	0.0184	1.0072, 0.0110, -0.0769	0.0181, 0.0160, 0.0090
Standard	1.0000	0.0000	0.9969, 0.0152, -0.0773	0.0000, 0.0000, 0.0000
	$\overline{\alpha}$	δ_{α}	$\overline{\mathbf{r}}$	$\delta_{\mathbf{r}}$
3-view	10.0600	0.1732	0.0227, 0.9112, 0.4108	0.0153, 0.0039, 0.0087
4-view	10.0365	0.1771	0.0218, 0.9108, 0.4118	0.0147, 0.0040, 0.0090
5-view	10.0261	0.1680	0.0220, 0.9106, 0.4123	0.0145, 0.0041, 0.0091
6-view	10.0244	0.1682	0.0219, 0.9105, 0.4125	0.0144, 0.0041, 0.0092
Method G	10.0177	0.1423	0.0222, 0.9103, 0.4129	0.0144, 0.0047, 0.0104
Standard	10.0000	0.0000	0.0184, 0.9087, 0.4170	0.0000, 0.0000, 0.0000
	$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
3-view	0.2956	0.3054	-0.0005, 0.0000	0.2092, 0.3700
4-view	0.2948	0.3054	-0.0003, 0.0000	0.2083, 0.3699
5-view	0.2945	0.3054	-0.0002, 0.0000	0.2078, 0.3700
6-view	0.2944	0.3055	-0.0002, -0.0000	0.2077, 0.3700
Method G	0.2937	0.3034	0.0000, -0.0000	0.2060, 0.3686
Standard	0.2999	0.3249	0.0041, -0.0002	0.2094, 0.3893
	$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
3-view	0.5450	0.0133	-0.0158, 0.0080, 0.5440	0.0180, 0.0228, 0.0133
4-view	0.5465	0.0144	-0.0146, 0.0033, 0.5456	0.0181, 0.0209, 0.0143
5-view	0.5458	0.0148	-0.0127, 0.0016, 0.5449	0.0182, 0.0205, 0.0147
6-view	0.5446	0.0147	-0.0108, 0.0006, 0.5438	0.0183, 0.0204, 0.0146
Method G	0.2114	0.0105	-0.0036, 0.0032, 0.2108	0.0120, 0.0105, 0.0103
Standard	0.0000	0.0000	0.0000, 0.0000, 0.0000	0.0000, 0.0000, 0.0000

Table 6.4: **Statistical evaluation of (3–6)-view incremental reconstructions for real Dinosaur37 with *Method I*** *(compare with Tab. 6.3).*

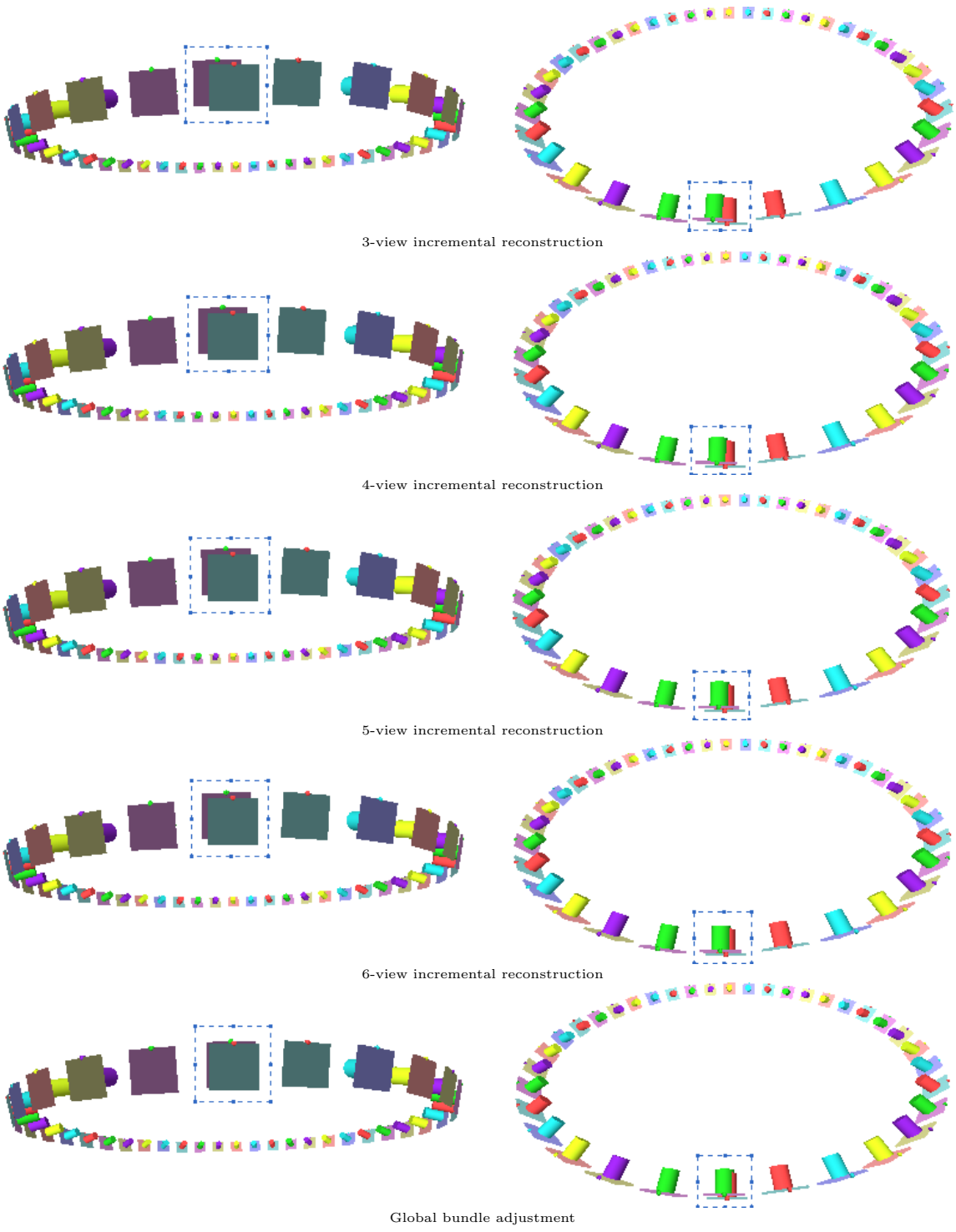


Figure 6.16: **Reconstructed camera motions for real Dinosaur37 with *Method I* and the global bundle adjustment** *(compare with Fig. 6.9)*.

