Metric 3D-reconstruction from Unordered and Uncalibrated Image Collections

Viktor Larsson

Master's thesis
2013:E42

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

Abstract

In this thesis the problem of Structure from Motion (SfM) for uncalibrated and unordered image collections is considered. The proposed framework is an adaptation of the framework for calibrated SfM proposed by Olsson-Enqvist (2011) to the uncalibrated case.

Olsson-Enqvist's framework consists of three main steps; pairwise relative rotation estimation, rotation averaging, and geometry estimation with known rotations. For this to work with uncalibrated images we also perform auto-calibration during the first step.

There is a well-known degeneracy for pairwise auto-calibration which occurs when the two principal axes meet in a point. This is unfortunately common for real images. To mitigate this the rotation estimation is instead performed by estimating image triplets. For image triplets the degenerate configurations are less likely to occur in practice. This is followed by estimation of the pairs which did not get a successful relative rotation from the previous step.

The framework is successfully applied to an uncalibrated and unordered collection of images of the cathedral in Lund. It is also applied to the well-known Oxford dinosaur sequence which consists of turntable motion. Image pairs from the turntable motion are in a degenerate configuration for auto-calibration since they both view the same point on the rotation axis.

Preface

This master thesis was written for the fulfillment of the thesis requirement for my Master of Science in Computer Science degree at Lund Institute of Technology (LTH).

This thesis concludes my undergraduate studies at LTH and ends a chapter of my life that I will look back upon fondly. First I would like to thank the faculty at the Centre for Mathematical Sciences for sparking my interest for mathematics through the various courses and projects that I have completed during these five years.

I would like to thank Rebecka Nyqvist, Bertil Larsson and Jesper Friholm for proofreading my report and giving valuable feedback and support.

I would also like to thank my supervisor Carl Olsson for sharing his infinite wisdom regarding all things Computer Vision related, for proofreading my thesis and for offering valuable advice and discussions.

Viktor 2013

Contents

Chapter 1

Introduction

This thesis deals with the problem of *Structure from Motion*, which is determining 3D structure from 2D images. The structure refers to a model of the 3D scene which is seen in the images. In this thesis will consider the case when the structure is modeled by point clouds, and we will refer to the individual 3D points as *structure points*. Motion refers to the relative position and orientation of the cameras which captured the images. The cameras will be modeled using the pinhole camera model.

For building accurate 3D reconstructions we will require some additional knowledge of the cameras beyond just their position and relative orientation. This knowledge is called the camera's *calibration*. The most important parameter is the camera's focal length. The focal length is the relative distance from the image sensor to the lens. Without knowledge of the camera's calibration the 3D scene can only be determined up to a projective ambiguity. In practice this means we are e.g. unable to determine the real angles between lines.

This thesis specifically considers the case of unordered image collections. By *unordered* we mean that we have no knowledge of which images were taken close to each other or even which view the same parts of the scene. Thus ideally no preference in the framework should be given based on the order of the images. The opposite is *ordered* image collections which often consists of image sequences with a small movement of the camera between each consecutive image.

In [20] Olsson and Enqvist propose a framework for structure from motion for unordered image collections. The framework depends on the cameras having known calibration from a previous offline calibration. This thesis will adapt the Olsson-Enqvist framework to the uncalibrated case by performing autocalibration using the images.

1.1 Notation

Vectors will be written with bold font (e.g. $\mathbf{x}$, $\mathbf{X}$, $\mathbf{a}$) and are assumed to be column vectors unless otherwise noted. Matrices will be denoted by uppercase letters (e.g. P, A, H). Once homogeneous objects are defined (see Section 2.1) we let equality signs between homogeneous objects denote equality up to scale unless otherwise noted.

1.2 Related work

In recent years there have been many proposed frameworks for structure from motion for unordered image collections. Some examples are Olsson and Enqvist [20], Snavely et. al [22] and Crandall et al. [3]. The problem of camera calibration is usually solved by assuming that it is known from a previous offline calibration or by using the focal length field in the EXIF tags as an initial guess and then improving it with bundle adjustment. This has the inherent problem of not working for images which have no EXIF tags or offline calibration is impossible.

There also exist frameworks for uncalibrated image sets which start by building a projective reconstruction and then perform auto-calibration. One example is the framework proposed by Pollefeys et. al in [21] where a projective reconstruction is built incrementally and then upgraded to a metric reconstruction by finding the absolute quadric Q_∞^*. A more recent example can be seen in Chen et. al [16] where the projective reconstruction is created hierarchically by merging smaller projective reconstructions. Similarly to [21] the projective reconstruction is upgraded to metric by estimating the absolute quadric.

1.3 Notes about implementation

The implementation was written in MATLAB using Olsson and Enqvist's code as a starting point. The implementation uses the MOSEK [1] library for solving linear programs and the point correspondences are found using the Lowe's SIFT implementation from [19].

Chapter 2

Background

We start off with some of the background needed to define the problem we are interested in. First comes a brief review of projective geometry. Then the pinhole camera model is introduced and the advantage of the projective geometry becomes apparent. Finally we discuss the difference between projective and metric reconstruction.

The interested (or confused) reader is referred to [11] which gives a more thorough review of the subjects.

2.1 Projective geometry

One of the tools we will need is projective geometry. Projective geometry deals with the geometry in a *projective space*. One such space is the *projective plane* denoted $\mathbb{P}^2$. The projective plane is an extension of the real plane $\mathbb{R}^2$.

The projective plane $\mathbb{P}^2$ can be thought of as equivalence classes in $\mathbb{R}^3\backslash\{0\}$ where two points $\mathbf{x}$ and $\mathbf{y}$ are equivalent if there exist some $\lambda \neq 0$ such that $\mathbf{x} = \lambda\mathbf{y}$. This means that the point $(x, y, z)^T$ denotes the same point as $(2x, 2y, 2z)^T$ and $(\lambda x, \lambda y, \lambda z)^T$.

This representation is called *homogeneous coordinates*. From now on we will let the equality sign denote equality up to scale when comparing homogeneous objects, e.g we can write $(1, 2, 3)^T = (2, 4, 6)^T$ which is true in $\mathbb{P}^2$.

To interpret points in $\mathbb{P}^2$ we let the point representatives with $z = 1$ correspond to points in the real plane $\mathbb{R}^2$, i.e.

$$(x, y) \in \mathbb{R}^2 \leftrightarrow (x, y, 1) \in \mathbb{P}^2. \tag{2.1}$$

Due to the scale invariance we have that $(x, y, 1) = (\lambda x, \lambda y, \lambda)$ in $\mathbb{P}^2$. Thus for any point $(x, y, z) \in \mathbb{P}^2$ we can find the corresponding point in $\mathbb{R}^2$ by dividing by the third coordinate, i.e. $(x, y, z) = (x/z, y/z, 1)$.

This becomes troublesome when the third coordinate is zero. Points where $z = 0$ does not correspond to any point in $\mathbb{R}^2$. In projective geometry these points are

called *ideal points* or *points at infinity*. To motivate this name consider

$$(x,\ y,\ \epsilon)^T = (x/\epsilon,\ y/\epsilon,\ 1)^T. \tag{2.2}$$

When ϵ goes to zero the point goes toward infinity in the direction $(x, y)^T$.

By adding coordinates it is simple to generalize the projective plane $\mathbb{P}^2$ to arbitrary dimensions, i.e. $\mathbb{P}^n$.

One of the advantages of homogeneous coordinates is that they allows for a larger class of transformations to be modeled using matrix multiplication, e.g. affine transformations. Affine transformations in $\mathbb{R}^n$ are on the form

$$\mathbb{R}^n \ni \mathbf{x} \mapsto A\mathbf{x} + \mathbf{t} \in \mathbb{R}^n. \tag{2.3}$$

The corresponding transformation in $\mathbb{P}^n$ can be written

$$\mathbb{P}^n \ni \begin{pmatrix} \lambda\mathbf{x} \\ \lambda \end{pmatrix} \mapsto \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} \lambda\mathbf{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} \lambda(A\mathbf{x} + \mathbf{t}) \\ \lambda \end{pmatrix} \in \mathbb{P}^n. \tag{2.4}$$

Another class of important transformations are *similarity transformations*. These consist of a scaling, rotation and translation of the points and can be written

$$H = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad \text{where} \quad R^T R = I, \quad \det R = 1, \quad s \in \mathbb{R}_+. \tag{2.5}$$

In general any invertible $(n+1) \times (n+1)$ matrix is called a *projective transformation*.

The homogeneous coordinates also allow for lines in $\mathbb{P}^2$ to be modeled using the scalar product. The equation for a line in $\mathbb{R}^2$ can be written

$$ax + by + c = 0. \tag{2.6}$$

In $\mathbb{P}^2$ we can write this as

$$(a,\ b,\ c) \begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix} = \lambda(ax + by + c) = 0. \tag{2.7}$$

Thus the point $\mathbf{x} = (x, y, z)^T \in \mathbb{P}^2$ belongs to the line $\mathbf{l} = (a, b, c)^T$ iff

$$\mathbf{l}^T\mathbf{x} = 0. \tag{2.8}$$

For $\mathbb{P}^3$ we instead have a similar construction for planes. In general we have that for $\mathbb{P}^n$ we can model $(n-1)$-dimensional hyperplanes using the scalar product.

2.2 The Pinhole camera model

The camera model we will use is the Pinhole camera model. A camera is a mapping which takes the 3D points to the image points. In the pinhole camera model we construct the line from the camera center to the 3D points. The

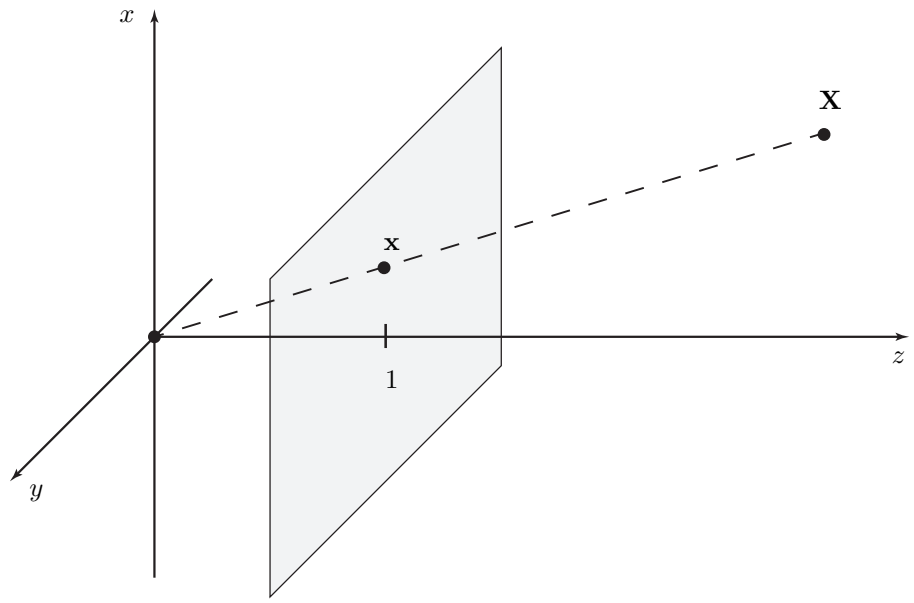


Figure 2.1: The pinhole camera model. The projection $\mathbf{x}$ is formed by taking the intersection of the plane $z = 1$ with the line going through both the origin and the 3D point $\mathbf{X}$.

intersection of this line and the image plane is taken as the projection. This can be seen in Figure 2.1.

Assume that the camera center lies at the origin and that the image plane is given by $z = 1$. To find the projection $\mathbf{x}$ of a 3D point $\mathbf{X} \in \mathbb{R}^3$ we construct the line going through both the origin and the 3D point,

$$l(\lambda) = \lambda \mathbf{X} = \lambda (x, y, z)^T. \tag{2.9}$$

The intersection with the plane $z = 1$ is then given by $\lambda z = 1 \Leftrightarrow \lambda = \frac{1}{z}$. Thus the projection on the image plane is given by

$$\mathbf{x} = \begin{pmatrix} x/z \\ y/z \end{pmatrix}. \tag{2.10}$$

Now if we consider the image points to be part of $\mathbb{P}^2$ and the structure points to be part of $\mathbb{P}^3$ we see that due to the scale invariance

$$\mathbf{x} = \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{2.11}$$

Thus in the projective setting the camera can be modeled using a *camera matrix*

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}. \tag{2.12}$$

This was under the assumption that the camera center was at the origin and the image plane was at $z = 1$. For general camera centers and image planes we simply translate and rotate the points such that this holds, i.e. $P = [R\ \mathbf{t}]$. Cameras that are on this form are called *calibrated cameras*.

To generalize even further we don't need to restrict ourselves to simply rotating the points but can have an arbitrary transformation, i.e $P = [A\ \mathbf{t}]$. Cameras on this form can always be factorized as $P = K[R\ \mathbf{t}]$ where K is upper triangular with positive diagonal elements using QR-factorization.

The matrix K is called the calibration matrix. The calibration matrix contains the camera's *intrinsic parameters*; focal length f, aspect ratio α, skew s and principal point (u_0, v_0).

$$K = \begin{bmatrix} \alpha f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.13}$$

The camera's *extrinsic parameters* consist of the translation and rotation.

2.3 Metric reconstruction

Given a set of images of a static scene a *projective reconstruction* is a set of cameras and 3D points, $\{P, \mathbf{X}\}$, such that

$$\mathbf{x}_j^i = P_i \mathbf{X}_j, \tag{2.14}$$

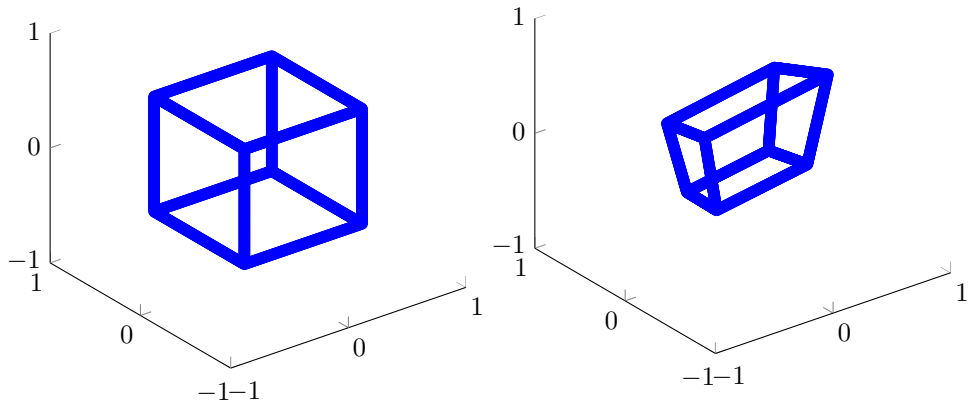
where $\mathbf{x}_j^i$ is the image point corresponding to the jth 3D point seen in the ith camera.

Let $\{P, \mathbf{X}\}$ be a projective reconstruction and H an arbitrary projective transform. Then $\{PH,\ H^{-1}\mathbf{X}\}$ is another projective reconstruction since

$$(P_i H)(H^{-1} X_j) = P_i \mathbf{X}_j = \mathbf{x}_j^i. \tag{2.15}$$

Thus for any scene there are infinitely many possible projective reconstructions which differ by a projective transformation. Since a projective transformation does not (in general) preserve angles, the structure points $\mathbf{X}$ can look quite different than the "real world" scene. See Figure 2.2.

We are interested in a reconstruction where the structure points are as close to the "real world" scene as possible. Since there is no clear choice of coordinate system in the "real world" we want it to differ with at most a similarity transform (rotation, scaling, translation) from the "real" scene. Such a reconstruction is called a *metric reconstruction*.



(a) A Cube

(b) The same cube after a projective transformation.

Figure 2.2

Chapter 3

Problem formulation

Given images of a static scene and putative pairwise point correspondences we want to estimate each camera's focal length and find a metric reconstruction describing the scene under the assumption that the cameras have zero skew, unit aspect ratio and known principal point. See Figure 3.1.

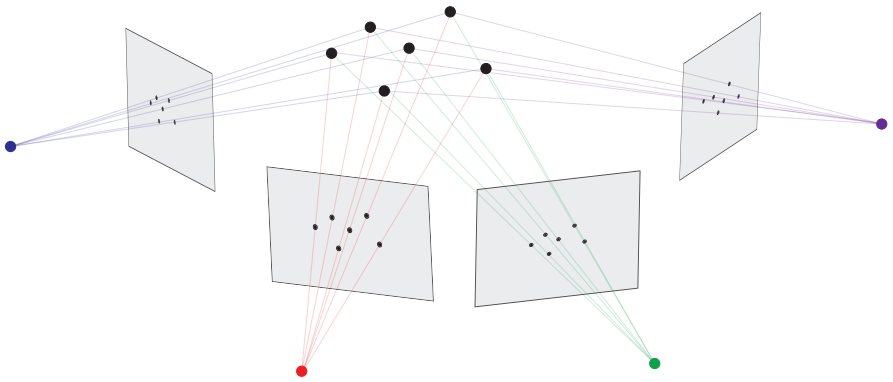


Figure 3.1: The Structure from Motion problem. Given point correspondences between images we want to find both the structure points and the relative position and orientation of the cameras.

Let $\mathbf{x}_j^i$ denote the jth structure point seen in the ith image. We want to find a reconstruction such that

$$\mathbf{x}_j^i = P_i \mathbf{X}_j = K_i[R_i \ \mathbf{t}_i]\mathbf{X}_j \quad \forall i, j, \tag{3.1}$$

where $K_i = \text{diag}(f_i, f_i, 1)$ and $R_i^T R_i = I$. Due to measurement noise in the image points it is likely that no reconstruction exists that satisfy (3.1) exactly for all image points. Instead we search for a solution which minimizes the reprojection error, i.e.

$$\min_{\{P,\mathbf{X}\}} \sum_{i,j} \left|\left| \left(x_j^i - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j}, \ y_j^i - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right) \right|\right|^2 . \tag{3.2}$$

3.1 Calibration assumptions

We assume that the cameras have zero skew, unit aspect ratio and known principal point. Since the principal point is known we can w.l.o.g. then assume that the calibration matrices K have the form $K = \text{diag}(f, f, 1)$ since the principal point can be moved to the origin otherwise.

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_p} \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f} \tag{3.3}$$

Transforming the image points with K_p^{-1} moves ths principal point to the origin. To see this consider

$$\mathbf{x} = K_p K_f [R \ \mathbf{t}] \quad \Leftrightarrow \quad K_p^{-1}\mathbf{x} = K_f [R \ \mathbf{t}]. \tag{3.4}$$

If we now consider $\tilde{\mathbf{x}} = K_p^{-1}\mathbf{x}$ to be the new image points they will correspond to the camera with calibration matrix K_f.