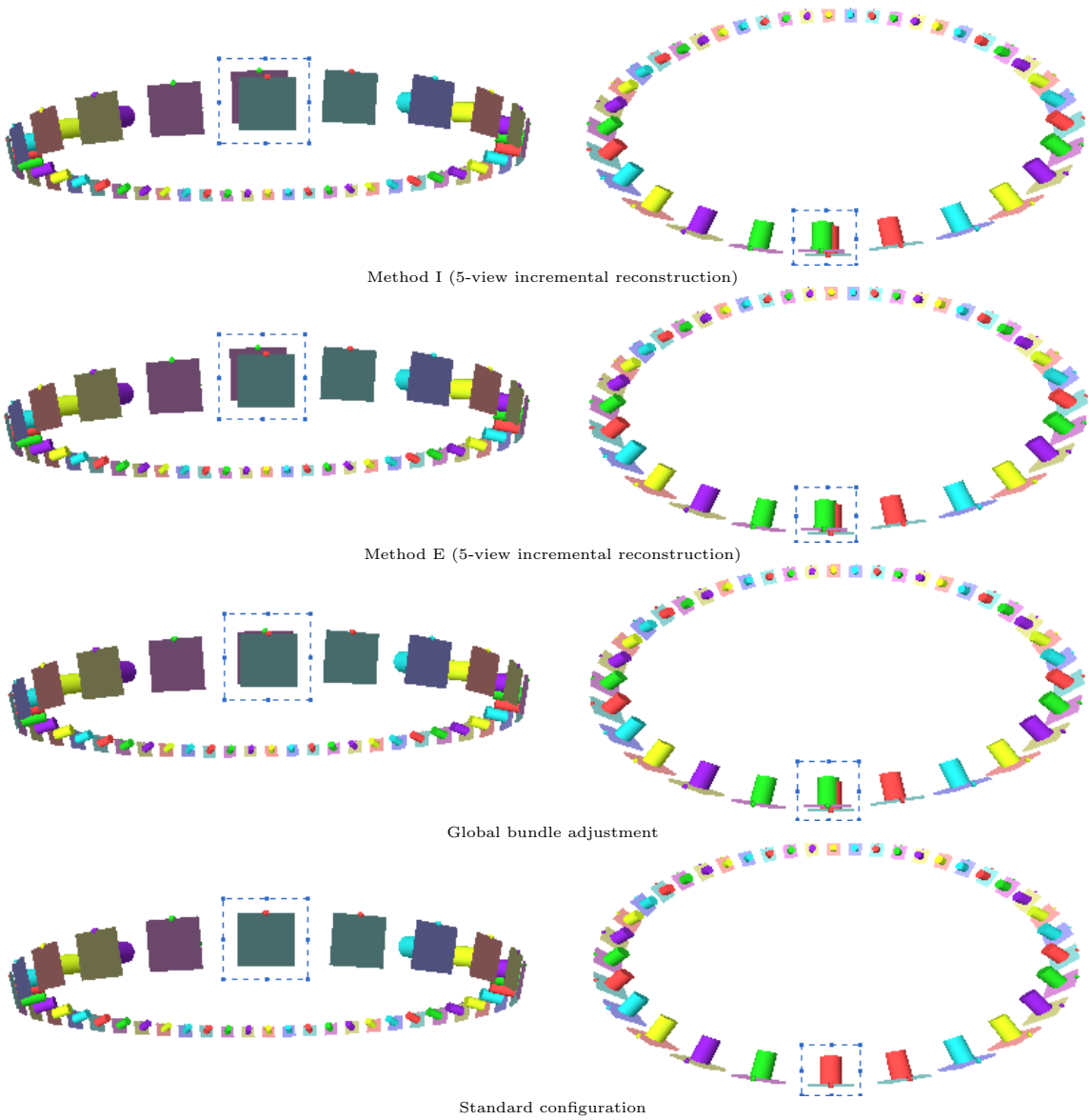


Figure 6.17: **Reconstructed camera motions for real Dinosaur37 using two 5-view-incremental-reconstruction methods (*Method I* and *Method E*) and the global bundle adjustment** *(compare with Fig. 6.7).*

	$\overline{\|\mathbf{t}\|}$	$\delta_{\|\mathbf{t}\|}$	$\overline{\mathbf{t}}$	$\delta_{\mathbf{t}}$
Method I	1.0248	0.0201	1.0216, 0.0114, -0.0783	0.0199, 0.0162, 0.0089
Method E	1.0248	0.0200	1.0216, 0.0114, -0.0783	0.0198, 0.0162, 0.0089
Method G	1.0104	0.0184	1.0072, 0.0110, -0.0769	0.0181, 0.0160, 0.0090
Standard	1.0000	0.0000	0.9969, 0.0152, -0.0773	0.0000, 0.0000, 0.0000
	$\overline{\alpha}$	δ_{α}	$\overline{\mathbf{r}}$	$\delta_{\mathbf{r}}$
Method I	10.0261	0.1680	0.0220, 0.9106, 0.4123	0.0145, 0.0041, 0.0091
Method E	10.0263	0.1677	0.0220, 0.9106, 0.4123	0.0145, 0.0041, 0.0091
Method G	10.0177	0.1423	0.0222, 0.9103, 0.4129	0.0144, 0.0047, 0.0104
Standard	10.0000	0.0000	0.0184, 0.9087, 0.4170	0.0000, 0.0000, 0.0000
	$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
Method I	0.2945	0.3054	-0.0003, 0.0000	0.2078, 0.3700
Method E	0.2945	0.3054	-0.0003, 0.0000	0.2078, 0.3700
Method G	0.2937	0.3034	0.0000, -0.0000	0.2060, 0.3686
Standard	0.2999	0.3249	0.0041, -0.0002	0.2094, 0.3893
	$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
Method I	0.5458	0.0148	-0.0127, 0.0016, 0.5449	0.0182, 0.0205, 0.0147
Method E	0.5456	0.0148	-0.0127, 0.0016, 0.5449	0.0182, 0.0205, 0.0147
Method G	0.2114	0.0105	-0.0036, 0.0032, 0.2108	0.0120, 0.0105, 0.0103
Standard	0.0000	0.0000	0.0000, 0.0000, 0.0000	0.0000, 0.0000, 0.0000

Table 6.5: **Statistical evaluation of 5-view incremental reconstructions for real Dinosaur37 with *Method I* and *Method E*** *(compare with Tab. 6.2).*

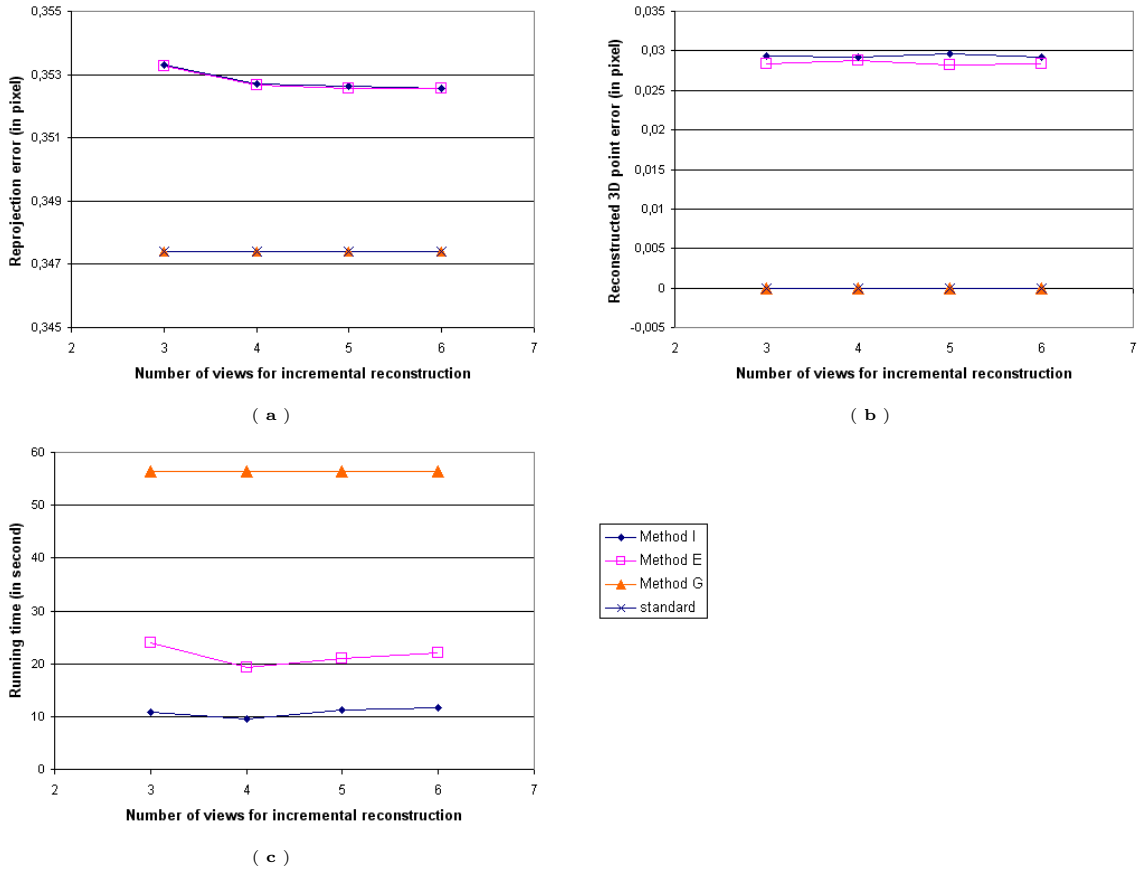


Figure 6.18: **(3–4)-view incremental reconstructions for real Mouse11** *(compare with Fig. 6.14).* (a) tslReprojection error vs. the number of views; (b) tslError of reconstructed 3D-points vs. the number of views; (c) tslRunning time vs. the number of views.