Chapter 4

Theory

Now follows a review of some of the theory that will be used in the rest of the thesis. On a first reading the unfamiliar reader might want to skip some of the details and derivations.

First we review some multiple view geometry and define the fundamental matrix and trifocal tensor. The fundamental matrix for a pair of uncalibrated cameras was introduced simultaneously by Hartley [10, 9] and Faugeras [4, 5]. The trifocal tensor (in matrix form) for a triplet of uncalibrated cameras was identified by Hartley in [7]. These objects are useful because they allow us to create an initial projective reconstruction from image pairs and triplets. By putting constraints on the projections they also help us decide which point correspondences are outliers.

Then follows a short introduction to conics and their generalizations to projective spaces and to higher dimensions. These are tools that will be needed when deriving methods for auto-calibration.

Finally in the section *geometry at infinity* we consider some of the conics and quadrics which are useful in auto-calibration. The absolute conic Ω_∞ was first used in computer vision by Faugeras and Maybank [6] and the absolute quadric was introduced by Triggs in [24].

For a more in-depth review of these subjects the reader is referred to [11].

4.1 Multiple view geometry

4.1.1 Fundamental matrix

In this section we will derive the fundamental matrix which captures the geometry of two cameras viewing a scene. The fundamental matrix puts constraints on the projections of 3D points seen in two views. These constraints are used to decide which point correspondences are outliers and to estimate the fundamental matrix. From the fundamental matrix we can also extract a pair of

cameras consistent with it. This allows us to form an initial projective reconstruction.

Assume we have a scene captured by two cameras and that the cameras are on the form $P_1 = [I\ \mathbf{0}]$ and $P_2 = [A\ \mathbf{t}]$. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be observations of the same 3D-point $\mathbf{X}$ in the two views.

First consider $\mathbf{x}_1$. This point will backproject to a line

$$\mathbf{l}(s) = \begin{pmatrix} \mathbf{x_1} \\ s \end{pmatrix}. \tag{4.1}$$

This line in 3D-space will project down to a line in the second image. Since the unknown 3D-point $\mathbf{X}$ must lie on the line $\mathbf{l}$, the second observation $\mathbf{x}_2$ must lie on the projection of the line. See Figure 4.1.

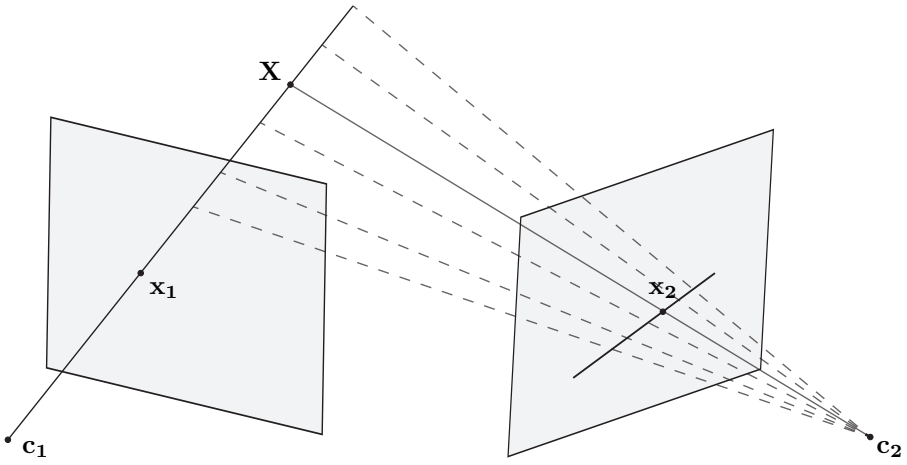


Figure 4.1: Epipolar geometry. The image point $\mathbf{x_1}$ backprojects to a line in 3D-space. Somewhere on this line lies the unknown 3D-points $\mathbf{X}$. The line downprojects onto a line in the second image which must contain the true projection $\mathbf{x}_2$.

We can find the projection of the line by considering the two extremes, $s = 0$ and $s = \infty$,

$$P_2\mathbf{l}(0) = A\mathbf{x}_1 \quad \text{and} \quad P_2\mathbf{l}(\infty) = \mathbf{t}. \tag{4.2}$$

The line through these points is given by $\mathbf{t} \times A\mathbf{x}_1 = [\mathbf{t}]_\times A\mathbf{x}_1$ where $[\mathbf{t}]_\times$ is the 3×3 matrix which captures the linear function $\mathbf{x} \mapsto \mathbf{t} \times \mathbf{x}$. Since $\mathbf{x}_2$ must lie on this line we get the condition

$$0 = \mathbf{x}_2^T [\mathbf{t}]_\times A\mathbf{x}_1 = \mathbf{x}_2^T F\mathbf{x}_1, \tag{4.3}$$

where $F = [\mathbf{t}]_\times A$ is the *fundamental matrix*.

Extracting cameras from F

Now we consider the problem of finding cameras P_1 and P_2 such that they correspond to a given fundamental matrix F. Similarly to the previous section

we choose the first camera as $P_1 = [I\ \mathbf{0}]$. The *epipole* is the projection of the other camera's camera center. Let $\mathbf{e}_2$ be the epipole in the second image. For the epipole $\mathbf{e}_2$ it holds that $\mathbf{e}_2^T F = 0$. Then the second camera can be chosen as $P_2 = [[\mathbf{e}_2]_\times F\ \mathbf{e}_2]$. To verify that this is correct we consider the projections of a 3D point $\mathbf{X} = \begin{pmatrix} \mathbf{x} \\ s \end{pmatrix}$.

$$\mathbf{x}_1 = P_1\mathbf{X} = \mathbf{x}, \quad \mathbf{x}_2 = P_2\mathbf{X} = [\mathbf{e}_2]_\times F\mathbf{x} + s\mathbf{e}_2, \tag{4.4}$$

and check that these points satisfy the epipolar constraint

$$\mathbf{x}_2^T F\mathbf{x}_1 = (\mathbf{x}^T F^T [\mathbf{e}_2]_\times^T + s\mathbf{e}_2^T)F\mathbf{x} = \underbrace{(F\mathbf{x})^T [\mathbf{e}_2]_\times^T (F\mathbf{x})}_{F\mathbf{x}\cdot(\mathbf{e}_2\times F\mathbf{x})=0} + s\underbrace{(\mathbf{e}_2^T F)}_{=0}\mathbf{x} = 0. \tag{4.5}$$

Projective invariance

The fundamental matrix is invariant to projective transformations of the cameras and 3D-points. This follows directly from the projective ambiguity for reconstructions. Let H be an arbitrary projective transformation and $\{P,\ \mathbf{X}\}$ a projective reconstruction. Since $\{PH,\ H^{-1}\mathbf{X}\}$ has the same projections as the first pair they must have the same fundamental matrix.

It is the projective invariance that allows us to w.l.o.g. assume that the first camera is on the form $P = [I\ \mathbf{0}]$. To see this consider a pair of cameras

$$P_1 = [A_1\ \mathbf{t}_1] \quad \text{and} \quad P_2 = [A_2\ \mathbf{t}_2]. \tag{4.6}$$

Transforming this pair with the transformation

$$H = \begin{bmatrix} A_1^{-1} & -A^{-1}\mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.7}$$

we get the pair

$$\tilde{P}_1 = P_1 H = [I\ \mathbf{0}] \quad \text{and} \quad \tilde{P}_2 = P_2 H = [\underbrace{A_2 A_1^{-1}}_{A}\ \underbrace{(\mathbf{t_2} - A_2 A_1^{-1}\mathbf{t}_1)}_{\mathbf{t}}] = [A\ \mathbf{t}]. \tag{4.8}$$

Since the pair $(\tilde{P}_1, \tilde{P}_2)$ only differs with a projective transformation from the pair (P_1, P_2) it will have the same fundamental matrix.

Note that the fundamental matrix is however not invariant to transformations of the image points. If we premultiply the cameras with matrices T_1 and T_2 the corresponding fundamental matrix for the new pair of cameras is

$$\hat{F} = T_2^{-T} F T_1^{-1}. \tag{4.9}$$

Essential matrix

The *essential matrix* describes the two view geometry for two calibrated cameras and is closesly related to the fundamental matrix. Let $P_1 = [I\ \mathbf{0}]$ and $P_2 = [R\ \mathbf{t}]$, then the essential matrix is given by $E = [\mathbf{t}]_\times R$.

If K_1 and K_2 are the calibration matrices for the camera pair (P_1, P_2) and F is the corresponding fundamental matrix then

$$E = K_2^T F K_1. \tag{4.10}$$

Similarly to the fundamental matrix it is possible to extract cameras from the essential matrix.

Planar motion

Now we consider the special case where the cameras have undergone planar motion, i.e. the camera centers and principal axes lie in a plane. We consider specifically translation in the xz-plane and rotation around the y-axis. As usual we let $P_1 = [I\ \mathbf{0}]$. The second camera $P_2 = [R\ \mathbf{t}]$ will have a special structure due to the planar motion with

$$R = \begin{bmatrix} \times & 0 & \times \\ 0 & 1 & 0 \\ \times & 0 & \times \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} \times \\ 0 \\ \times \end{pmatrix}, \tag{4.11}$$

where $\times$ denote a possibly non-zero element. The essential matrix will then have the structure

$$E = [\mathbf{t}]_\times R = \begin{bmatrix} 0 & \times & 0 \\ \times & 0 & \times \\ 0 & \times & 0 \end{bmatrix} \begin{bmatrix} \times & 0 & \times \\ 0 & 1 & 0 \\ \times & 0 & \times \end{bmatrix} = \begin{bmatrix} 0 & \times & 0 \\ \times & 0 & \times \\ 0 & \times & 0 \end{bmatrix}. \tag{4.12}$$

For the uncalibrated cameras $K_1[I\ \mathbf{0}]$ and $K_2[R\ \mathbf{t}]$ the fundamental matrix will have the same structure under the assumption that K_1 and K_2 are diagonal matrices.

Properties of the fundamental matrix

We now list some properties of the fundamental matrix F.

- F is a 3×3 matrix with rank 2. This implies that $\det F = 0$.
- The epipoles lie in the left and right nullspaces, i.e. $F\mathbf{e}_1 = \mathbf{e}_2^T F = 0$.
- If F is the fundamental matrix for the pair (P_1, P_2) then F^T is the fundamental matrix for the pair (P_2, P_1).
- F has 7 degrees of freedom while E has 5.

4.1.2 Trifocal tensor

Now we consider a scene viewed from three cameras. The object for three views that corresponds to what the fundamental matrix is for two view is called the *trifocal tensor*. The most basic constraint for the trifocal tensor is for a line in $\mathbb{P}^3$ seen in three views.

Let $\mathbf{L}$ be a line in $\mathbb{P}^3$ and $P_1 = [I\ \mathbf{0}]$, $P_2 = [A\ \mathbf{a}_4]$, and $P_3 = [B\ \mathbf{b}_4]$ be three cameras viewing the line. Denote the projections of the line $\mathbf{L}$ in the cameras: $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$.

The three imaged lines backproject into three planes

$$\boldsymbol{\pi}_1 = P_1^T\mathbf{l}_1 = \begin{pmatrix}\mathbf{l}_1\\0\end{pmatrix}, \quad \boldsymbol{\pi}_2 = P_2^T\mathbf{l}_2 = \begin{pmatrix}A^T\mathbf{l}_2\\\mathbf{a}_4^T\mathbf{l}_2\end{pmatrix}, \quad \boldsymbol{\pi}_3 = P_3^T\mathbf{l}_3 = \begin{pmatrix}B^T\mathbf{l}_3\\\mathbf{b}_4^T\mathbf{l}_3\end{pmatrix}, \tag{4.13}$$

which intersect in the line $\mathbf{L}$. This is illustrated in Figure 4.2

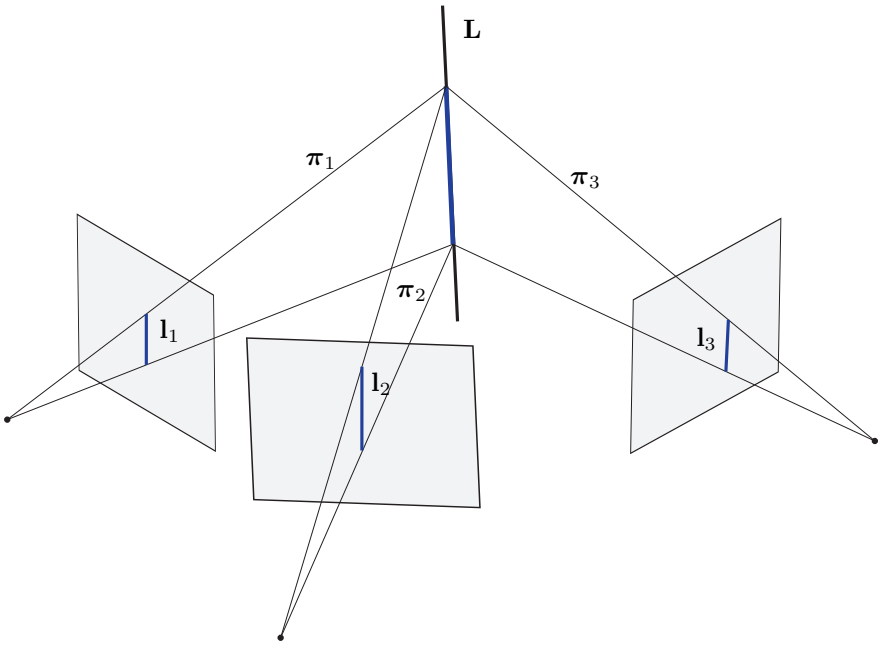


Figure 4.2: The line $\mathbf{L}$ which lies in 3D-space is projected down into three cameras. The projections $\mathbf{l}_1, \mathbf{l}_2$, and $\mathbf{l}_3$ backprojects onto three planes. The line $\mathbf{L}$ must lie in the intersection of the three planes.

Let M be the 4×3 matrix with the three planes as columns, i.e. $M = [\boldsymbol{\pi}_1\ \boldsymbol{\pi}_2\ \boldsymbol{\pi}_3]$. Then

$$\mathbf{X} \in \mathbf{L} \quad \Leftrightarrow \quad M^T\mathbf{X} = 0. \tag{4.14}$$

It follows that M must have a 2-dimensional left nullspace. (Note that if it was 1-dimensional it would correspond to a single point due to scale invariance.) Since M is a 4×3 matrix this means that M has rank 2. The three columns of M must then be linearly dependent, i.e. $\boldsymbol{\pi}_1 = \alpha\boldsymbol{\pi}_2 + \beta\boldsymbol{\pi}_3$ for some α, β.

For the last coordinate of the planes (4.13) this means that

$$0 = \alpha\mathbf{a}_4^T\mathbf{l}_2 + \beta\mathbf{b}_4^T\mathbf{l}_3 \quad \Leftrightarrow \quad \begin{cases}\alpha = \lambda\mathbf{b}_4^T\mathbf{l}_3\\ \beta = -\lambda\mathbf{a}_4^T\mathbf{l}_2\end{cases} \quad \text{for some } \lambda \in \mathbb{R} \tag{4.15}$$

From the top three coordinates we get (with equality up to scale)

$$\mathbf{l}_1 = \alpha A^T \mathbf{l}_2 + \beta B^T \mathbf{l}_3 = (\mathbf{b}_4^T \mathbf{l}_3) A^T \mathbf{l}_2 - (\mathbf{a}_4^T \mathbf{l}_2) B^T \mathbf{l}_3 = \mathbf{l}_3^T (\mathbf{b}_4 A^T) \mathbf{l}_2 - \mathbf{l}_2^T (\mathbf{a}_4 B^T) \mathbf{l}_3. \tag{4.16}$$

This equation can be simplified by introducing the matrices $T_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T$ for $i = 1, 2, 3$. Using these we get

$$\mathbf{l}_1 = \begin{bmatrix} \mathbf{l}_2^T T_1 \mathbf{l}_3 & \mathbf{l}_2^T T_2 \mathbf{l}_3 & \mathbf{l}_2^T T_3 \mathbf{l}_3 \end{bmatrix}^T. \tag{4.17}$$

The three matrices, T_1, T_2, and T_3 form the *trifocal tensor* (in matrix form). The trifocal tensor is sometimes denoted by $\mathcal{T}$.

Point-line-line correspondence

Now we choose a single point $\mathbf{X}$ on the line $\mathbf{L}$. Let $\mathbf{x}_1$ be the projection in the first image. Since it resides on the line in 3-space it must also lie on the projection on the line, i.e. $0 = \mathbf{l}_1^T \mathbf{x}_1 = \sum_i l_1^i x_1^i$ where we let x_1^i denotes the ith coordinate of $\mathbf{x}_1$ and similarly for the lines. Combining this with the line-line-line relationship (4.17) we get

$$0 = \sum_i (\mathbf{l}_2^T T_i \mathbf{l}_3) x_1^i = \mathbf{l}_2^T (\sum_i x_1^i T_i) \mathbf{l}_3, \tag{4.18}$$

which is the *point-line-line* correspondence for the trifocal tensor.

Point-line-point correspondence

Now we consider the situation where the projection of $\mathbf{X}$ is also known in the third view. The line $\mathbf{l}_2$ backprojects to the plane $\boldsymbol{\pi}_2$ which must contain $\mathbf{X}$. It is known that for points lying on a plane there exists a projective mapping which takes the projections in one view to another, i.e. there exist H such that

$$\mathbf{x}_3 = H \mathbf{x}_1. \tag{4.19}$$

The corresponding relationship between lines in the first and third view is given by

$$\mathbf{l}_1 = H^T \mathbf{l}_3. \tag{4.20}$$

From (4.17) we see that

$$H^T = \begin{bmatrix} \mathbf{l}_2^T T_1 \\ \mathbf{l}_2^T T_2 \\ \mathbf{l}_2^T T_3 \end{bmatrix} \quad \Rightarrow \quad H = \begin{bmatrix} T_1^T \mathbf{l}_2 & T_2^T \mathbf{l}_2 & T_3^T \mathbf{l}_2 \end{bmatrix}. \tag{4.21}$$

Inserting this into (4.19) we get

$$\mathbf{x}_3 = \begin{bmatrix} T_1^T \mathbf{l}_2 & T_2^T \mathbf{l}_2 & T_3^T \mathbf{l}_2 \end{bmatrix} \mathbf{x}_1 = (\sum_i x_1^i T_i^T) \mathbf{l}_2. \tag{4.22}$$

By transposing and taking the cross product with $\mathbf{x}_3$ on both sides we get

$$\mathbf{0}^T = \mathbf{x}_3^T [\mathbf{x}_3]_\times = \mathbf{l}_2^T (\sum_i x_1^i T_i) [\mathbf{x}_3]_\times, \tag{4.23}$$

which gives us the *point-line-point* correspondence for the trifocal tensor.