		$\overline{\|\Delta\mathbf{x}\|}$	$\delta_{\|\Delta\mathbf{x}\|}$	$\overline{\Delta\mathbf{x}}$	$\delta_{\Delta\mathbf{x}}$
3-view	**I**	0.35329	0.23913	-0.00106, -0.00065	0.26278, 0.33409
	E	0.35326	0.23917	-0.00230, -0.00017	0.26291, 0.33398
4-view	**I**	0.35271	0.23902	-0.00092, -0.00007	0.26218, 0.33388
	E	0.35268	0.23899	-0.00136, -0.00002	0.26210, 0.33389
5-view	**I**	0.35262	0.23899	-0.00059, -0.00011	0.26216, 0.33384
	E	0.35256	0.23900	-0.00032, 0.00001	0.26191, 0.33391
6-view	**I**	0.35256	0.23891	-0.00006, -0.00009	0.26187, 0.33390
	E	0.35255	0.23894	-0.00013, 0.00001	0.26181, 0.33395
Method G		0.34742	0.23880	-0.00000, -0.00012	0.26164, 0.33383
		$\overline{\|\Delta\mathbf{X}\|}$	$\delta_{\|\Delta\mathbf{X}\|}$	$\overline{\Delta\mathbf{X}}$	$\delta_{\Delta\mathbf{X}}$
3-view	**I**	0.02937	0.03742	-0.00354, 0.00329, -0.03093	0.01674, 0.01165, 0.03809
	E	0.02836	0.03602	-0.00327, 0.00273, -0.02713	0.01444, 0.00935, 0.03242
4-view	**I**	0.02918	0.04096	-0.00387, 0.00334, -0.03165	0.01807, 0.01184, 0.03746
	E	0.02879	0.03456	-0.00338, 0.00285, -0.02748	0.01555, 0.01030, 0.03565
5-view	**I**	0.02963	0.03857	-0.00367, 0.00325, -0.03118	0.01731, 0.01104, 0.03531
	E	0.02822	0.03828	-0.00312, 0.00278, -0.02697	0.01507, 0.00985, 0.03453
6-view	**I**	0.02922	0.03805	-0.00346, 0.00321, -0.03080	0.01703, 0.01096, 0.03486
	E	0.02829	0.03890	-0.00307, 0.00283, -0.02703	0.01530, 0.01008, 0.03507
Method G		0.00000	0.00000	0.00000, 0.00000, 0.00000	0.00000, 0.00000, 0.00000

Table 6.6: **Statistical evaluation of the reconstructions for real Mouse11 with *Method I*, *Method E* and *Method G*(the global bundle adjustment).**

7 Summary

7.1 Conclusions

As the title suggests, this dissertation attempts to solve the problem of 3D reconstruction from multiple images in a more efficient manner. The research is known as *Multiple-View Reconstruction*. As is mentioned in Chapter 2, multiple-view reconstruction is a comprehensive literature in computer vision. Three techniques are developed in the dissertation, which solve respectively three fundamental problems within this topic.

Firstly, a new linear and non-iterative method to reconstruct a 3D-point in space from its projections in multiple views with known projection matrices is proposed in Chapter 3. This method is called *1st-order MLE*, since it converts the original reconstruction problem into one of linearly-constrained quadratic optimization through a first-order approximation to the epipolar constraints.

Chapter 4 proposes a linear iterative least-squares method for estimating the fundamental matrix between two un-calibrated perspective views. Like in chapter 3 the problem is converted into a least-squares problem by a first-order approximation to the epipolar constraints. Adaptively, the algebraic cost function minimized in the least-squares problem approaches the geometric error, and a more accurate fundamental matrix is obtained iteratively.

The techniques presented in Chapter 5 are extensive applications of the above *1st-order MLE* method to the problem of *bundle adjustment*. By this, the cost function of bundle adjustment is partly linearized, and accordingly the minimization process is accelerated.

All the above techniques preserve the error model of the measured image points, and allow the assignment of individual covariance to each image measurement. Ex-

periments show that the accuracy of these algorithms is consistently comparable to that of a *maximum likelihood estimation* using numerical Newton-type optimization, however, at a much reduced computational cost.

Finally, based on the above techniques, an incremental multiple view reconstruction method is developed in Chapter 6. The higher efficiency of these techniques allows the incremental reconstruction method to take point-matches across multiple views into consideration, and thus more accurate results are achieved. This is different from previous approaches which consider only point matches across two or three views. Therefore both higher accuracy and efficiency can be achieved at the same time by the proposed multiple view reconstruction method.

7.2 Future Work

People can easily manipulate in a three dimensional world, although they only sense 2D projections of it. For years, researchers have worked on this seemingly effortless behavior. However, despite the fact that we seem to have known quite a lot about vision, the state-of-the-art computer vision systems still have no match for human vision, especially when the efficiency of the process is concerned. Though mathematics allows us to study the fundamental geometric principles underlying visual perception as this dissertation has shown, time used to process the massive computations is still by far incomparable to that of the human vision system. A thorough understanding and simulation of the phenomenon of vision not only depends on the advancement in mathematics, computer science and electronic engineering, but also relies on more interdisciplinary efforts from many other disciplines such as neuroscience, psychophysics, and cognitive science.

A Projective Geometry and Transformations

The properties and entities of projective space, especially $\mathcal{P}^2$ and $\mathcal{P}^3$, are described in this appendix. Projective space is an augmented Euclidean space with a set of ideal points at infinity. Homogeneous coordinates play an important role in it, with all dimensions increased by one.

This appendix begins with describing the homogeneous representation of points, lines and planes in projective space, and how these entities map under projective transformations. Then it introduces a hierarchy of projective transformations and the invariant properties under different levels of transformations.

A.1 The Projective Space - Homogeneous Coordinates

It is common to identify a plane with 2D Euclidean space $\mathcal{R}^2$, and a 3-space with 3D Euclidean space $\mathcal{R}^3$; and the finite points in them are represented by 2-vectors and 3-vectors respectively. *Projective space* is an extension to Euclidean space in which points, lines or planes at infinity are treated no differently from those in finite space. The following of this section will introduce the *homogeneous* representation of points, lines, and planes in *projective space*.

Points in projective n-space, $\mathcal{P}^n$

A point in projective n-space, $\mathcal{P}^n$, is given by a $(n+1)$-vector of coordinates $\tilde{\mathbf{x}} = (x_1, \cdots, x_{n+1})^\top$. At least one of these coordinates should differ from zero. These

coordinates are called homogeneous coordinates. When $x_{n+1} \neq 0$, $\tilde{\mathbf{x}}$ corresponds the point $\mathbf{x} = (x_1/x_{n+1}, \cdots, x_n/x_{n+1})^\top$ in Euclidean space $\mathcal{R}^n$; when $x_{n+1} = 0$, the point is known as a *ideal point*, corresponding to the point at infinity in the direction of $(x_1, \cdots, x_n)^\top$.