Point-point-point correspondence

In practice the most useful correspondence between three views is the *point-point-point* correspondence where we have a single structure point $\mathbf{X}$ seen in all three views.

Let $\mathbf{l}_2$ be any line passing through the projection $\mathbf{x}_2$. Then $\mathbf{l}_2 = \mathbf{x}_2 \times \mathbf{y}$ for some $\mathbf{y} \neq \mathbf{x}_2$. Using (4.23) we get

$$\mathbf{y}^T [\mathbf{x}_2]_\times (\sum_i x_1^i T_i)[\mathbf{x}_3]_\times = \mathbf{0}^T. \tag{4.24}$$

But since this must hold for any line $\mathbf{l}_2$ passing through $\mathbf{x}_2$ and thereby for any $\mathbf{y}$ and we get that

$$[\mathbf{x}_2]_\times (\sum_i x_1^i T_i)[\mathbf{x}_3]_\times = 0_{3\times 3}. \tag{4.25}$$

Properties of the trifocal tensor

We now list some properties of the trifocal tensor $\mathcal{T}$.

- Each T_i is a 3×3 matrix with rank 2.
- Similarly to the fundamental matrix we can compute a camera triplet consistent with a given $\mathcal{T}$.
- The trifocal tensor is invariant to projective transformations of 3D-space.
- Only four of the nine equations in (4.25) are linearly independent.
- The trifocal tensor $\mathcal{T}$ has 18 degrees of freedom.

4.2 Conics and quadrics

4.2.1 Conics

Conics (or point conics) in $\mathbb{R}^2$ are curves defined by a second order polynomial constraint on the coordinates (x, y) and can be written on the form

$$ax^2 + by^2 + cxy + dx + ey + f = 0. \tag{4.26}$$

This can be generalized to $\mathbb{P}^2$ by introducing a homogeneous coordinate z. Setting $(x, y)^T \mapsto (x/z,\ y/z)^T$ we get

$$ax^2 + by^2 + cxy + dxz + eyz + fz^2 = 0. \tag{4.27}$$

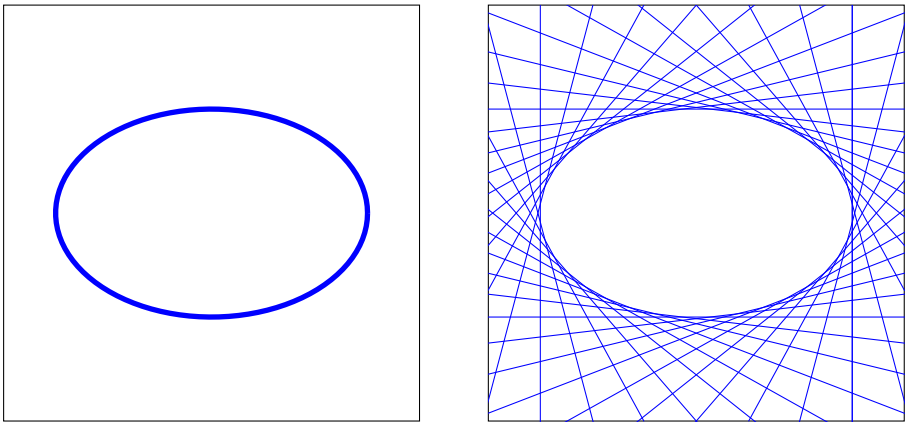
This is a quadratic form and can be expressed using a symmetric matrix.

$$\begin{pmatrix} x & y & z \end{pmatrix} \begin{bmatrix} a & c/2 & d/2 \\ c/2 & b & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = 0 \tag{4.28}$$

Every symmetric 3×3 matrix defines a conic in $\mathbb{P}^2$. Since the conic is a homogenous entity it is only defined up to scale. If the matrix is singular we say that the conic is degenerate. The non-degenerate conics are circles, ellipses, parabolas and hyperbolas. If a conic is degenerate it consists of either two lines (rank 2) or a repeated line (rank 1).

4.2.2 Dual conics

For every point conic there is a dual (line) conic which puts constraints on which lines lie tangent to the original point conic. One example of this can be seen in Figure 4.3. If the conic C is non-degenerate the symmetric matrix representing the dual conic C^* is equal to the inverse of C.



(a) Conic (b) Dual conic

Figure 4.3: The conic $x^2 + 2.25y^2 - 2.25z^2 = 0$ and its dual.

4.2.3 Polar lines

Let C be a non-degenerate conic and $\mathbf{x}$ be a point on the conic. Then $\mathbf{l} = C\mathbf{x}$ is the tangent line to C at $\mathbf{x}$. To see this we first note that the line is tangent if it only has a single intersection with the conic C. Now assume that $\mathbf{y}$ is another point on the conic which lies on the line $\mathbf{l} = C\mathbf{x}$. That is

$$\mathbf{y}^T C\mathbf{y} = 0 \quad \text{and} \quad \mathbf{y}^T C\mathbf{x} = 0. \tag{4.29}$$

Consider a linear combination of $\mathbf{x}$ and $\mathbf{y}$. It is clear to see that it lies both on the conic

$$(\mathbf{x} + \lambda\mathbf{y})^T C(\mathbf{x} + \lambda\mathbf{y}) = \mathbf{x}^T C\mathbf{x} + 2\lambda\mathbf{y}^T C\mathbf{x} + \lambda^2\mathbf{y}^T C\mathbf{y} = 0, \tag{4.30}$$

and the line

$$(\mathbf{x} + \lambda\mathbf{y})^T C\mathbf{x} = 0. \tag{4.31}$$

But this holds for any λ and thus we have a line segment which lies on the conic which is a contradiction since we assumed that it was non-degenerate.

This was assuming that $\mathbf{x}$ was lying on the conic but we can also consider the line $\mathbf{l} = C\mathbf{x}$ when $\mathbf{x}$ does not belong to C. In general the line $\mathbf{l} = C\mathbf{x}$ is called the *polar line* of $\mathbf{x}$ with respect to C. The polar line has the property that it intersects the conic at two points and the tangent lines at those points meet in $\mathbf{x}$. This is illustrated in Figure 4.4.

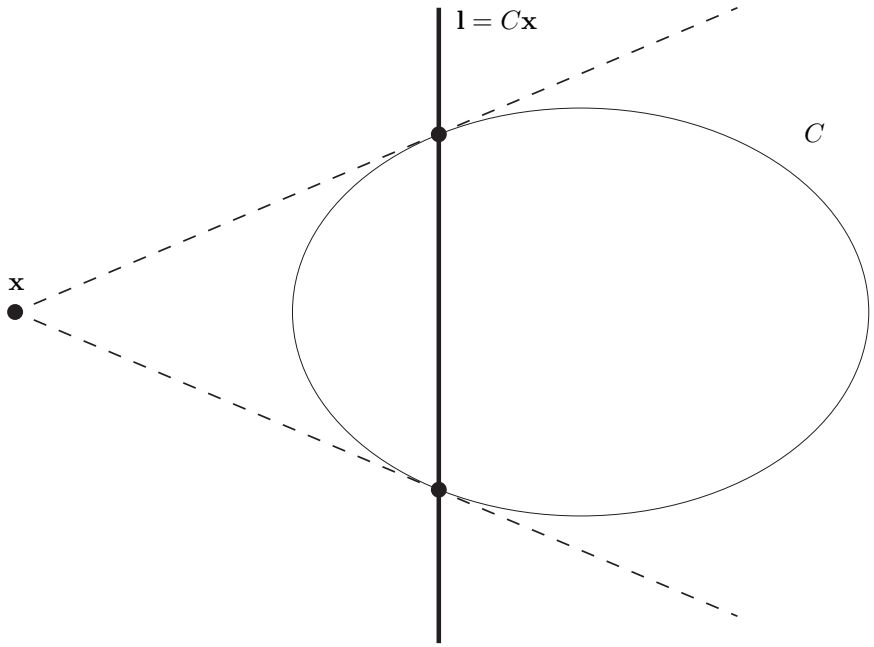


Figure 4.4: The polar line of $\mathbf{x}$ with respect to the conic C.

To see this let $\mathbf{y}$ be one of the intersections of $\mathbf{l} = C\mathbf{x}$ and C. Thus

$$\mathbf{y}^T C\mathbf{y} = 0 \quad \text{and} \quad \mathbf{y}^T\mathbf{l} = \mathbf{y}^T C\mathbf{x} = 0. \tag{4.32}$$

Consider $\mathbf{y}^T C\mathbf{x} = (C\mathbf{y})^T\mathbf{x} = 0$. Thus we have that $\mathbf{x}$ lies on the line $\mathbf{l} = C\mathbf{y}$ which we now know is the tangent to C at $\mathbf{y}$.

4.2.4 Quadrics

The generalization of this conics to $\mathbb{P}^n$ for arbitrary n is called *quadrics*. Similarly as for $n = 2$ we have that the points in $\mathbb{P}^n$ belong to the quadric if they satisfy

$$\mathbf{x}^T Q\mathbf{x} = 0 \tag{4.33}$$

where Q is a $(n+1) \times (n+1)$ symmetric matrix.

Similarly as for conics we can define dual quadrics. Dual quadrics specify which $(n-1)$-dimensional hyperplanes lie tangent to the quadric.

4.3 Geometry at infinity

The ideal points in $\mathbb{P}^2$ is a one parameter family and can be thought of as a line. This line is often called the *line at infinity* and is denoted $\mathbf{l}_\infty$. Similarly for $\mathbb{P}^3$ the ideal points are a two parameter family and form the *plane at infinity* denoted $\boldsymbol{\pi}_\infty$. Since the ideal points are the points where the last coordinate is zero we have that $\mathbf{l}_\infty = (0,0,1)^T$ and $\boldsymbol{\pi}_\infty = (0,0,0,1)^T$.

4.3.1 The absolute conic Ω_∞

On the plane $\boldsymbol{\pi}_\infty$ lies a particularly important conic, the absolute conic Ω_∞, defined by the equations

$$\mathbf{X} = \begin{pmatrix} \mathbf{d} \\ w \end{pmatrix} \in \Omega_\infty \quad \Leftrightarrow \quad \begin{cases} \mathbf{d}^T\mathbf{d} = 0 \\ w = 0 \end{cases} \tag{4.34}$$

One of the properties of the absolute conic is that is allows us to compute angles between vectors in a projective frame. Let $\mathbf{p}$ and $\mathbf{q} \in \mathbb{R}^3$ be two direction vectors in the true metric scene. Since the vectors represent directions they correspond to points on $\boldsymbol{\pi}_\infty$, i.e. $(\mathbf{p}^T,\ 0)^T$ and $(\mathbf{q}^T,\ 0)^T$. From basic linear algebra we know that the angle between two vectors in $\mathbb{R}^3$ is given by

$$\cos\theta = \frac{\mathbf{p}^T\mathbf{q}}{\sqrt{\mathbf{p}^T\mathbf{p}}\sqrt{\mathbf{q}^T\mathbf{q}}}. \tag{4.35}$$

In the metric frame the absolute conic has the form $\Omega_\infty = I_{3\times 3}$ for points on $\boldsymbol{\pi}_\infty$. Thus (4.35) can be written as

$$\cos\theta = \frac{\mathbf{p}^T I\mathbf{q}}{\sqrt{\mathbf{p}^T I\mathbf{p}}\sqrt{\mathbf{q}^T I\mathbf{q}}} = \frac{\mathbf{p}^T\Omega_\infty\mathbf{q}}{\sqrt{\mathbf{p}^T\Omega_\infty\mathbf{p}}\sqrt{\mathbf{q}^T\Omega_\infty\mathbf{q}}}. \tag{4.36}$$

This expression is invariant to projective transforms due to the way conics transform.

$$\mathbf{x} \mapsto H\mathbf{x} \quad \Rightarrow \quad C \mapsto H^{-T}CH^{-1}. \tag{4.37}$$

Thus if we know Ω_∞ and the support plane $\boldsymbol{\pi}_\infty$ in the projective frame we can compute the real angles between two vectors.

The absolute conic is invariant to similarity transforms. To see this consider a general projective transformation

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & 1 \end{bmatrix}. \tag{4.38}$$

Since the plane at infinity $\boldsymbol{\pi}_\infty$ must be preserved (Ω_∞ lies on it) we have that the transformation must be affine, i.e. $\mathbf{v} = 0$. Now consider $\mathbf{X} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} \in \Omega_\infty$. Then

$$0 = \mathbf{d}^T\mathbf{d} = \mathbf{d}^T I\mathbf{d}. \tag{4.39}$$

Thus for points on $\boldsymbol{\pi}_\infty$ we have that the top three coordinates lie on the conic I. For this to be preserved we have

$$I = A^{-T} I A^{-1} \quad \Rightarrow \quad A^T A = I \quad \Rightarrow \quad A \text{ orthogonal.} \tag{4.40}$$

Thus H must be a similarity transform.

4.3.2 Image of the absolute conic ω

The absolute conic is of interest due to its close relationship with the calibration matrices K. Consider a camera $P = K[R\ \mathbf{t}]$ and the projection of a point $\mathbf{X} \in \Omega_\infty$.

$$\mathbf{x} = K[R\,\mathbf{t}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = KR\mathbf{d} \quad \Leftrightarrow \quad K^{-1}\mathbf{x} = R\mathbf{d} \quad \Leftrightarrow \quad \mathbf{x}^T (KK^T)^{-1} \mathbf{x} = \mathbf{d}^T \mathbf{d} = 0. \tag{4.41}$$

The projections lie on the conic $\omega = (KK^T)^{-1}$. In the case of partially calibrated cameras that we are considering, the image of the absolute conic reduces to

$$\omega = \text{diag}(1, 1, f^2). \tag{4.42}$$

Note that this conic only contains complex points.

4.3.3 The absolute quadric Q^*_∞

One of the disadvantages of working with the absolute conic Ω_∞ is that it can't be represented using a single matrix.

Since Ω_∞ is a point conic which lies on the plane at infinity its dual will describe the lines on the plane at infinity which tangent it. The set of planes that intersects the plane at infinity in these lines is captured by a plane quadric called the (dual) *absolute quadric* Q^*_∞. This is illustrated in Figure 4.5

In the metric frame the absolute quadric can be represented by the 4×4 matrix

$$Q^*_\infty = \begin{bmatrix} I_{3\times 3} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}. \tag{4.43}$$

In this settings it is clear that Q^*_∞ is a semidefinite rank 3 matrix and that the plane at infinity $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)^T$ lies in its null space.

Now let H be an arbitrary projective transformation. By considering how points and planes transform we easily deduce how the absolute quadric must transform.

$$\mathbf{X} \mapsto H\mathbf{X} \quad \Rightarrow \quad \boldsymbol{\pi} \mapsto H^{-T}\boldsymbol{\pi} \quad \Rightarrow \quad Q^*_\infty \mapsto H Q^*_\infty H^T. \tag{4.44}$$

Using this it is easy to see that the transformed $\boldsymbol{\pi}_\infty$ still lies in the null space

$$Q^*_\infty \boldsymbol{\pi}_\infty = 0 \quad \Rightarrow \quad (HQ^*_\infty H^T)(H^{-T}\boldsymbol{\pi}_\infty) = H(Q^*_\infty \boldsymbol{\pi}_\infty) = 0. \tag{4.45}$$

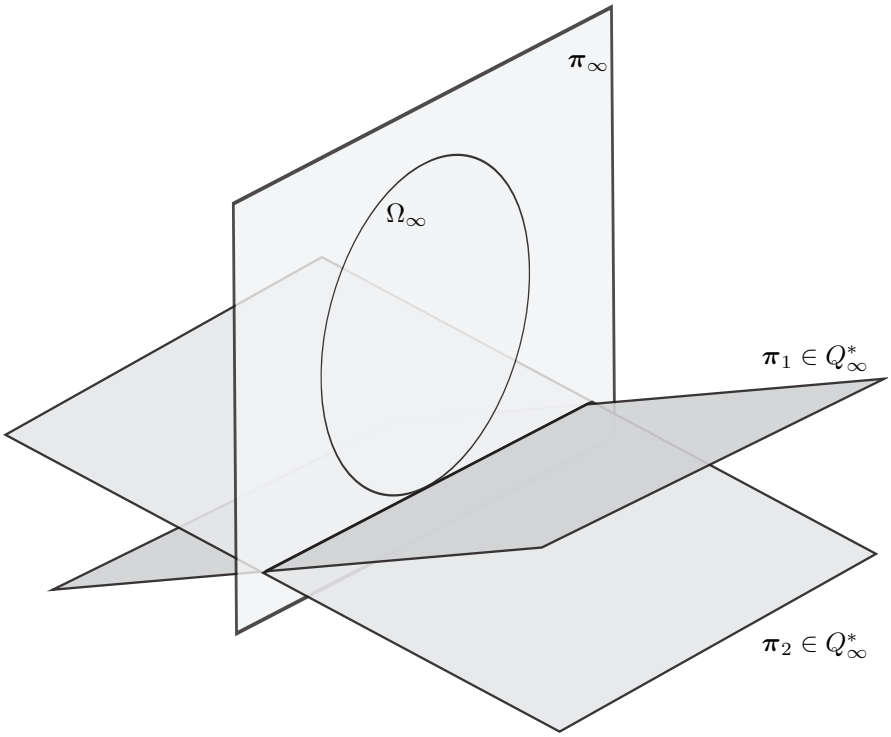


Figure 4.5: The absolute conic Ω_∞ which lies on the plane at infinity $\boldsymbol{\pi}_\infty$. The absolute quadric Q_∞^* describes those planes which tangent Ω_∞ when they intersect $\boldsymbol{\pi}_\infty$.

Just like the absolute conic the absolute quadric is invariant to similarity transforms. This is easily seen from

$$HQ_\infty^* H^T = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & 1 \end{bmatrix} \begin{bmatrix} I_{3\times 3} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} A^T & \mathbf{v} \\ \mathbf{t}^T & 1 \end{bmatrix} = \begin{bmatrix} AA^T & A\mathbf{v} \\ \mathbf{v}^T A^T & \mathbf{v}^T\mathbf{v} \end{bmatrix} = Q_\infty^*. \quad (4.46)$$

It follows that $\mathbf{v} = 0$ and $AA^T = I$. Thus H must be a similarity transform.

The projection of the absolute quadric Q_∞^* is the dual to the image of the absolute conic ω^*. By the projection of the absolute quadric Q_∞^* in a camera P, we mean the intersection of the image plane and the planes which both lie on the absolute quadric and contain the camera center of P. This is illustrated in Figure 4.6.