The other way round, the homogeneous corresponds for a point $\mathbf{x} = (x_1, \cdots, x_n)^\top$ in Euclidean space $\mathcal{R}^n$ may be written as $\tilde{\mathbf{x}} = (x_1, \cdots, x_n, 1)^\top$.

Two homogeneous points $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ are equal if and only if there exists a nonzero scalar λ such that $\tilde{\mathbf{p}} = \lambda\tilde{\mathbf{q}}$. This relationship is indicated by $\tilde{\mathbf{p}} \sim \tilde{\mathbf{q}}$, with $\sim$ meaning equality up to a nonzero scalar factor.

Points and Lines in $\mathcal{P}^2$

A line in projective 2-space, $\mathcal{P}^2$, e.g. $l_1x + l_2y + l_3 = 0$, is represented by the homogeneous 3-vector $\mathbf{l} = (l_1, l_2, l_3)^\top$. As with homogeneous points, only the ratio of homogeneous line coordinates is significant, but not the scaling. The line $\mathbf{l}_\infty = (0, 0, 1)^\top$ is known as the *line at infinity*. All the ideal points in $\mathcal{P}^2$ lie on $\mathbf{l}_\infty$, since $(0, 0, 1)(x_1, x_2, 0)^\top = 0$.

The join and incidence relations between the points and lines in $\mathcal{P}^2$ are summarized as follows:

- A point $\tilde{\mathbf{x}}$ lies on a line $\mathbf{l}$ if and only if $\mathbf{l}^\top\tilde{\mathbf{x}} = 0$, i.e. $\tilde{\mathbf{x}}^\top\mathbf{l} = 0$.
- Two distinct points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ define a line: $\mathbf{l} = \tilde{\mathbf{x}} \times \tilde{\mathbf{x}}' = [\tilde{\mathbf{x}}]_\times\tilde{\mathbf{x}}'$.
- Two distinct lines $\mathbf{l}$ and $\mathbf{l}'$ define a point: $\tilde{\mathbf{x}} = \mathbf{l} \times \mathbf{l}' = [\tilde{\mathbf{l}}_\times\tilde{\mathbf{l}}'$.

The $\mathbf{v}_\times$ above for a 3-vector $\mathbf{v} = (v_1, v_2, v_3)^\top$ is the *matrix notation for vector product* given by

$$\mathbf{v}_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}.$$

That means, the vector product $\tilde{\mathbf{v}} \times \tilde{\mathbf{x}}$ can be represented as a matrix multiplication $\tilde{\mathbf{v}}_\times\tilde{\mathbf{x}}$. $\tilde{\mathbf{v}}_\times$ is a 3×3 skew-symmetric matrix of rank 2. Its null-vector is $\tilde{\mathbf{v}}$, since $\tilde{\mathbf{v}} \times \tilde{\mathbf{v}} = 0$.

Points, Lines and Planes in $\mathcal{P}^3$

A plane in $\mathcal{P}^3$, e.g. $\pi_1x + \pi_2y + \pi_3z + \pi_4 = 0$, is denoted by the homogeneous 4-vector $\pi = (\pi_1, \pi_2, \pi_3, \pi_4)^\top$. The scaling of the homogeneous plane vector is also insignificant. Every point at infinity $(x_1, x_2, x_3, 0)^\top$ in $\mathcal{P}^3$ lies on a single plane, the *plane at infinity*, denoted by $\pi_\infty = (0, 0, 0, 1)^\top$, for $(0, 0, 0, 1)(x_1, x_2, x_3, 0)^\top = 0$.

The join and incidence relations of the entities in $\mathcal{P}^3$ are given as follows:

- A point $\tilde{\mathbf{X}}$ lies on a plane π if and only if $\pi^\top \tilde{\mathbf{X}} = 0$, i.e. $\tilde{\mathbf{X}}^\top \pi = 0$.
- A plane π is uniquely defined by the join of three non-coplanar points $\tilde{\mathbf{X}}_1$, $\tilde{\mathbf{X}}_2$ and $\tilde{\mathbf{X}}_3$, which satisfies

$$\pi^\top \begin{bmatrix} \tilde{\mathbf{X}}_1 & \tilde{\mathbf{X}}_2 & \tilde{\mathbf{X}}_3 \end{bmatrix} = \mathbf{0}^\top. \tag{A.1}$$

- Three distinct planes π_1, π_2 and π_3 intersect in a unique point $\tilde{\mathbf{X}}$, which satisfies

$$\tilde{\mathbf{X}}^\top \begin{bmatrix} \pi_1 & \pi_2 & \pi_3^\top \end{bmatrix} = \mathbf{0}^\top. \tag{A.2}$$

- A line is uniquely defined by the join of two distinct points $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$, which can be represented by a 4×2 matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} \tilde{\mathbf{X}}_1 & \tilde{\mathbf{X}}_2 \end{bmatrix};$$

  and the line is composed of the pencil of points $\lambda\tilde{\mathbf{X}}_1 + \mu\tilde{\mathbf{X}}_2$, the span of the column space of $\mathbf{W}$, where λ and μ are random scalars.

- A line is uniquely defined by the intersection of two distinct planes π_1 and π_2, which can also be represented by a 4×2 matrix:

$$\mathbf{W}^* = \begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix}.$$

  It is known as the *dual representation* of a line, and the line is the axis of the pencil of planes $\lambda'\tilde{\mathbf{X}}_1 + \mu'\tilde{\mathbf{X}}_2$, the span of the column space of $\mathbf{W}^*$.
  The two representations above are related by $\mathbf{W}^\top \mathbf{W}^* = \mathbf{W}^{*\top} \mathbf{W} = \mathbf{0}_{2\times 2}$, where $\mathbf{0}_{2\times 2}$ is a 2×2 null matrix.