The projection in a camera P can be written

$$\omega^* = PQ_\infty^* P^T. \quad (4.47)$$

To see this consider a line $\mathbf{l}$ which lies on ω^*, i.e. $\mathbf{l}^T\omega^*\mathbf{l} = 0$. This line will backproject onto the plane $\boldsymbol{\pi} = P^T\mathbf{l}$ which must lie on the absolute quadric.

$$0 = \boldsymbol{\pi}^T Q_\infty^* \boldsymbol{\pi} = \mathbf{l}^T PQ_\infty^* P^T\mathbf{l} = \mathbf{l}^T\omega^*\mathbf{l}. \quad (4.48)$$

Since this will hold for any $\mathbf{l} \in \omega^*$ we have that $\omega^* = PQ_\infty^* P^T$.

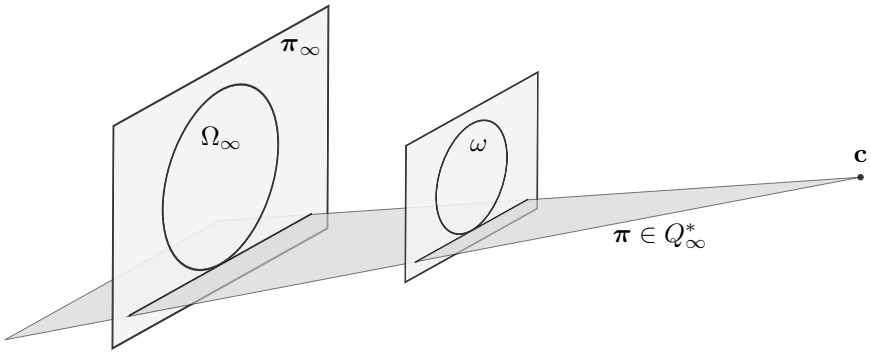


Figure 4.6: The absolute conic Ω_∞ projects down to the image of the absolute conic ω. The projection of the absolute quadric Q^*_∞ is the dual of the image of the absolute conic ω^*, which is a line conic. The projection is formed by intersecting those planes which belong to the quadric and contain the camera center with the image plane.

Chapter 5

Estimation methods

In the previous chapter we introduced a number of objects that can be used to determine the geometry of a scene. These objects will be used as building blocks for the full metric reconstruction. In this chapter we will present algorithms for accurately estimating these objects from given point correspondences in the presence of noise and outliers.

5.1 Homogeneous least squares

For many of the homogeneous entities (e.g. $F, \mathcal{T}$) used in reconstruction there exist some linear constraint on their elements. Rearranging the elements into a vector we can model them with a system of the form

$$A\mathbf{w} = \mathbf{0}. \tag{5.1}$$

To avoid the trivial solution $\mathbf{w} = \mathbf{0}$ we add the constraint $||\mathbf{w}|| = 1$. This can be done w.l.o.g. due to the scale invariance of $\mathbf{w}$.

If A is constructed using image data it is unlikely that there exist $\mathbf{w}$ such that the equations are satisfied exactly. Instead we desire a least squares solution, i.e. we consider the following optimization problem

$$\min_{\mathbf{w}} ||A\mathbf{w}||^2 \quad \text{s.t.} \quad ||\mathbf{w}|| = 1. \tag{5.2}$$

This problem can be solved by finding the *singular value decomposition* (SVD) of A. The SVD of A is given by

$$A = USV^T \quad \text{where} \quad UU^T = I, \quad VV^T = I, \quad S = \text{diag}(\sigma_1, ..., \sigma_n), \tag{5.3}$$

where $\sigma_1 \geq ... \geq \sigma_n \geq 0$ are the singular values of A.

The objective function can then be written

$$||A\mathbf{w}||^2 = ||USV^T\mathbf{w}||^2 = ||S\underbrace{V^T\mathbf{w}}_{\hat{\mathbf{w}}}||^2 = ||S\hat{\mathbf{w}}||^2, \tag{5.4}$$

where the second equality follows from the fact that orthogonal matrices preserve length. The constraint simply becomes $||\mathbf{w}|| = ||V\hat{\mathbf{w}}|| = ||\hat{\mathbf{w}}|| = 1$. Since the singular values are sorted in descending order, (5.4) is minimized by letting $\hat{\mathbf{w}} = (0, ..., 0, 1)^T$ which then corresponds to $\mathbf{w}$ is being the last column of V.

So to solve the homogeneous least squares problem we simply find the SVD of the measurement matrix and take the rightmost column of V.

5.1.1 Equality constraints

Sometimes we have additional constraints on $\mathbf{w}$, e.g. $C\mathbf{w} = \mathbf{0}$ for some matrix C. Thus we consider the problem

$$\min_{\mathbf{w}} ||A\mathbf{w}||^2 \quad \text{s.t.} \quad ||\mathbf{w}|| = 1, \quad C\mathbf{w} = \mathbf{0}. \tag{5.5}$$

This problem can be solved by finding a basis for the null space of C. We assume that C has a nontrivial null space since otherwise the problem lacks a solution. By computing the SVD of C, the basis can be formed as the columns of V which correspond to the singular values which are zero.

Let $C^\perp$ be the matrix with the basis vectors as columns. Then any $\mathbf{w}$ that can be formed as $\mathbf{w} = C^\perp\hat{\mathbf{w}}$ for some $\hat{\mathbf{w}}$ will satisfy the constraint $C\mathbf{w} = \mathbf{0}$ since $CC^\perp = 0$.

We then solve the problem

$$\min_{\hat{\mathbf{w}}} ||AC^\perp\hat{\mathbf{w}}||^2 \quad \text{s.t.} \quad ||\hat{\mathbf{w}}|| = 1 \tag{5.6}$$

using the original algorithm. Note that $||\mathbf{w}|| = ||C^\perp\hat{\mathbf{w}}|| = ||\hat{\mathbf{w}}||$ since the columns of $C^\perp$ are orthogonal unit vectors. The solution to the original constrained problem (5.5) is then given by $\mathbf{w} = C^\perp\hat{\mathbf{w}}$.

5.1.2 Subspace constraints

A very similar problem is

$$\min_{\hat{\mathbf{w}}} ||A\mathbf{w}||^2 \quad \text{s.t.} \quad ||\mathbf{w}|| = 1, \quad \mathbf{w} = E\hat{\mathbf{w}}, \tag{5.7}$$

which is another way of saying that $\mathbf{w}$ should lie in the column space of E. We assume that E does not have full rank otherwise the problem is equivalent to the original problem (5.2).

Let $r = \text{rank}(E)$ and compute the SVD of $E = USV^T$. Construct U_r as the first r columns of U. Then U_r spans the column space of E. Similarly to the previous section the problem can be solved by finding

$$\min_{\hat{\mathbf{w}}} ||AU_r\hat{\mathbf{w}}||^2 \quad \text{s.t.} \quad ||\hat{\mathbf{w}}|| = 1, \tag{5.8}$$

using the original algorithm and then setting $\mathbf{w} = U_r\hat{\mathbf{w}}$.

5.1.3 Sampson approximation

Often the measurement matrix A is formed using image points and it is of interest how well a given solution $\mathbf{w}$ is described by these points.

In the homogeneous least squares method we minimized the algebraic error $\boldsymbol{\epsilon} = A\mathbf{w}$ but a better metric would be the geometric error, i.e. the minimum distance that the image points have to be moved for the equation $A\mathbf{w} = \mathbf{0}$ to hold exactly.

Let $\mathbf{x}$ be a vector containing the image points and $C(\mathbf{x})$ be the algebraic error corresponding to the measurement matrix constructed using these points, i.e. $C(\mathbf{x}) = A\mathbf{w}$ where A depends on $\mathbf{x}$. We are interested in the problem of finding the $\boldsymbol{\delta}$ of minimum length such that $C(\mathbf{x} + \boldsymbol{\delta}) = \mathbf{0}$.

By Taylor expansion around $\mathbf{x}$ we get

$$C(\mathbf{x} + \boldsymbol{\delta}) = C(\mathbf{x}) + J\boldsymbol{\delta} = \mathbf{0}, \tag{5.9}$$

where J is the Jacobian. To find an approximation of the geometric error we are interested in solving

$$\min_{\boldsymbol{\delta}} ||\boldsymbol{\delta}||^2 \quad \text{s.t.} \quad C(\mathbf{x}) + J\boldsymbol{\delta} = \mathbf{0}. \tag{5.10}$$

We form the Lagrangian function $L(\boldsymbol{\delta}, \boldsymbol{\lambda}) = \boldsymbol{\delta}^T\boldsymbol{\delta} + 2\boldsymbol{\lambda}^T(C(\mathbf{x}) + J\boldsymbol{\delta})$.

Differentiating we get

$$L_{\boldsymbol{\delta}} = 2\boldsymbol{\delta}^T + 2\boldsymbol{\lambda}^T J = \mathbf{0}, \tag{5.11}$$

$$L_{\boldsymbol{\lambda}} = C(\mathbf{x}) + J\boldsymbol{\delta} = \mathbf{0}. \tag{5.12}$$

By substituting the first equation into the second we get

$$C(\mathbf{x}) = JJ^T\boldsymbol{\lambda} \quad \Leftrightarrow \quad \boldsymbol{\lambda} = (JJ^T)^{-1}C(\mathbf{x}). \tag{5.13}$$

Finally we can solve for $\boldsymbol{\delta}$ and get $\boldsymbol{\delta} = J^T(JJ^T)^{-1}C(\mathbf{x})$ which gives us

$$||\boldsymbol{\delta}||^2 = C(\mathbf{x})^T(JJ^T)^{-1}C(\mathbf{x}), \tag{5.14}$$

which is a first order approximation of the geometric error.

5.2 Fundamental matrix estimation

We will now demonstrate how to use the concepts introduced in the previous section to estimate the fundamental matrix defined in Section 4.1.1.

The 8 point algorithm was first introduced by Longuet-Higgins in [18] who used it for estimation relative position and orientation for calibrated cameras. The algebraic minimization algorithm was presented by Hartley in [14]. For a review of some of the methods for fundamental matrix estimation the reader is referred to a paper by Zhang [26] or the excellent book by Hartley and Zisserman [11].

5.2.1 Normalized 8 point algorithm

The fundamental matrix F puts constraints on corresponding points in two views. If $\mathbf{x}$ and $\bar{\mathbf{x}}$ are projections of the same 3D point it holds that

$$\bar{\mathbf{x}}^T F \mathbf{x} = 0. \tag{5.15}$$

Given corresponding points $\mathbf{x} \leftrightarrow \bar{\mathbf{x}}$ this forms a linear equation in the elements of F. Since we have 9 unknowns (elements of F) but are only interested in a solution up to scale we need at least 8 point correspondences.

By arranging the elements of F into a vector $\mathbf{f}$ we can express the equations using a matrix

$$A\mathbf{f} = \mathbf{0}, \tag{5.16}$$

where the measurement matrix A is formed using the point correspondences.

Since the equations are unlikely to be satisfied exactly due to noise we instead consider the homogeneous least squares problem

$$\min_{\mathbf{f}} ||A\mathbf{f}||^2 \quad \text{s.t.} \quad ||\mathbf{f}|| = 1, \tag{5.17}$$

where the constraint makes sure that we throw away the trivial solution $\mathbf{f} = \mathbf{0}$. This optimization problem can be solved using the SVD based algorithm in Section 5.1.

In [13] Hartley shows the importance of normalizing the image points before estimating F. Hartley suggest simply scaling and translating the points such that they are centered on the origin with a mean distance of $\sqrt{2}$.

5.2.2 Algebraic minimization algorithm

The main problem with the normalized 8 point algorithm is that it doesn't enforce the rank deficiency of F.

From previous chapters we know that for a pair of cameras it holds that

$$P_1 = [I \ \mathbf{0}], \ P_2 = [A \ \mathbf{t}] \quad \Rightarrow \quad F = [\mathbf{t}]_\times A. \tag{5.18}$$

For fix $\mathbf{t}$ the equation $F = [\mathbf{t}]_\times A$ is linear in the elements of F and A. Let $\mathbf{f}$ and $\mathbf{a}$ be vectors that contain the elements of the two matrices. Then the equation can be written $\mathbf{f} = E_\mathbf{t}\mathbf{a}$ where $E_\mathbf{t}$ is the matrix formed from $\mathbf{t}$.

So assume that we know $\mathbf{t}$. Then to find F we could solve the problem

$$\min_{\mathbf{a}} ||A\mathbf{f}||^2 \quad \text{s.t.} \quad ||\mathbf{f}|| = 1, \quad \mathbf{f} = E_\mathbf{t}\mathbf{a}. \tag{5.19}$$

Which is a constrained homogeneous least squares problem which we can be solved using the method presented in Section 5.1.2. The resulting F will have the required rank deficiency since $[\mathbf{t}]_\times$ has rank 2.

In general $\mathbf{t}$ is unknown and we instead consider the optimization problem

$$\min_{\mathbf{t}} \left\{ \min_{\mathbf{a}} ||A\mathbf{f}||^2 \quad \text{s.t.} \quad ||\mathbf{f}|| = 1, \quad \mathbf{f} = E_\mathbf{t}\mathbf{a} \right\} \tag{5.20}$$

where the outer minimization is performed using numerical differentiation and is initialized using the normalized 8 point method.

The size of measurement matrix A is $n \times 9$ where n is the number of point correspondences. The computational complexity of the problem can be reduced using the SVD of A since

$$A = USV^T \Rightarrow ||A\mathbf{f}|| = ||USV^T\mathbf{f}|| = ||SV^T\mathbf{f}||, \tag{5.21}$$

and only the top 9 rows of S are non-zero. Thus we can replace the measurement matrix by the top 9 rows of SV^T.

5.2.3 Comparison

To evaluate the performance of these algorithms a small experiment was performed. The test environment consisted of two synthethic cameras viewing a scene consisting of 25 3D points. The projections were disturbed by gaussian noise with varying variance. The fundamental matrix was estimated from the noisy points and the mean geometric error was calculated using the Sampson approximation.

The three methods considered were

1. Normalized 8p algorithm.
2. Algebraic minimization algorithm.
3. Algebraic minimization algorithm followed by bundle adjustment. (See Section 5.6.)

In Figure 5.1 we can see the resultings errors. The test was repeated 100 times for each standard deviation and the error was then averaged.

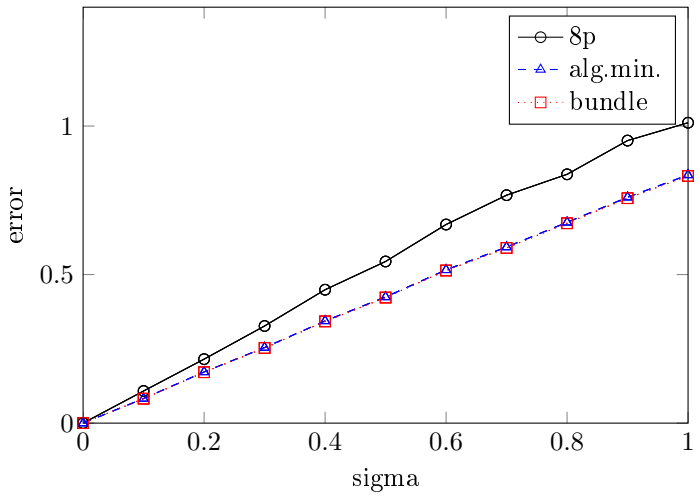


Figure 5.1

We see that the algebraic minimization method has better performance than the 8 point algorithm and that the bundle adjustment performed after the algebraic

minimization only gives a very small reduction in error. The computational cost of the bundle adjustment is a lot higher than for the algebraic minimization method. In Figure 5.2 we can see the computation time needed for the two methods for a varying number of points.

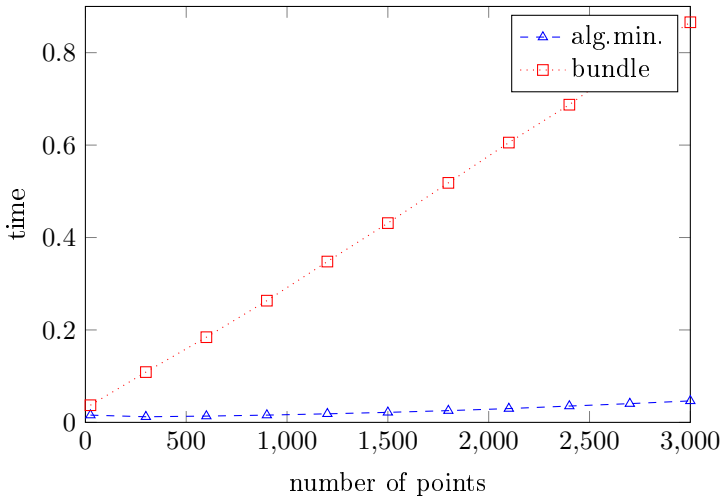


Figure 5.2: Comparison of computation time for estimating the fundamental matrix using algebraic minimization and bundle adjustment.

5.2.4 Minimal solver - 7 point algorithm

In robust estimation methods like RANSAC (see Section 5.5) it is of interest to find estimates using as little data as possible. This gives a need for so called *minimal solvers* which uses the minimum number of point correspondences possible, for the fundamental matrix this is 7 points.

If we construct the measurement matrix A using only 7 points (i.e. A is a 7×9 matrix) it will have a 2 dimensional null space. Let $\mathbf{f}_1$ and $\mathbf{f}_2$ be orthogonal unit vectors which span the null space. Then our sought F can be written $F = \alpha F_1 + (1 - \alpha)F_2$ for some unknown α.

One way to enforce the rank deficiency of F is to consider the constraint

$$\det(F) = 0 \quad \Leftrightarrow \quad \det(\alpha F_1 + (1 - \alpha)F_2) = 0. \tag{5.22}$$

For fix F_1 and F_2 this is a third degree polynomial equation in α. Solving this equation we get either one or three real solutions. If there are multiple solutions we can select one by testing which one best corresponds with the image points.

5.3 Trifocal tensor estimation

The estimation methods for the trifocal tensor that we consider will now be derived in the same manner as for the fundamental matrix.

The linear algorithm for trifocal tensor estimation was first used by Hartley in [8]. The algebraic minimization method is also due Hartley [14]. The minimal solver for the trifocal tensor from point correspondences was presented by Zisserman and Torr in [27].

5.3.1 Normalized linear algorithm

For the trifocal tensor's point-point-point correspondence it holds that

$$[\mathbf{x}_2]_\times(\sum_i x_1^i T_i)[\mathbf{x}_3]_\times = 0_{3\times 3}. \tag{5.23}$$

Given a point correspondence between three images this gives us nine linear equations for the elements of $\mathcal{T}$. Unfortunately only four are linearly independent. One choice of four linearly independent equations are

$$\sum_k x_1^k(x_2^i x_3^l T_k^{33} - x_3^l T_k^{i3} - x_1^i T_k^{3l} + T_k^{il}) = 0, \tag{5.24}$$

where $i, l \in \{1, 2\}$ and T_k^{il} denotes the element (i, l) of the matrix T_k. Since $\mathcal{T}$ has 27 elements we will require at least 7 point correspondences.