A.2 Projective Transformations

A *projective transformation* is a linear transformation on homogeneous coordinates. For $\mathcal{P}^2$, it is represented by a non-singular 3×3 matrix, and by a non-singular 4×4 matrix for $\mathcal{P}^3$. Tab. A.1 illustrates how points, lines and planes are transformed under projective transformation in $\mathcal{P}^2$ and $\mathcal{P}^3$.

Both in $\mathcal{P}^2$ and $\mathcal{P}^3$ the transformation on homogeneous coordinates can be classified into four levels: projective, affine, similarity, and Euclidean. A hierarchy is set up by the four levels of transformation. The set of affine transformations is a

	$\mathcal{P}^2$	$\mathcal{P}^3$
Size of non-singular transformation matrix ($\mathbf{H}$)	3×3	4×4
Point ($\mathbf{x}$)	$\mathbf{x}' = \mathbf{H}\mathbf{x}$	$\mathbf{x}' = \mathbf{H}\mathbf{x}$
line ($\mathbf{l}$)	$\mathbf{l}' = \mathbf{H}^{-\top}\mathbf{l}$	-
Plane (π)	-	$\pi' = \mathbf{H}^{-\top}\pi$

Table A.1: Mapping of points, lines and planes under projective transformations.

Transformations	$\mathcal{P}^2$		$\mathcal{P}^3$		Invariant Measurements
	DOF	Distortion	DOF	Distortion	
Euclidean	3		6		angles, distances
Similarity	4		7		angles, relative distances
Affine	6		12		parallelism, center of mass
Projective	8		15		collinearity, cross ratio

Table A.2: Comparison of different levels of transformations.

sub-group of projective transformations, and so is similarity to affine, and Euclidean to similarity.

Under different levels of transformations, different geometric properties are invariant. Tab. A.2 lists the invariant properties and the possible distortions of a geometric structure under different levels of transformations in $\mathcal{P}^2$ and $\mathcal{P}^3$ respectively.

Bibliography

[1] Armstrong, M., A. Zisserman and R. Hartley: *Euclidean Reconstruction from Image Triplets*. In *ECCV*, pages 3–16. Lecture Notes in Computer Science, Springer-Verlag, 1996.

[2] Armstrong, M.N.: *Self-Calibration from Image Sequences*. PhD thesis, 1996.

[3] Avidan, S. and A. Shashua: *Threading Fundamental Matrices*. In Burkhardt, H. and B. Neumann (editors): *The 5th European Conference on Computer Vision*, pages 124–140, Freiburg, Germany, 1998.

[4] Ayache, N.: *Artificial Vision for Mobile Robots*. MIT Press, Cambridge, 1991.

[5] Ayache, N. and C. Hansen: *Rectification of images for binocular and trinocular stereovision*. In *Internaltional Conference on Pattern Recognition*, pages 11–16, 1988.

[6] Bartoli, A: *On the Non-Linear Optimization of Projective Motion Using Minimal Parameters*. In *European Conference on Computer Vision*, pages 340–354, Copenhagen, Denmark, May 2002.

[7] Bartoli, A: *A Unified Framework for Quasi-Linear Bundle Adjustment*. In *The Sixteenth IAPR International Conference on Pattern Recognition*, pages 560–563, Quebec, Canada, 2002.

[8] Brown, D.: *The Bundle Adjustment - Progress and Prospect*. XIII Congress of the ISPRS, Helsinki, 1976.

[9] Burington, R.S. and D.C. May: *Handbook of Probability and Statistics with Tables*. McGraw-Hill Book Company, 1970.

[10] Carter, E.(S.): *Generating Gaussian Random Numbers*. http://www.taygeta.com/random/gaussian.html, 2002.

[11] Chen, Q. and G. Medioni: *Efficient Iterative Solution to M-View Projective Reconstruction*. In *European Conference on Computer Vision*, 1996.

[12] Csurka, G., C. Zeller, Z. Zhang and O. Faugeras: *Characterizing the Uncertainty of the Fundamental Matrix*. Computer Vision and Image Understanding, 68(1):18–36, 1997.

[13] Curless, B. and M. Levoy: *A Volumetric Method for Building Complex Models from Range Images*. In *SIGGRAPH'96*, 1996.

[14] Demirdjian, D., A. Zisserman and R. Horaud: *Stereo autocalibration from one plane*. In al., A. Heyden et (editor): *6th European Conference on Computer Vision*, volume II, pages 625Ű–639, Dublin, 2000. Springer-Verlag, Berlin, Heidelberg.

[15] Faugeras, O.: *Three-Dimensional Computer Vision : A Geometric Viewpoint (Artificial Intelligence)*. The MIT Press, 1999.

[16] Faugeras, O. and Q.-T. Luong: *The Geometry of Multiple Images*. The MIT Press, 2001.

[17] Faugeras, O., Q.-T. Luong and S. Maybank: *Camera Self-Calibration: Theory and Experiments*. In *European Conference on Computer Vision, LNCS 588*, pages 321–334. Springer-Verlag, 1992.

[18] Fitzgibbon, A.W. and A. Zisserman: *Automatic Camera Recovery for Closed or Open Image Sequences*. In *European Conference on Computer Vision*, pages 311–326, Freiburg,Germany, 1998.

[19] Golub, Gene H. and Charles F. Van Loan: *Matrix Computations*. The Johns Hopkins University Press, 1996.

[20] Harris, C. and M. Stephens: *A Combined Corner and Edge Detector*. In *4th Alvey Vision Conference*, pages 189–192, 1988.

[21] Hartley, R.: *Euclidean reconstruction from uncalibrated views*. Applications of Invariance in Computer Vision, 825:237–256, 1994.

[22] Hartley, R.: *Projective Reconstruction from Line Correspondences.* In *IEEE Conference on Computer Vision and Pattern Reognition.* IEEE Computer Society Press, 1994.

[23] Hartley, R.: *Self-Calibration from Multiple Views with a Rotating Camera.* In *European Conference on Computer Vision, LNCA 800/801*, pages 471–478. Springer-Verlag, 1994.

[24] Hartley, R.: *A Linear Method for Reconstruction from Points and Lines.* In *International Conference on Computer Vision*, pages 882–887, 1995.

[25] Hartley, R.: *In defence of the 8-point algorithm.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(6):580–593, 1997.

[26] Hartley, R.: *Minimizing Algebraic Error.* In *International Conference on Computer Vision*, pages 469–476, Bombay, India, 1998. IEEE, Narosa Publishing House.

[27] Hartley, R. and P. Sturm: *Triangulation.* Computer Vision and Image Understanding, 68(2), 1997.

[28] Hartley, R. and A. Zisserman: *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2001.

[29] Heyden, A., and K. Aström: *Euclidean Reconstruction from Constant Intrinsic Parameters.* In *13th International Conference on Pattern Recognition*, pages 339–343. IEEE Computer Society Press, 1996.

[30] Heyden, A.: *Projective Structure and Motion from Image Sequences Using Subspace Methods.* In *Scandinavian Conference on Image Analysis*, pages 963–968, 1997.

[31] Heyden, A. and K. Aström: *Euclidean Reconstruction from Image Sequences with Varying and Unknown Focal Length and Principal Point.* In *CVPR*, pages 438–443, San Juan, Puerto Rico, 1997. IEEE Computer Society.

[32] Heyden, A., R. Berthilsson and G. Sparr: *An iterative factorization method for projective structure and motion from image sequence.* Image and Vision Computing, 17(13):981–991, 1999.

[33] Irani, M. and P. Anandan: *Factorization with Uncertainty.* In Vernon, D. (editor): *European Conference on Computer Vision*, pages 539–553, Dublin, Ireland, 200. Springer-Verlag, Berlin, Heidelberg.

[34] Kinatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier, 1996.

[35] Knight, J. and I. Reid: *Self-calibration of a stereo-rig in a planar scene by data combination.* In *International Conference on Pattern Recognition*, pages 411–414, 2000.

[36] Koch, R.: *3-D Surface Reconstruction from Stereoscopic Image Sequences.* In *Fifth International Conference on Computer Vision*, pages 109–114. IEEE Computer Society Press, 1995.

[37] Lampton, M.: *Damping-Undamping Strategies for the Levenberg-Marquardt Nonlinear Least-Squares Method.* Computers in Physics, 11(1), 1996.

[38] Laveau, S.: *Géométrie d'un système de N caméras. Théorie, estimation et applications.* PhD thesis, 1996.

[39] Lewis, J.P.: *Fast Template Matching.* Vision Interface, pages 120–123, 1995.

[40] Liu, B., D. Maier, M. Schill and R. Männer: *Robust Real-time Tracking of Surgical Instruments in the Eye Surgery Simulator (EyeSi).* In *The Fourth IASTED International Conference on Signal and Image Processing, SIP 2002*, pages 441–446, Kauai, Hawaii, USA, August 2002.

[41] Liu, B. and R. Männer: *A Linear Iterative Least-Squares Method for Estimating the Fundamental Matrix.* In *The Seventh International Symposium on Signal Processing and its Applications (IEEE-ISSPA 2003)*, Paris, France, 2003.

[42] Liu, B., M. Yu, D. Maier and R. Männer: *Accelerated Bundle Adjustment in Multiple-View Reconstruction.* In Palade, V., R.J. Howlett and L.C. Jain (editors): *Seventh International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, Oxford, UK, September 2003.

[43] Liu, B., M. Yu, D. Maier and R. Männer: *An Efficient and Accurate Method for 3D-Point Reconstruction from Multiple Views.* Submitted to International Journal of Computer Vision, 2003.

[44] Luong, Q.T. and O. Faugeras: *The Fundamental Matrix: Theory, Algorithms, and Stability Analysis*. The International Journal of Computer Vision, 17(1):43–76, 1995.

[45] Luong, Q.T. and T. ViŽville: *Canonical Representations for the Geometries of Multiple Projective Views*. Computer Vision and Image Understanding, 64(2):193–229, 1996.

[46] Mahamud, S., M. Herbert, Y. Omori and J. Ponce: *Provably-Convergent Iterative Methods for Projective Structure and Motion*. In *CVPR*, 2001.

[47] Ma, Y., J. Kosecka and S. S. Sastry: *Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation*. International Journal of Computer Vision, 44(3):219–249, 2001.

[48] Ma, Y., R. Vidal, S. Hsu and S. Sastry: *Optimal Motion Estimation from Multiple Images by Normalized Epipolar Constraint*. Journal of Communications in Information and Systems, (1):51–73, 2001.

[49] Moons, T., L. Van Gool, M. Proesmans and E. Pauwels: *Affine reconstruction from perspective image pairs with a relative object-camera translation in between*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(1):77–83, 1997.

[50] More, J.: *The Levengerg-Marquadt Algorithm, Implementation and Theory. Numerical Analysis*, Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

[51] Morris, D.D. and T. Kanade: *A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertain Models*. In *ICCV*, pages 696–702, 1998.

[52] Nocedal, J. and S.J. Wright: *Numerical Optimization*. Springer-Verlag, 1999.

[53] Pollefeys, M.: *Visual 3D Modeling from Images*. Tutorial notes, tutorial organized in conjunction with ECCV 2000, Dublin, Ireland, June, 2002.

[54] Pollefeys, M. and L. Van Gool: *Self-calibration from the absolute conic on the plane at infinity*. In *Computer Analysis of Images and Patterns*, volume 1296, pages 175–182. Lecture Notes in Computer Science, Springer-Verlag, 1997.

[55] Pollefeys, M., L. Van Gool and A. Oosterlinck: *The Modulus Constraint: A New Constraint for Self-Calibration.* In *13th International Conference on Pattern Recognition*. IEEE Computer Soc. Press.