From this we can then construct the measurement matrix A such that $A\mathbf{t} = \mathbf{0}$ where $\mathbf{t}$ is the vector containing the 27 elements of $\mathcal{T}$. Since the system is over-determined we consider the homogeneous least squares problem

$$\min_{\mathbf{t}} ||A\mathbf{t}||^2 \quad \text{s.t.} \quad ||\mathbf{t}|| = 1, \tag{5.25}$$

which can be solved using the SVD based method presented in Section 5.1.

As with F estimation it is necessary to perform some normalization on the image points.

5.3.2 Algebraic minimization algorithm

The drawback of this approach is that the solution in general will not be a valid trifocal tensor since each T_i is not required to have rank 2.

If $P_1 = [I\ \mathbf{0}]$, $P_2 = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3\ \mathbf{a}_4]$ and $P_3 = [\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3\ \mathbf{b}_4]$ then the associated trifocal tensor is given by

$$T_i = \mathbf{a}_i\mathbf{b}_4^T - \mathbf{a}_4\mathbf{b}_i^T, \tag{5.26}$$

which is linear in $\mathbf{a}_i$ and $\mathbf{b}_i$ for $i = 1, 2, 3$.

So for fixed $\mathbf{a}_4$ and $\mathbf{b}_4$ the trifocal tensor can be parametrized with the 18 elements of the first three columns in the two cameras. This can be written $\mathbf{t} = E\hat{\mathbf{t}}$ where E is the 27×18 matrix which captures equation (5.26).

To find a trifocal tensor which satisfies the rank constraint we consider the problem

$$\min_{\mathbf{a}_4,\mathbf{b}_4} \left\{ \min_{\hat{\mathbf{t}}} ||A\mathbf{t}||^2 \quad \text{s.t.} \quad ||\mathbf{t}|| = 1, \quad \mathbf{t} = E\hat{\mathbf{t}} \right\}, \tag{5.27}$$

where the inner minimization is a constrained homogeneous least squares problem and the outer minimization is performed using numerical differentiation and is initialized using the linear approach.

5.3.3 Comparison

To evaluate the performance of these algorithms a small experiment was performed. The test environment consisted of three synthethic cameras viewing a scene consisting of 25 3D points. The projections were disturbed by gaussian noise with varying variance. The trifocal tensor was estimated from the noisy points and the mean geometric error was calculated using the Sampson approximation.

The three methods considered were

1. Normalized linear algorithm
2. Algebraic minimization algorithm
3. Algebraic minimization algorithm followed by bundle adjustment. (See Section 5.6)

In Figure 5.3 we can see the resulting errors. The test was repeated 100 times for each standard deviation and the error was then averaged.

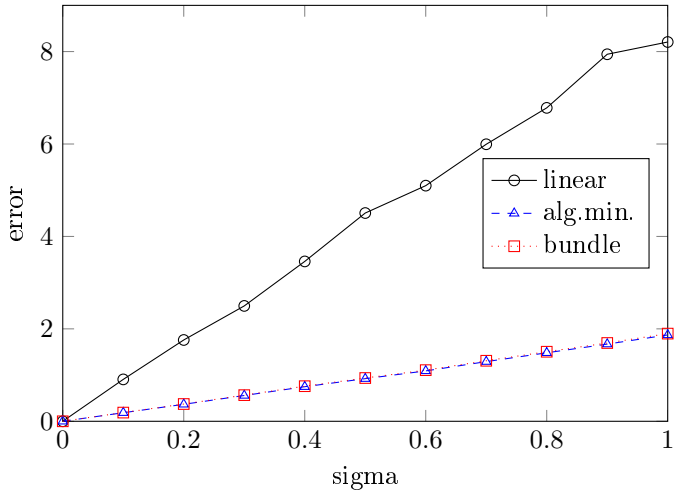


Figure 5.3

The performance is similar to the fundamental matrix. The algebraic minimization method has better performance than the linear algorithm and that bundle adjustment performed after the algebraic minimization only gives a very small reduction in error. In Figure 5.4 we see a comparison of the computation time

for the two methods for a varying number of points. Once again we see that the algebraic minimization method performs better than bundle adjustment when the number of points grow.

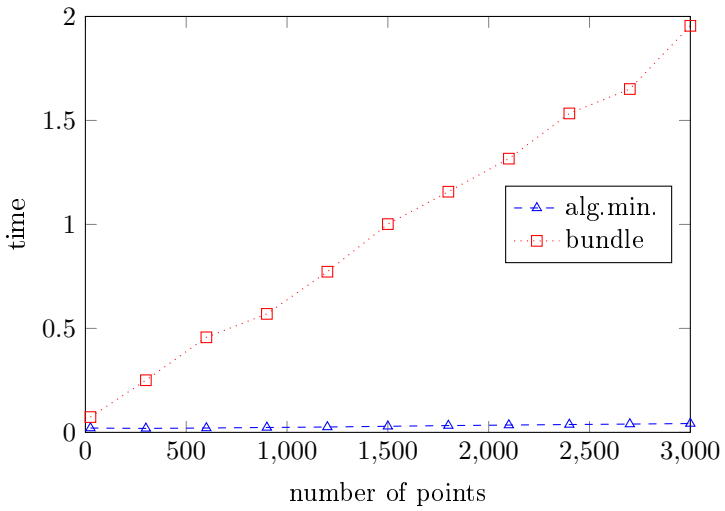


Figure 5.4

5.3.4 Minimal solver - 6 point algorithm

There exists an algorithm for solving for the trifocal tensor from only 6 point correspondences. The algorithm uses Carlsson-Weinshall duality to construct a projective reconstruction from the 6 points and then constructs the trifocal tensor from the three cameras.

5.4 Auto-calibration

The previous sections described methods for building projective reconstructions from image pairs and triplets. To be able to upgrade the projective reconstructions into metric reconstructions we will have to estimate the calibration of the cameras. The process of finding the calibration from the images is called *auto-calibration*.

5.4.1 Hartley's method for pairwise auto-calibration

In [12] Hartley propose a method for finding the focal lengths of a pair of partially calibrated cameras where the only unknown intrinsic parameters are the focal lengths. The calibration matrix K and the image of the absolute conic

ω will then be

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \omega = (KK^T)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f^2 \end{bmatrix}. \tag{5.28}$$

We can w.l.o.g. assume that the epipoles lie on one of the axis in the images, i.e. $\mathbf{e} = \begin{pmatrix} e_1, & 0, & e_3 \end{pmatrix}^T$. If this is not the case the cameras can be rotated around their principal axes. Due to the special structure on K this rotation preserves the focal lengths.

It can be shown that if $\mathbf{e} = \begin{pmatrix} e_1, & 0, & e_3 \end{pmatrix}^T$ and $\bar{\mathbf{e}} = \begin{pmatrix} \bar{e}_1, & 0, & \bar{e}_3 \end{pmatrix}^T$ are the two epipoles, the fundamental matrix F can be decomposed as

$$F = \begin{bmatrix} \bar{e}_3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\bar{e}_1 \end{bmatrix} \begin{bmatrix} a & b & a \\ c & d & c \\ a & b & a \end{bmatrix} \begin{bmatrix} e_3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -e_1 \end{bmatrix}, \tag{5.29}$$

since $F\mathbf{e} = 0$ and $\bar{\mathbf{e}}^T F = 0$.

The polar line w.r.t. ω of the epipole in the left image is given by $\omega\mathbf{e} = \begin{pmatrix} e_1, & 0, & f^2 e_3 \end{pmatrix}^T$ (see Section 4.2.3). The intersection $\mathbf{x} = (u, v, 1)^T$ of the polar line and the image of the absolute conic ω is given by

$$\begin{cases} (\omega\mathbf{e})^T\mathbf{x} = 0 \\ \mathbf{x}^T\omega\mathbf{x} = 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} e_1 u + f^2 e_3 = 0 \\ u^2 + v^2 + f^2 = 0 \end{cases} \tag{5.30}$$

which gives us the intersection points $\mathbf{x} = \begin{pmatrix} -f^2 e_3, & if(e_1^2 + f^2 e_3^2)^{1/2}, & e_1 \end{pmatrix}^T$ and its complex conjugate. The geometry of the situation is illustrated in Figure 5.5.

Let $\Delta = (e_1^2 + f^2 e_3^2)^{1/2}$ and consider the epipolar line corresponding to $\mathbf{x}$.

$$F\mathbf{x} = F \begin{pmatrix} -f^2 e_3 \\ if\Delta \\ e_1 \end{pmatrix} = \begin{pmatrix} -\bar{e}_3 \\ \frac{-ac\Delta^2 - bdf^2 + if(ad-bc)\Delta}{a^2\Delta^2 + b^2 f^2} \\ \bar{e}_1 \end{pmatrix} \triangleq \begin{pmatrix} -\bar{e}_3 \\ \mu + i\nu \\ \bar{e}_1 \end{pmatrix}. \tag{5.31}$$

The epipolar line corresponding to the other intersection point is given by

$$F\mathbf{x}^* = (F\mathbf{x})^* = \begin{pmatrix} -\bar{e}_3, & \mu - i\nu, & \bar{e}_1 \end{pmatrix}^T. \tag{5.32}$$

These two lines pass through the epipole $\bar{\mathbf{e}}$ and lie tangent to ω in the second image. Since ω is a circle of complex points centered on the origin and the epipole $\bar{\mathbf{e}}$ lies on the x-axis in the image we have that the lines must be symmetric around the x-axis. This is illustrated in Figure 5.6. This implies that $(F\mathbf{x})_2 = -(F\mathbf{x}^*)_2$. From (5.31) and (5.32) we get that

$$\mu + i\nu = -(\mu - i\nu) \quad \Rightarrow \quad \mu = 0. \tag{5.33}$$

Using this we get

$$\mu = \frac{-ac\Delta^2 - bdf^2}{a^2\Delta^2 + b^2 f^2} = 0 \quad \Rightarrow \quad -ac\Delta^2 - bdf^2 = 0 \quad \Rightarrow \quad f^2 = \frac{-ace_1^2}{ace_3^2 + bd}. \tag{5.34}$$

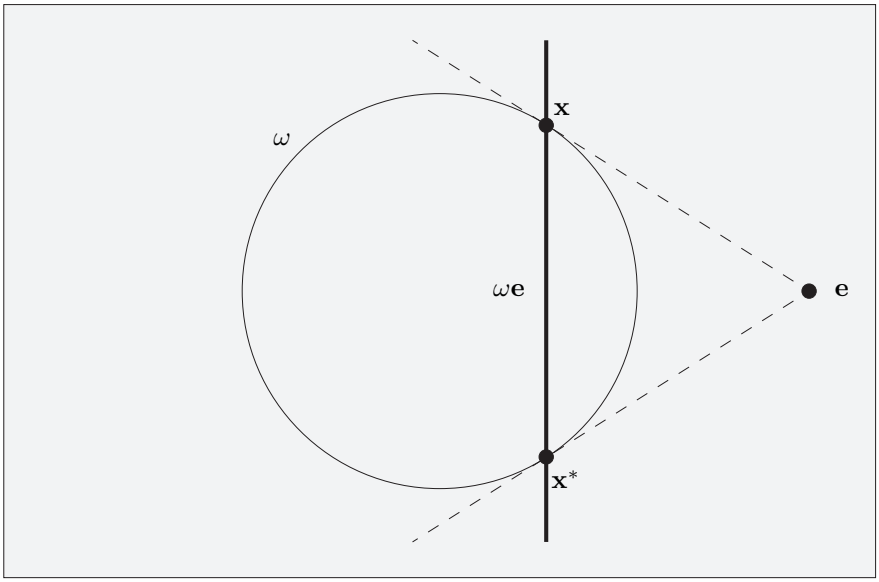


Figure 5.5

By reversing the roles of the images we get that

$$\bar{f}^2 = \frac{-ab\bar{e}_1^2}{ab\bar{e}_3^2 + cd}. \tag{5.35}$$

We note that if $a = 0$ then

$$\mu = \frac{-bdf^2}{b^2 f^2} = \frac{-d}{b} = 0 \quad \Rightarrow \quad d = 0, \tag{5.36}$$

which is true independently of f. In this case the focal length can not be determined from the camera pair. This degenerate configuration appears when the baseline and the two principal axes are coplanar. Then the principal rays will intersect in a point which will be projected onto the origin on both images. Thus it holds that

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} F \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad F_{33} = 0 \quad \Rightarrow \quad a = 0. \tag{5.37}$$

There is another degenerate case that occurs when $b = 0$. This case is less important in practice though since it occurs when one principal axis is orthogonal to the plane spanned by the other principal axis and the baseline.

Degenerate configurations

We now consider the degenerate case when the princripal axis and baseline are coplanar. Since the cameras are assumed to be rotated such that the epipoles lie on one of the image axis this configuration corresponds to planar motion.

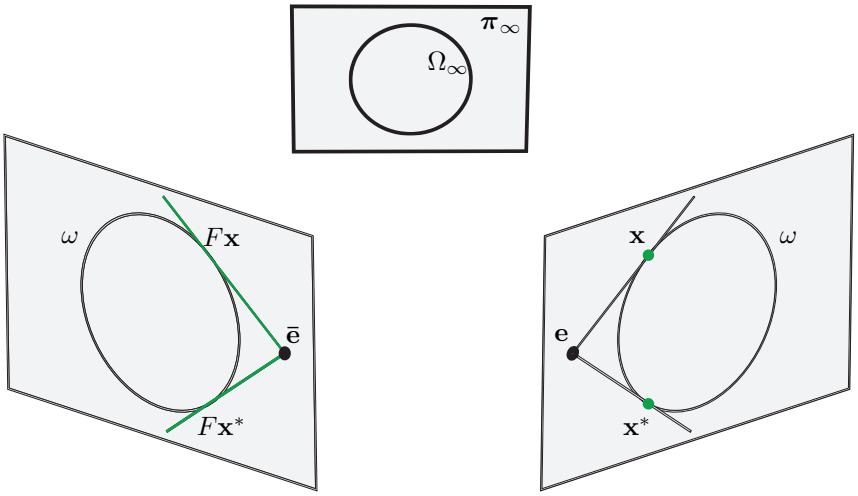


Figure 5.6

For planar motion it is known that the fundamental matrix F will have a special structure. See Section 4.1.1.

$$F = \begin{bmatrix} 0 & a & 0 \\ b & 0 & c \\ 0 & d & 0 \end{bmatrix} = \begin{pmatrix} a \\ 0 \\ d \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} b & 0 & c \end{pmatrix}. \tag{5.38}$$

The essential matrix E is formed by multiplying F by the calibration matrices K and $\bar{K}$.

$$E = \operatorname{diag}(\bar{f}, \bar{f}, 1)\ F\ \operatorname{diag}(f, f, 1) = \begin{pmatrix} f\bar{f}a \\ 0 \\ fd \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} f\bar{f}b & 0 & \bar{f}c \end{pmatrix}. \tag{5.39}$$

It's clear that the singular values of E is given by

$$\sigma_1 = \sqrt{(f\bar{f}a)^2 + (fd)^2}, \quad \sigma_2 = \sqrt{(f\bar{f}b)^2 + (\bar{f}c)^2}, \quad \sigma_3 = 0.$$

Setting the first two singular values equal and squaring gives us

$$G(f, \bar{f}) = d^2 f^2 + (a^2 - b^2) f^2 \bar{f}^2 - c^2 \bar{f}^2 = 0. \tag{5.40}$$

In [12] Hartley gives an equivalent derivation of (5.40) using the image of absolute conic.

5.4.2 Linear method for Q_∞^* estimation

The absolute quadric Q_∞^* is a degenerate plane quadric which is the dual to the absolute conic. Thus the projection of the absolute quadric is the dual to the

projection of the absolute conic. To estimate the absolute quadric we consider the form of its projection

$$PQ_\infty^* P^T = \omega^* = \begin{bmatrix} f^2 & 0 & 0 \\ 0 & f^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.41}$$

This gives us four constraints which are linear in the elements of Q_∞^*

$$\begin{aligned} \omega_{12}^* = \omega_{13}^* = \omega_{23}^* = 0, \\ \omega_{11}^* = \omega_{22}^*. \end{aligned} \tag{5.42}$$

Since Q_∞^* is a symmetric 4×4 matrix we can parametrize it by the 10 elements above the diagonal. Let $\mathbf{q}$ be the vector of those elements and construct a matrix A from the projective cameras which captures the constraints (5.42) for each of the three cameras.

An estimate of the absolute quadric Q_∞^* can then found by solving

$$\min_{\mathbf{q}} ||A\mathbf{q}||^2 \quad \text{s.t.} \quad ||\mathbf{q}|| = 1, \tag{5.43}$$

using the SVD based algorithm presented in Section 5.1.

Similarly to the linear estimation of the trifocal tensor this approach has the drawback that is doesn't consider the rank deficiency of Q_∞^*. A simple method for correcting the rank deficiency is to simple set the smallest singular value to zero.

Iterative improvement

If we fix the plane at infinity $\boldsymbol{\pi}_\infty$ the constraint $Q_\infty^* \boldsymbol{\pi}_\infty = 0$ becomes linear in the elements of $\mathbf{q}$. To estimate the absolute quadric we consider the minimization problem

$$\min_{\boldsymbol{\pi}_\infty} \left\{ \min_{\mathbf{q}} ||A\mathbf{q}||^2 \quad \text{s.t.} \quad ||\mathbf{q}|| = 1 \text{ and } Q_\infty^* \boldsymbol{\pi}_\infty = 0 \right\}, \tag{5.44}$$

where the inner minimization is performed using the algorithm for constrained homogeneous least squares and the outer minimization is performed by numerical differentiation. The initial guess for $\boldsymbol{\pi}_\infty$ is provided by the linear method.

In practice this method has turned out to have slightly better performance than correcting the last singular value. See Section 5.4.2 for a comparison.

Metric upgrade

Let H be the projective transformation which transforms the projective reconstruction to the metric, i.e.

$${}^M P = PH \quad \text{and} \quad {}^M \mathbf{X} = H^{-1}\mathbf{X}. \tag{5.45}$$

Since a plane transforms as $\boldsymbol{\pi} \mapsto H^T\boldsymbol{\pi}$, the absolute quadric (which is a plane quadric) will transform as

$$\boldsymbol{\pi}^T Q_\infty^* \boldsymbol{\pi} = \boldsymbol{\pi}^T H({}^M Q_\infty^*) H^T \boldsymbol{\pi} \quad \Rightarrow \quad Q_\infty^* = H({}^M Q_\infty^*) H^T \tag{5.46}$$

In the metric frame the absolute quadric has the form ${}^M Q_\infty^* = \begin{bmatrix} I_{3\times 3} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$.