[56] Pollefeys, M., R. Koch and L. Van Gool: *Self Calibration and Metric Reconstruction in Spite of Varying and Unknown Intrinsic Camera Parameters.* In *CVPR*, pages 90–96, Bombay, India, 1998.

[57] Pollefeys, M., R. Koch and L. Van Gool: *A simple and efficient rectification method for general motion.* In *ICCV*, pages 496–501, Corfu, Greece, 1999.

[58] Quan, L., A. Heyden and F. Kahl: *Minimal Projective Reconstruction with Missing Data.* In *CVPR*, pages 210–216, Fort Collins, Colorado, 1999.

[59] Quan, L. and T. Kanade: *A Factorization Method for Affine Structure from Line Correspondences.* In *CVPR*, pages 803–808, San Francisco, CA, 1996.

[60] Schaffalitzky, F., A. Zisserman, R. I. Hartley and P.H.S. Torr: *A Six Point Solution for Structure and Motion.* In *European Conference on Computer Vision*, pages 632–648, Dublin,Ireland, 2000.

[61] Shashua, A.: *A Six Point Solution for Structure and Motion.* In *Computer Vision - ECCV'94*, volume 801, pages 479–484. Lecture Notes in Computer Science, Springer-Verlag, 1994.

[62] Shi, J. and C. Tomasi: *Good Features to Track.* In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, 1994.

[63] Shum, H.-Y., Q. Ke and Z. Zhang: *Efficient Bundle Adjustment with Virtual Key Frames: A Hierarchical Approach to Multi-frame Structure from Motion.* In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2538–2543, Fort Collins, Colorado, 1999.

[64] Slama, C.: *Manual of Photogrammetry.* Falls Church, VA, USA, 1980.

[65] Spetsakis, M. and J. Aloimonos: *Structure from Motion Using Line Correspondences.* International Journal of Computer Vision, 4(3):171–183, 1990.

[66] Tomasi, C. and T. Kanade: *Shape and Motion from Image Streams under Orthography: a Factorization Method.* International Journal of Computer Vision, 9(2):137–154, 1992.

[67] Tommasini, T., A. Fusiello, E. Trucco and V. Roberto: *Making Good features Track Better*. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 178–183, Santa Barbara, USA, 1998.

[68] Torr, P. and A. Zisserman: *Robust parametrization and computation of the trifocal tensor*. Image and Vision Computing, 15:591–605, 1997.

[69] Torr, P. and A. Zisserman: *Robust Computation and Parameterization of Multiple View Relations*. In *International Conference on Computer Vision*, pages 727–732, San Francisco,CA, 1998. Narosa Publishing House.

[70] Torr, P., A. Zisserman and S. Maybank: *Robust Detection of Degenerate Configurations for the Fundamental Matrix*. In *The 5th International Conference on Computer Vision*, pages 1037–1042, Boston, MA, 1995. IEEE Computer Society Press.

[71] Triggs, B.: *Factorization methods for projective structure and motion*. In *CVPR*, pages 845–851, San Francisco,CA, 1996.

[72] Triggs, B.: *Auto-Calibration and the Absolute Quadric*. In *CVPR*, pages 609–614, 1997.

[73] Triggs, B.: *Autocalibration from Planar Scenes*. In *The 5th European Conference on Computer Vision*, Freiburg, Germany, 1998.

[74] Triggs, B.: *Plane + Parallax, Tensors and Factorization*. In *European Conference on Computer Vision*, pages 522–538, Dublin,Ireland, 2000.

[75] Triggs, B., P. McLauchlan, R. Hartley and A. Fitzgibbon: *Bundle Adjustment A Modern Synthesis*. In *International Workshop on Vision Algorithms: theory and Practice*, pages 864–884, Corfu, Greece, 1999.

[76] Turk, G. and M. Levoy: *Zippered Polygon Meshes from Range Images*. In *SIGGRAPH'94*, pages 311–318, 1994.

[77] Tuytelaars, T. and L. Van Gool: *Wide Baseline Stereo based on Local, Affinely invariant Regions*. In *British Machine Vision Conference*, pages 412–422, 2000.

[78] Vidal, R., Y. Ma, S. Hsu and S. Sastry: *Optimal Motion Estimation from Multiview Normalized Epipolar Constraint*. In *International Conference on Computer Vision*, volume 1, pages 34–41, Vancouver, Canada, 2001.

[79] Weng, J., T.S. Huang and N. Ahuja: *Optimal Motion and Structure Estimation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(9):864–884, 1993.

[80] Zhang, Z.: *Motion and Structure from Two Perspective Views: from Essential Parameters to Euclidean Motion via Fundamental Matrix.* Journal of the Optical Society of America A, 1997.

[81] Zhang, Z.: *Determining the Epipolar Geometry and its Uncertainty: A Review.* The International Journal of Computer Vision, 27(2):161–195, 1998.

[82] Zhang, Z., R. Deriche, O. Faugeras and Q.-T. Luong: *A Robust Technique for Matching Two Uncalibrated Images though the Recovery of the Unknown Epipolar Geometry.* Artificial Intelligence Journal, 78:87–119, 1995.

[83] Zhang, Z. and O. Faugeras: *3D Dynamic Scene Analysis.* Springer-Verlag, 1992.

[84] Zhang, Z. and C. Loop: *Estimating the Fundamental Matrix by Transforming Image Points in Projective Space.* Computer Vision and Imaging Understanding, 82(2):174–180, 2001.

[85] Zhang, Z. and Y. Shan: *Incremental Motion Estimation Through Local Bundle Adjustment.* Technical Report MSR-TR-01-54, 2001.

[86] Zisserman, A., P. Beardsley and I. Reid: *Metric Calibration of a Stereo Rig.* In *IEEE Workshop on Representation of Visual Scenes*, pages 93–100, Boston, 1995.