To find the metric upgrade H we perform eigenvalue decomposition on the estimated absolute quadric. Since Q_∞^* is symmetric we know that the eigenvectors form an orthogonal basis and the decomposition becomes $Q_\infty^* = VDV^T$ with $VV^T = I$. Let D be ordered so that the zero eigenvalue is at D_{44}. Then

$$Q_\infty^* = VDV^T = \underbrace{V\sqrt{D}}_{H} \begin{bmatrix} I_{3\times 3} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \underbrace{\sqrt{D}V^T}_{H^T}. \tag{5.47}$$

Since we require H to be invertible we replace the zero diagonal element of $\sqrt{D}$ with ± 1. The sign is chosen such that the majority of the 3D-points are infront of the first camera.

Finally the projective cameras are transformed by H and the structure points by H^{-1}. RQ-factorization is performed on the cameras to extract the rotation and calibration matrices. Due to noise the calibration matrices will not be completely diagonal and to correct this we simply discard the non-zero elements outside the diagonal.

Degenerate configurations

In the previous sections we estimated the absolute quadric Q_∞^* by considering the form of its projection $\omega^* = PQ_\infty^* P^T$. This resulted in a set of linear equations on the elements of Q_∞^* which we then solved in a least squares sense.

In [17] Kahl et. al investigates under what conditions there can exist a *false* absolute quadric Q_f^* which also projects down onto a conic of the correct form. For the partially calibrated cameras we are considering this means that

$$PQ_\infty^* P^T = \mathrm{diag}(f^2,\ f^2,\ 1), \quad \text{and} \quad PQ_f^* P^T = \mathrm{diag}(\hat{f}^2,\ \hat{f}^2,\ 1). \tag{5.48}$$

From Section 5.4.2 we know that we can extract a metric upgrade from Q_∞^*. Thus when there can exists false absolute quadrics there can also exist false metric reconstructions which have calibration matrices on the correct form.

In some cases it is possible to identify false absolute quadrics Q_f^* by considering *chirality*. Chirality is the concept that points should lie in front of the cameras.

For cameras with unit aspect ratio, zero skew and known principal point Kahl et. al shows that there are three different cases when auto-calibration becomes degenerate. These cases are:

(a) If there are only two camera centers. For a triplet this means that two of the cameras are at the same position.

(b) If each camera center lie on one of two conics (one ellipse and one hyperbola) whose support planes are orthogonal and the principal axes tangent the conics at each camera's position.

(c) If all cameras and principal axes are contained in a line.

The three cases are illustrated in Figure 5.7.

It becomes clear that the degenerate cases are much less likely to occur in practice for a triplet of images than for a pair.

Comparison

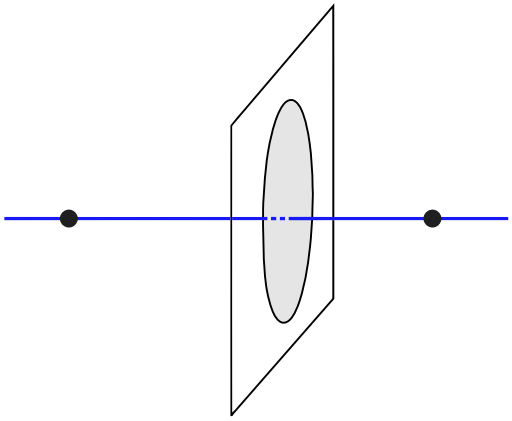
To evaluate the performance of the auto-calibration methods for an image triplet a small experiment was performed. The test environment consisted of three randomly generated synthetic cameras viewing a scene consisting of 750 3D points. The three cameras have a focal lengths of 500, 550 and 600. The projections were disturbed by Gaussian noise. Using the generated image points the trifocal tensor was estimated. Cameras were extracted and structure points triangulated. Finally the reconstruction was improved using bundle adjustment.

Using this projective reconstruction the focal lengths of the three cameras were then estimate using different methods. The three methods considered were

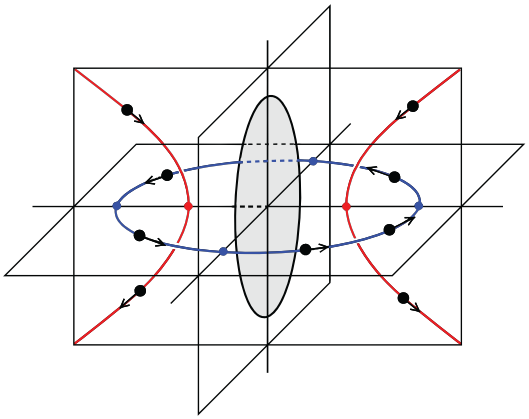
1. Linear method.
2. Quasi-linear method (Iterative improvement).
3. Quasi-linear method followed by metric upgrade and bundle adjustment.

The test was repeated 1000 times and in Figure 5.8 we can see the resulting relative errors.

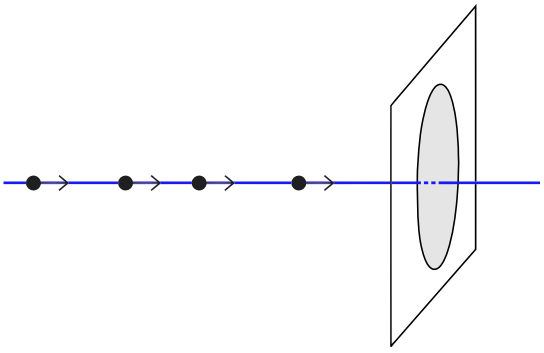
We can see that the quasi-linear method has slightly better performance than the linear method. Performing the metric upgrade and performing bundle adjustment has the best performance.



(a) Only two camera centers.



(b) Cameras lie on one of two conics with support planes which are orthogonal. The principal axes tangent the conics.



(c) Camera centers and principal axes are contained within a line.

Figure 5.7: The degenerate configurations for auto-calibration with unit aspect ratio, zero skew and principal point at the origin. The images are from Kahl [17] and are used with permission.

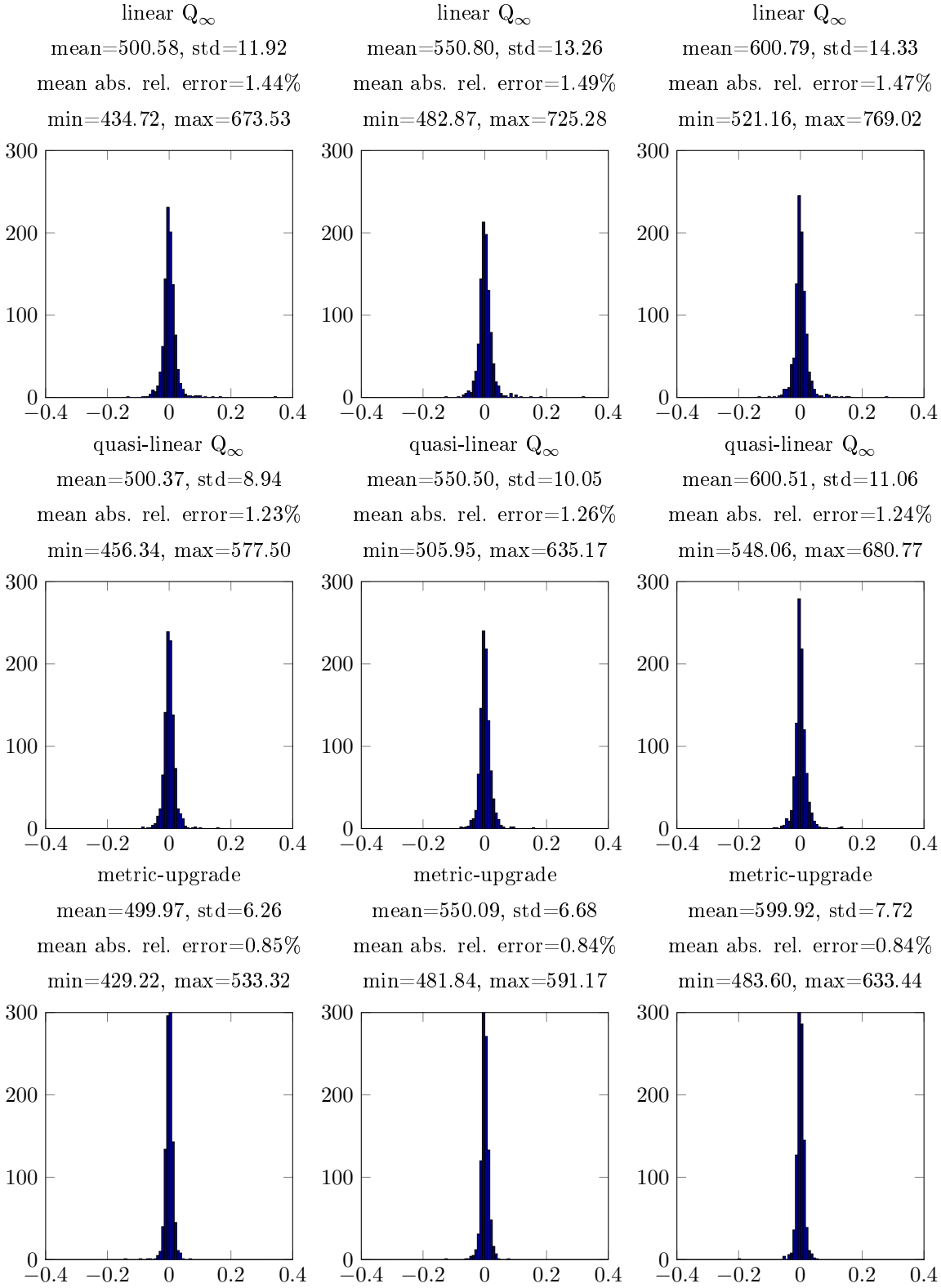


Figure 5.8: Relative errors for the different methods of Q^*_∞ estimation.

5.5 RANSAC

Random Sample and Consensus (RANSAC) is a technique for robust model fitting where there are outliers present in the data. The basic idea is to randomly sample the minimum number of points required to construct the model and then validate it against the whole dataset. This is then iterated enough times so that there is a high probability that an outlier free sample has been chosen. The parameters which fit the most data points during validation is then taken as the model. Optionally the model can then be refitted using the inliers.

5.6 Bundle adjustment

Once an initial reconstruction of a scene is attained it can be improved using *bundle adjustment*. Bundle adjustment is simply using nonlinear optimization methods for minimizing the reprojection error over all the parameters (cameras and structure points).

The problem we solve is

$$\min_{\{P,\mathbf{X}\}} \sum_{i,j} \left|\left| \left(x_j^i - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j},\ y_j^i - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right) \right|\right|^2 . \tag{5.49}$$

This problem can be solved efficiently using the Levenberg-Marquardt method. For more information on this and bundle adjustment in general the reader is referred to Triggs et. al [25].

Since each pair of cameras correspond to a fundamental matrix we can use this as a parametrization of F. First two cameras are chosen from an initial F estimate and the corresponding 3D points are triangulated. Then bundle adjustment on the cameras and 3D points is performed. Constructing F from the updated cameras gives us a better estimation of F. This can of course be done similarly for the trifocal tensor.

Chapter 6

Metric 3D-reconstruction from general image collections

In the previous chapters we have defined objects, and presented methods for estimation, which will serve as building blocks for the final metric reconstruction pipeline. These objects have mostly been focused on the projective geometry of two or three cameras viewing a scene. In this chapter we will present the proposed framework for metric 3D-reconstruction for general image collections.

First we will present the steps used to create metric reconstructions from image pairs and triplets using the estimation methods presented in the previous chapter. Then the pipeline for unordered structure from motion is presented. The pipeline is adaptation of the pipeline for calibrated SfM presented by Olsson and Enqvist in [20]. The steps of the proposed pipeline can be seen in Figure 6.1.

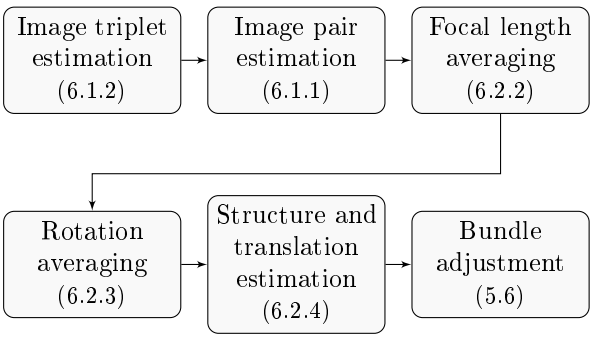


Figure 6.1: The proposed pipeline.

6.1 Metric reconstruction of image pairs and triplets

As an intermediate step towards a full metric reconstruction of a scene, we compute partial reconstructions from image pairs and triplets.

The input to each problem is the set of putative point correspondences between the images. The goal is then to find the focal lengths and the relative rotations between the cameras. Since the problem of auto-calibration is very sensitive and a bad projective reconstruction could lead to very inaccurate focal length estimates we perform a series of *sanity checks* during the reconstruction.

6.1.1 Image pairs

1. Outlier detection
The first step is to determine which of the point correspondences are inliers. To do this we perform RANSAC (Section 5.5) with the 7 point algorithm for fundamental matrix estimation (Section 5.2.4). For each point correspondence the Sampson approximation of the geometric error is computed (Section 5.1.3).

Sanity check: If there is not enough points left after removing outliers the pair is discarded. In the implementation this threshold is set at 50 points.

2. Re-estimation
Using the inliers from the previous step we compute the fundamental matrix using the algebraic minimization algorithm (Section 5.2.2). The fundamental matrix estimate is improved by bundle adjustment (Section 5.6).

3. Auto-calibration
Hartley's method is then used to extract focal length estimates from the fundamental matrix (Section 5.4.1). If the camera configuration is degenerate we save the degeneracy condition on the focal lengths and return.

Sanity check: If the focal length is negative or complex the pair is discarded.

4. Metric reconstruction
Assuming that the cameras are in a non-degenerate configuration and a focal length estimate was obtained from the previous step we compute the essential matrix with

$$E = K_2^T F K_1$$

From the essential matrix we can extract calibrated cameras and structure points are then triangulated.

5. Bundle adjustment
Bundle adjustment is then performed to improve the solution (Section 5.6). The optimization is performed over all parameters: focal lengths, rotations, translations and structure points.

Sanity check: If there is not enough points with reprojection error less than 5 pixels the pair is discarded.

Sanity check: If the focal lengths are out of bounds the pair is discarded. The bounds are defined relative to the size of the image diagonal. The bounds used in

the implementation are (0.5, 5). Thus for an image with resolution 2000×1000 (diagonal of ≈ 2236) the focal length is accepted if it lies within the interval $(1118, 11180)$.

6.1.2 Image triplets

1. Outlier detection
First outliers are found by performing RANSAC (Section 5.5) with the 6 point algorithm for trifocal tensor estimation (Section 5.3.4). The geometric error for each point correspondences is estimated by computing the Sampson approximation (Section 5.1.3).

Sanity check: If there is not enough points left after removing outliers the triplet is discarded. In the implementation this threshold is set at 100 points.

2. Re-estimation
Using the inliers from the previous step we compute the trifocal tensor using the algebraic minimization algorithm (Section 5.3.2). By extracting cameras from the trifocal tensor and triangulating the 3D points we get an initial projective reconstruction of the scene.

3. Improving the reconstruction
So far the we have only used points which are seen in all three images. The set of structure points is increased by triangulating points which are seen in only two of the images. Points which have a reprojection error larger than 5 pixels are removed. Bundle adjustment is performed to improve the reconstruction (Section 5.6).

Sanity check: If there are not enough points left after removing the points with large reprojection error the triplet is discarded.

4. Auto-calibration
The absolute quadric Q_∞^* is then estimated using the (quasi-)linear algorithm (Section 5.4.2). From the eigenvalue decomposition of Q_∞^* we extract the metric upgrade which is applied to the cameras and structure points (Section 5.4.2).

Sanity check: If the Q_∞^* estimate is not semi-definite we discard the triplet.

5. Bundle adjustment
Bundle adjustment is then performed to improve the solution (Section 5.6). The optimization is performed over all parameters: focal lengths, rotations, translations and structure points.

Sanity check: If there is not enough points with reprojection error less than 5 pixels the triplet is discarded.

Sanity check: If there is not enough points which lie in front of all the camera the triplet is discarded. (Chirality)

Sanity check: If focal lengths are out of bounds the triplet is discarded.

Sanity check: If each pair within the triplet does not share at least 50 successful points we discard the triplet.

6.2 Unordered Structure from Motion

Now we have all the building blocks required to perform a full metric reconstruction. The reconstruction will be using a slightly modified version of the unordered structure from motion pipeline for images with known calibration presented by Olsson and Enqvist in [20].

The Olsson-Enqvist pipeline consists of three main steps. First the relative rotations between each pair of cameras is computed. Then an assignment of absolute rotations that is consistent with the relative rotations is found using a RANSAC-like procedure. By absolute rotations we mean that the rotations are assigned with respect to the same coordinate system. Finally the structure and translation is estimated which can be done efficiently due to the rotation being known.

We will now briefly review the different steps of the pipeline and highlight the changes needed to deal with uncalibrated image collections.

6.2.1 Determining relative rotations and focal lengths

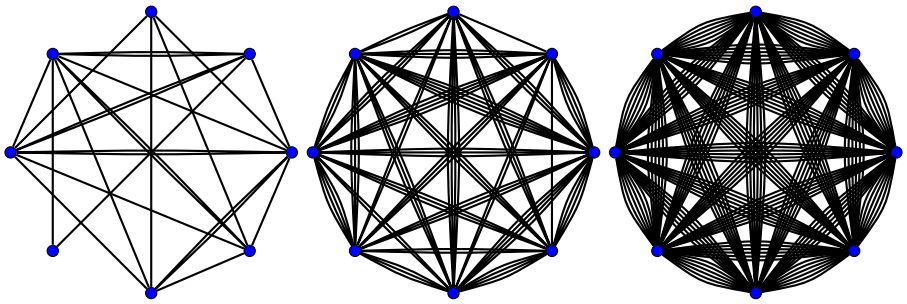
In the first step we want to find the relative rotations between the cameras. The main difference from the Olsson-Enqvist pipeline is that since we are dealing with uncalibrated images we have to also estimate the calibration. To do this we use the methods for metric reconstructions of image pairs and triplets which was described in Section 6.1. For each triplet we get three relative rotations and three focal length estimates and for each pair we get one relative rotation and two focal length estimates or if the pair was degenerate we get a degeneracy condition for the focal lengths (see Section 5.4.1).

A naive approach to selecting which triplets to perform the metric reconstruction for is to simply selecting all triplets which have enough point correspondences. While this approach is reasonable for pairs it becomes very computationally expensive for triplets. This is due to the number of possible triplets of n images grows as $\binom{n}{3} \approx n^3$, e.g. if we have 200 images there exists $\binom{200}{3} = 1313400$ possible triplets. There is also a diminishing return for including more triplets since it is possible to get multiple estimates of the same relative rotation.

Another extreme is choosing the best (most shared points) triplet for each image. This results in a very clustered camera graph with low connectivity.

The heuristic approach proposed here is to select the best triplet for each pair of cameras. That is for each pair of images (i, j) we select the best image $k \notin \{i, j\}$ such that the triplet (i, j, k) is not already selected. By best image we mean the image k which shares the most points with the pair (i, j). This approach gives a more connected camera graph while choosing at most $\binom{n}{2}$ triplets. For comparison $\binom{200}{2} = 19900$. See Figure 6.2 for a comparison of the three approaches.

For pairs we simply select every pair which did not get a relative orientation from the triplet estimation.



(a) One triplet per camera. (b) One triplet per pair. (c) All triplets.

Figure 6.2: Example of the resulting pairwise estimates for the three approaches for selecting triplets. The eight vertices correspond to cameras and the edges correspond to the resulting pairwise relative rotations. We see that in (a) there is a camera which can get disconnected by a single triplet estimation failing.

Now we have estimates of both the focal lengths and the relative rotations between the cameras. Since each camera can be part of many triplets and pairs we will have multiple focal length estimates for each camera. In the next step will consolidate the focal length estimates to a single estimate for each camera.

6.2.2 Robust focal length averaging

As part of the reconstruction pipeline we have to assign a focal length to each camera. From the previous steps we get focal length estimates for pairs and there can be multiple estimates for each pair of cameras. Due to the sensitive nature of auto-calibration some of these pairs will be outliers and very far from the true focal length. To further complicate things we have that some of the pairs were in a degenerate configuration and instead of a focal length estimate we have a condition on the form $G(f_i, f_j) = 0$. The goal is now to consolidate these estimates into a single focal length estimate for each camera.

To accomplish this we consider the two types of estimates we have, degenerate and non-degenerate. For a non-degenerate pair (i, j) we have a focal estimate for the two cameras. Let $\hat{\mathbf{f}}_{i,j}$ be a vector with the focal length estimate from the pair. We then construct a cost function associated with this estimate

$$C_{ij}(\mathbf{f}) = ||\hat{\mathbf{f}}_{i,j} - \begin{pmatrix} f_i \\ f_j \end{pmatrix}||. \tag{6.1}$$

For a degenerate pair (i, j) we instead have a condition on the focal lengths. Let $G(f_i, f_j) = 0$ be the condition. We let the cost function associated with this pair be

$$D_{ij}(\mathbf{f}) = \min_{a,b} ||\begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} f_i \\ f_j \end{pmatrix}|| \quad \text{s.t.} \quad G(a, b) = 0, \tag{6.2}$$

i.e. we take the minimum distance to the implicit curve G as the cost. To avoid having to solve the minimization problem we instead consider the first order approximation of the distance. See Section 6.2.2.

To find an estimate of all the focal lengths we then want to find an assignment $\mathbf{f}$ that minimizes the total cost, i.e. we want to solve the following optimization problem

$$\min_{\mathbf{f}} \sum_{i,j} C_{ij}(\mathbf{f})^2 + \sum_{i,j} D_{ij}(\mathbf{f})^2, \tag{6.3}$$

where the sums are only taken over the pairs which there are estimates for.

To account for the varying certainty of the pairs we add weights to the terms.

$$\min_{\mathbf{f}} \sum_{i,j} w_{ij} C_{ij}(\mathbf{f})^2 + \sum_{i,j} w_{ij} D_{ij}(\mathbf{f})^2 \tag{6.4}$$

The weights are chosen to be the number of inliers found when the pair was estimated.

Some of the estimates might be outliers and thus be very far from the true focal length. To prevent these from affecting the final assignment we use a robust cost function. The cost function we consider is a truncated quadratic.

$$\varphi(x) = \begin{cases} x^2 & \text{if} \quad |x| < \delta \\ \delta^2 & \text{if} \quad |x| > \delta \end{cases}. \tag{6.5}$$

See Figure 6.3. This cost functions only considers points close to the current estimate since estimates which are far away are given a constant cost. Note that which points are inliers can change during the optimization.

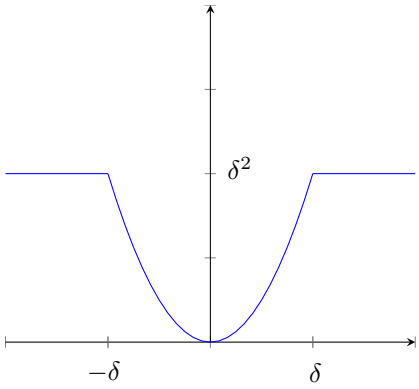


Figure 6.3: The robust error function φ.

To let the cost functions be invariant to different resolutions for different images we divide all the focal lengths by the size of the corresponding image diagonal. The reasoning behind this normalization is that the costs should allow for relative errors should be approximately the same size regardless of image resolution. This also allows us to use a single δ for all images. Typically $\delta = 0.5$ in the implementation.

Finally the optimization problem we consider is

$$\min_{\mathbf{f}} \sum_{i,j} w_{ij}\ \varphi(C_{ij}(\mathbf{f})) + \sum_{i,j} w_{ij}\ \varphi(D_{ij}(\mathbf{f})). \tag{6.6}$$

Which we solve using Levenberg-Marquardt. The initial guess is taken as the median of the non-degenerate estimates for each camera.

Degenerate pairs

From image pairs in degenerate configurations we get a constraint on the focal lengths which is on the form

$$G(f_1, f_2) = af_1^2 + bf_1^2 f_2^2 + cf_2^2 = 0. \tag{6.7}$$

Due to noise this constraint will probably not be satisfied exactly by the real focal lengths. Instead we consider the distance to the curve $G(f_1, f_2) = 0$ as the cost for a candidate focal length assignment.

The distance to an implicit curve $G(\mathbf{f}) = 0$ from a point $\mathbf{f_0}$ is given by

$$\min_{\mathbf{f}} ||\mathbf{f} - \mathbf{f_0}||^2 \quad \text{s.t.} \quad G(\mathbf{f}) = 0. \tag{6.8}$$

Since this optimization problem is somewhat troublesome we instead consider the first order approximation of the constraint. The Taylor expansion of G around $\mathbf{f_0}$ is given by

$$G(\mathbf{f}) = G(\mathbf{f_0}) + \nabla G(\mathbf{f_0})^T(\mathbf{f} - \mathbf{f_0}). \tag{6.9}$$

Using this as the constraint in our optimization problem we get

$$\min_{\mathbf{f}} ||\mathbf{f} - \mathbf{f_0}||^2 \quad \text{s.t.} \quad G(\mathbf{f_0}) + \nabla G(\mathbf{f_0})^T(\mathbf{f} - \mathbf{f_0}) = 0. \tag{6.10}$$

The Lagrangian function is given by

$$L(\mathbf{f}, \lambda) = (\mathbf{f} - \mathbf{f_0})^T(\mathbf{f} - \mathbf{f_0}) + 2\lambda(G(\mathbf{f_0}) + \nabla G(\mathbf{f_0})^T(\mathbf{f} - \mathbf{f_0})). \tag{6.11}$$

Setting the partial derivatives to zero we get

$$L_{\mathbf{f}} = 2(\mathbf{f} - \mathbf{f_0})^T + 2\lambda \nabla G(\mathbf{f_0})^T = 0, \tag{6.12}$$

$$L_\lambda = G(\mathbf{f_0}) + \nabla G(\mathbf{f_0})^T(\mathbf{f} - \mathbf{f_0}) = 0. \tag{6.13}$$

Inserting the first equation into the second we get

$$G(\mathbf{f_0}) - \lambda ||\nabla G(\mathbf{f_0})||^2 = 0 \quad \Rightarrow \quad \lambda = \frac{G(\mathbf{f_0})}{||\nabla G(\mathbf{f_0})||^2}. \tag{6.14}$$

Substituting into the first equation we get

$$\mathbf{f} - \mathbf{f_0} = -G(\mathbf{f_0}) \frac{\nabla G(\mathbf{f_0})}{||\nabla G(\mathbf{f_0})||^2}. \tag{6.15}$$

Taking the norm

$$||\mathbf{f} - \mathbf{f_0}|| = \frac{|G(\mathbf{f_0})|}{||\nabla G(\mathbf{f_0})||}. \tag{6.16}$$

Which is the first order approximation of the distance to the implicit curve $G(\mathbf{f}) = 0$ from the point $\mathbf{f_0}$. For a more in-depth study of distance approximations to implicit curves the reader is referred to Taubin [23].

To avoid problems where $||\nabla G(\mathbf{f_0})||$ is very close to zero we instead use the approximation

$$||\mathbf{f} - \mathbf{f_0}|| \approx \frac{|G(\mathbf{f_0})|}{||\nabla G(\mathbf{f_0})|| + \epsilon}, \tag{6.17}$$

where $\epsilon > 0$ is a small number. In the implementation $\epsilon = 10^{-8}$ is used.

6.2.3 Rotation averaging

In a previous step we have estimated the relative rotations between the cameras. Now the goal is to use this information and assign the absolute rotations of the cameras.

The relative rotations from the previous step can be organized as a graph where the cameras are on the vertices and the relative rotations are on the edges. Since each pair can be part of multiple triplets we can have multiple relative rotations and thus edges for each pair of vertices. For each edge we also define a weight, which in the Olsson-Enqvist pipeline is the number of inliers used for the rotation estimation

$$w_e = n_{inl}. \tag{6.18}$$

The goal is now to assign an absolute rotation for each camera that is consistent with as many of the relative rotations as possible. In the Olsson-Enqvist [20] framework this is done by randomly selecting a spanning tree from the graph and then assigning the rotations starting from the root. See Figure 6.4.

The edges chosen when constructing the spanning tree are chosen with a probability that is proportional to the edge weight, i.e. number of inliers used for estimation. The success of the assignment is measured by counting the number of relative rotations which are consistent with the assignment. This procedure is then iterated and the best assignment is kept.

Since the quality of the rotation estimate depends on how well the auto-calibration went we instead use the edge weights

$$w_e = \frac{n_{inl}}{1 + ||\hat{\mathbf{f}}_e - (f_i, f_j)^T||}, \tag{6.19}$$

where $\hat{\mathbf{f}}_e$ is the focal length estimate from the relative rotation estimation and (f_i, f_j) are the assigned focal lengths from the focal length averaging. This will put a lower weight on rotation estimates where the focal length was far from the average.

To decide which rotations are consistent with an assignment we need to be able to decide how close two rotations are. The metric we use is given by

$$d(R_1, R_2) = ||\log(R_1^T R_2)||, \tag{6.20}$$

which corresponds to the angular difference between the rotations. For a comparison of different metrics for rotations the reader is refered to Huyhn [15].

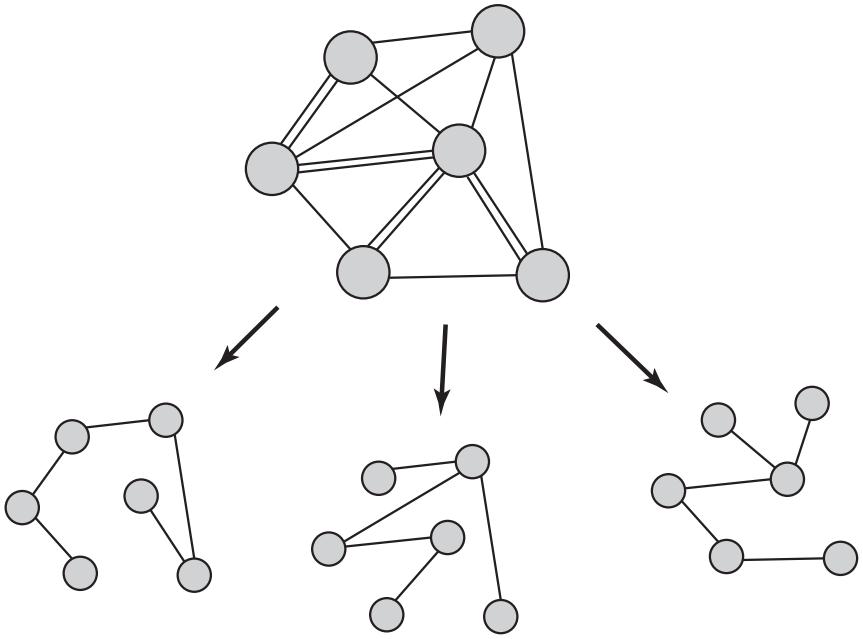


Figure 6.4: From the camera graph we randomly select spanning trees.

6.2.4 Translation and structure estimation

In the previous steps we have estimated the rotation matrices R and the calibration matrices K for all the cameras. The remaining unknowns are the positions of the cameras $\mathbf{t}$ and the structure points $\mathbf{X}$. Now assume that all image points have been transformed by the inverse of their corresponding calibration matrix. The cameras we are searching for will then be calibrated cameras on the form $P = [R\ \mathbf{t}]$.

The reprojection error of an observed image point $\mathbf{x}_i$ is given by

$$r_i(\mathbf{X}, \mathbf{t}) = \left|\left|\left(x_i - \frac{R_1\mathbf{X} + t_1}{R_3\mathbf{X} + t_3},\ y_i - \frac{R_2\mathbf{X} + t_2}{R_3\mathbf{X} + t_3} \right)\right|\right|_2, \tag{6.21}$$

where R_k is the kth row of the rotation matrix R. By stacking the unknowns in a vector $\mathbf{z}$ we can write the residual on the form

$$r_i(\mathbf{z}) = \left|\left|\left(\frac{\mathbf{a}_i^T\mathbf{z} + \alpha_i}{\mathbf{c}_i^T\mathbf{z} + \gamma_i}, \frac{\mathbf{b}_i^T\mathbf{z} + \beta_i}{\mathbf{c}_i^T\mathbf{z} + \gamma_i} \right)\right|\right|_2. \tag{6.22}$$

Now consider the problem of minimizing the largest residual, i.e.

$$\min_{\{\mathbf{X},\mathbf{t}\}} \max_i \ r_i. \tag{6.23}$$

This problem can also be written as

$$\min_{\{\epsilon,\mathbf{X},\mathbf{t}\}} \ \epsilon \quad \text{s.t.} \quad \forall i \quad r_i \leq \epsilon. \tag{6.24}$$

For points lying infront of the camera it holds that $R_3\mathbf{X} + t_3 > 0$ and thus that $\mathbf{c}_i^T\mathbf{z} + \gamma_i > 0$. Using this we can rewrite (6.24) as

$$\begin{aligned} \min_{\{\epsilon,\mathbf{X},\mathbf{t}\}} \quad & \epsilon \\ \text{s.t.} \quad & ||(\mathbf{a}_i^T\mathbf{z} + \alpha_i,\ \mathbf{b}_i^T\mathbf{z} + \beta_i)||_2 \leq \epsilon(\mathbf{c}_i^T\mathbf{z} + \gamma_i). \end{aligned} \tag{6.25}$$

To make the problem more tractable Olsson and Enqvist instead consider the problem where the supremum norm is used in the residuals, i.e.

$$r_i(\mathbf{z}) = \max\left(\left|\frac{\mathbf{a}_i^T\mathbf{z} + \alpha_i}{\mathbf{c}_i^T\mathbf{z} + \gamma_i}\right|,\ \left|\frac{\mathbf{b}_i^T\mathbf{z} + \beta_i}{\mathbf{c}_i^T\mathbf{z} + \gamma_i}\right|\right). \tag{6.26}$$

Then the constraints in the problem (6.25) become linear for fix ϵ.

The problem with minimizing the largest residual is that it behaves poorly when there are outliers present in the data. To deal with this Olsson and Enqvist propose that we instead consider the optimization problem

$$\begin{aligned} \min_{\{\epsilon,s,\mathbf{X},\mathbf{t}\}} \quad & \sum_i s_i \\ \text{s.t.} \quad & ||(\mathbf{a}_i^T\mathbf{z} + \alpha_i,\ \mathbf{b}_i^T\mathbf{z} + \beta_i)||_\infty \leq \epsilon(\mathbf{c}_i^T\mathbf{z} + \gamma_i) + s_i \\ & s_i \geq 0 \end{aligned} \tag{6.27}$$

For fix ϵ this problem is a linear program and can be solved efficiently.

The resulting reconstruction is then improved using bundle adjustment.

Chapter 7

Results

7.1 Synthetic images

To verify that the proposed pipeline for metric reconstruction works, we will first consider synthethic images.

The synthetic scene consists of 750 3D points. See Figure 7.1. The scene is viewed by 10 randomly generated cameras. The cameras have a focal length of 2000. The image points are then disturbed by Gaussian noise with varying standard deviation σ.

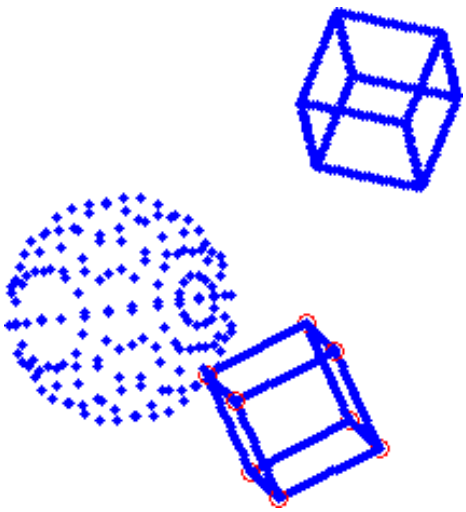


Figure 7.1: The synthetic 3D scene consisting of two cubes and a sphere. The red circles correspond to the eight corner points used for validation.

To measure the success of the reconstructions we consider three different metrics.

- **RMS** The Root Mean Square of the reprojection error for all the points.

The RMS error is given by

$$RMS = \sqrt{\frac{1}{N}\sum_i ||\mathbf{x}_i - \bar{\mathbf{x}}_i||^2}, \tag{7.1}$$

where $\mathbf{x}_i$ is the observed image point and $\bar{\mathbf{x}}_i$ is the reprojection.

- $\boldsymbol{\Delta f}$ The mean of the absolute focal length error.

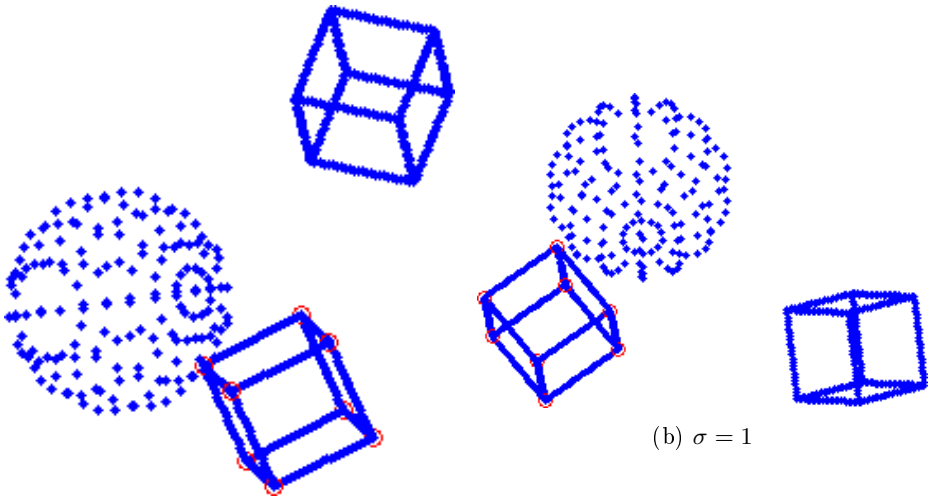
$$\Delta f = \frac{1}{N}\sum_i |f_i - 2000|. \tag{7.2}$$

- $\boldsymbol{\Delta\theta}$ The mean angular error. We want to measure the effect of errors in the calibration. To do this we save the true projections (without noise) of the eight corners of one of the cubes in the scene. From these we can then re-triangulate the eight corners of the cube using the resulting camera estimates. For each of the eight corners we compute the three angles to the adjoining corners which ideally should be 90°. This gives us $8 \times 3 = 24$ angles. We let $\Delta\theta$ to be the mean absolute error of the angles.

Reconstructions were performed for $\sigma = 0, 1, 5$, and 10. The resulting errors can be seen in Table 7.1. In Figure 7.2 the resulting structure points can be seen for varying σ and in Figure 7.3 shows the reprojections of the first camera.

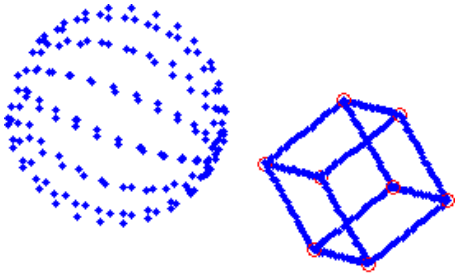
σ	0	1	5	10
RMS	4.0359e-13	1.2881	5.2347	4.379
Δf	1.4779e-12	0.75704	3.9972	10.297
$\Delta\theta$	6.7798e-13	0.0046645	0.024143	0.076394

Table 7.1: The resulting errors after reconstructing the synthetic scene with varying noise levels. The RMS error and focal length is measured in pixels and the angular error is measured in degrees.

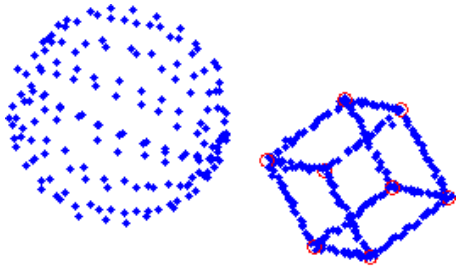


(a) $\sigma = 0$

(b) $\sigma = 1$

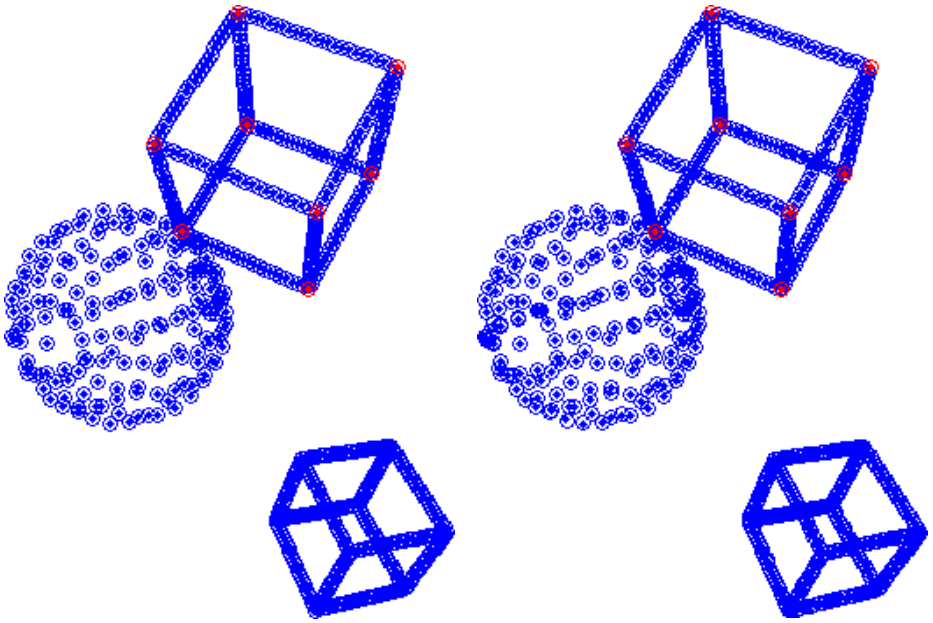


(c) $\sigma = 5$



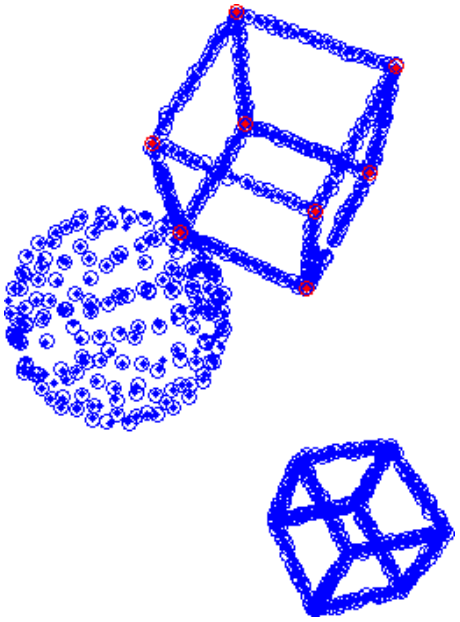
(d) $\sigma = 10$

Figure 7.2: The resulting structure points for each noise level. The red circles correspond to the triangulated validation points.

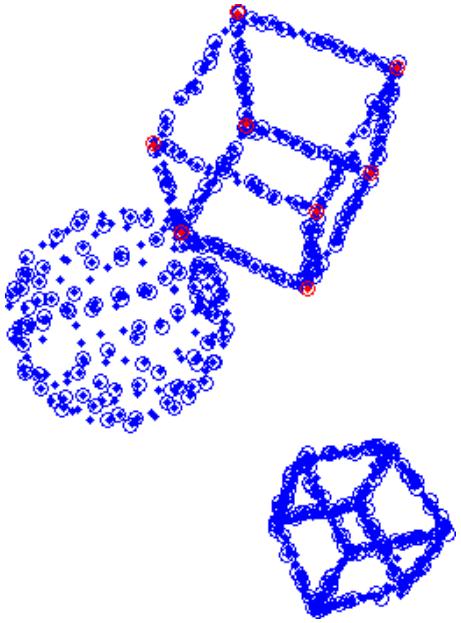


(a) $\sigma = 0$

(b) $\sigma = 1$



(c) $\sigma = 5$



(d) $\sigma = 10$

Figure 7.3: The reprojections in the first camera. The dots are the reprojections and the circles the image points. Missing circles correspond to points being classified as outliers in this camera. The red points correspond to the validation data.

7.2 Real images

In this section we will now present reconstructions of real scenes built using the proposed pipeline.

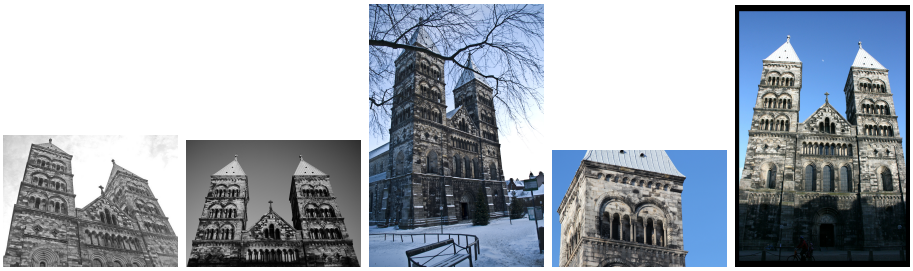
7.2.1 Lund Cathedral

The first reconstruction is built from 236 images of the cathedral in Lund. The images comes from three different sources:

- 79 images taken with a DSLR camera. The images were provided by Carl Olsson. The images have a resolution of 1936×1296 and have a focal length of approximately 2500.

- 75 images taken with a mobile phone camera. The images have a resolution of 2048×1232 and have a focal length of approximately 1800.

- 82 images from the online image sharing website www.flickr.com. The images have varying resolutions ranging from 427×640 to 5184×3456. The images were found using the queries *lund domkyrka* and *lund cathedral*.



The images were calibrated independently and no information of the origins of each image was used in the pipeline.

The resulting reconstruction contains 60595 structure points seen in 190 cameras. See Figure 7.5, 7.6, and 7.7. Due to memory limitations only points which were seen in 7 or more images were included. To build the reconstruction 10327 image triplets were estimated. Of the 10327 triplets there were 6973 which were successfully estimated. In Table 7.2 the different failure points of the triplet

#	%	Result
6973	67.52	Successfully reconstructed.
25	0.24	Not enough inliers after trifocal RANSAC.
1	0.01	Not enough inliers after full trifocal estimation.
1566	15.16	Q_∞^* not semi-definite.
61	0.59	Not enough inliers after bundle adjustment.
505	4.89	Chirality failed.
979	9.48	Focal lengths out of bounds.
217	2.10	Not enough pairwise inliers.
10327	100	Total

Table 7.2

estimation can be seen. Of the failed image triplets the most common failure was an indefinite Q_∞^* estimate.

After image triplet estimation 4849 image pairs were selected. The pairs that were selected were those that did not get a successful relative rotation in the previous step. Of those pairs only 2296 (47.35%) were successfully reconstructed. The lower success rate than for triplets can be explained by there being a selection bias towards pairs with low number of matches or many outliers (since a good pair is more likely to have succeeded during the previous step). 297 pairs were classified as degenerate.

Of the 236 images there were 190 which were able to be successfully estimated. For the three different image sources we had; DSLR images 77 or 79, Mobile camera 60 of 75, and Flickr 53 of 82. In Figure 7.4 we can see the resulting focal lengths, reprojection errors and number of points seen for each of the image sources. The images from the DSLR camera gave the lowest reprojection error and saw the most points in average.

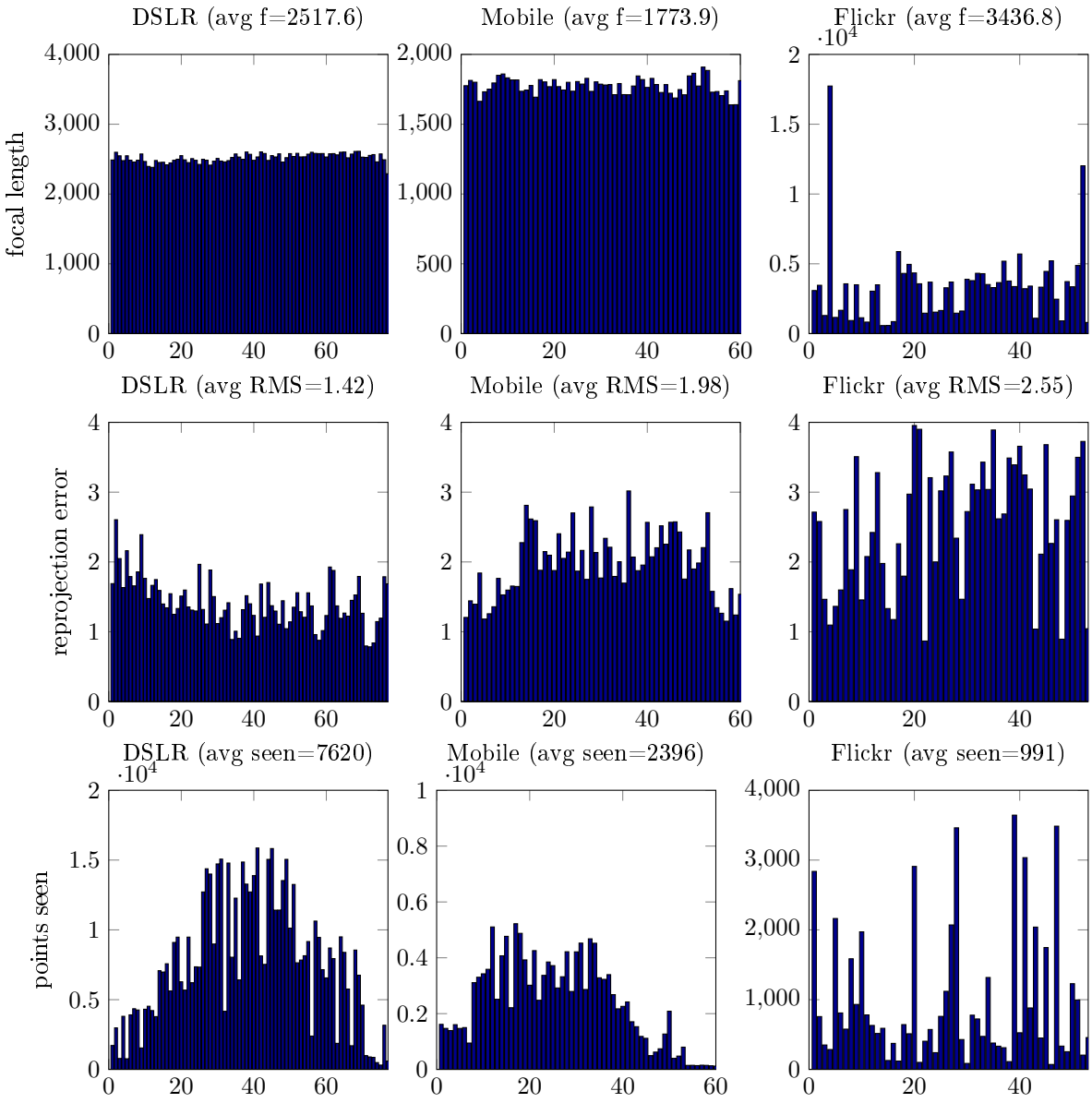


Figure 7.4: The resulting focal lengths, reprojection errors and number of points seen for each image for the three different image sources.

Figure 7.5: Lund Cathedral

Figure 7.6: Lund Cathedral

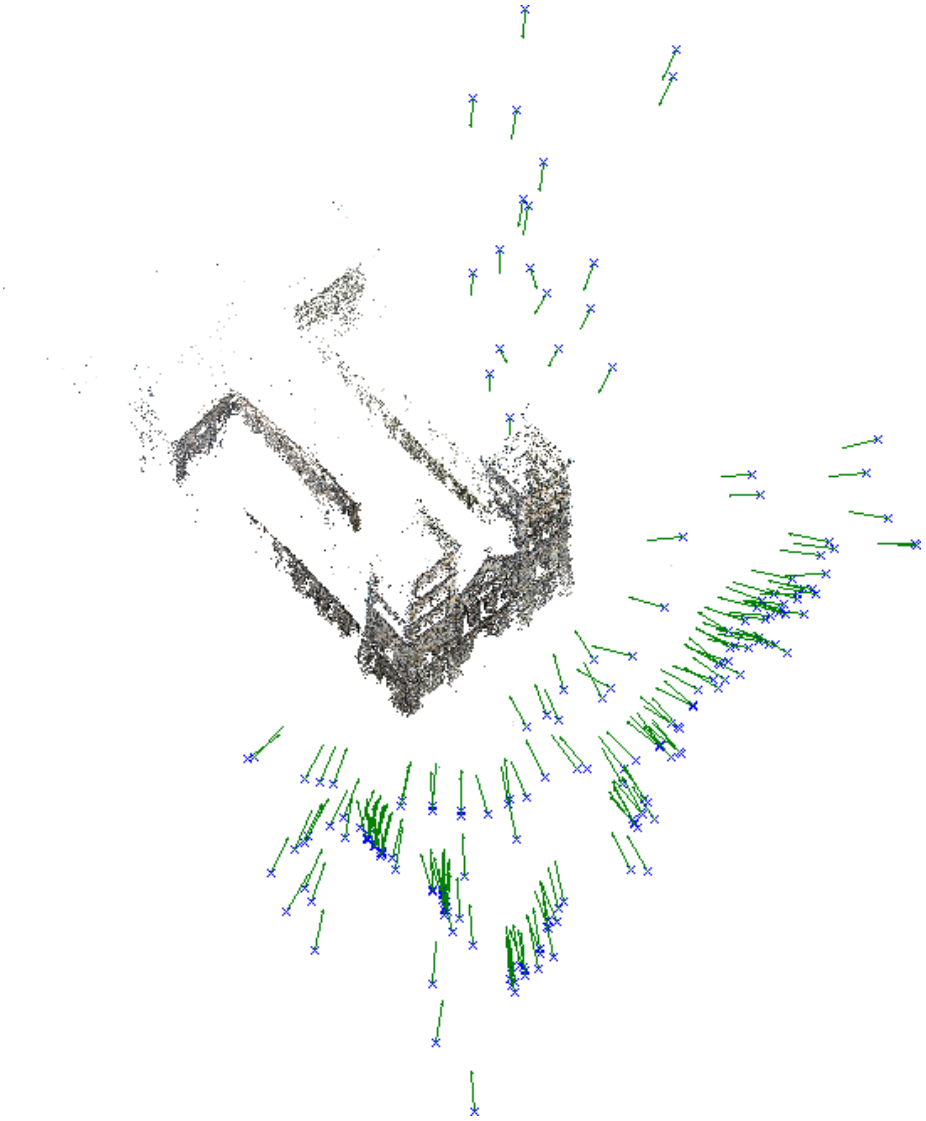


Figure 7.7: Lund Cathedral including camera positions.

7.2.2 Oxford dinosaur

Next we consider a reconstruction of the well-known Oxford dinosaur sequence. The image sequence consists of a camera rotated around a toy dinosaur. Since the cameras point toward the same point on the rotation axis, the motion becomes degenerate for pairs. The image sequence contains 36 images with a resolution of 720×576. Some of the images can be seen in Figure 7.8.

Figure 7.8: Sample images of the Oxford dinosaur sequence.

The resulting reconstruction consists of 1508 structure points seen in 36 images. See Figure 7.9 and Figure 7.10. During the reconstruction 34 pairs were estimated, 31 were correctly classified as degenerate.

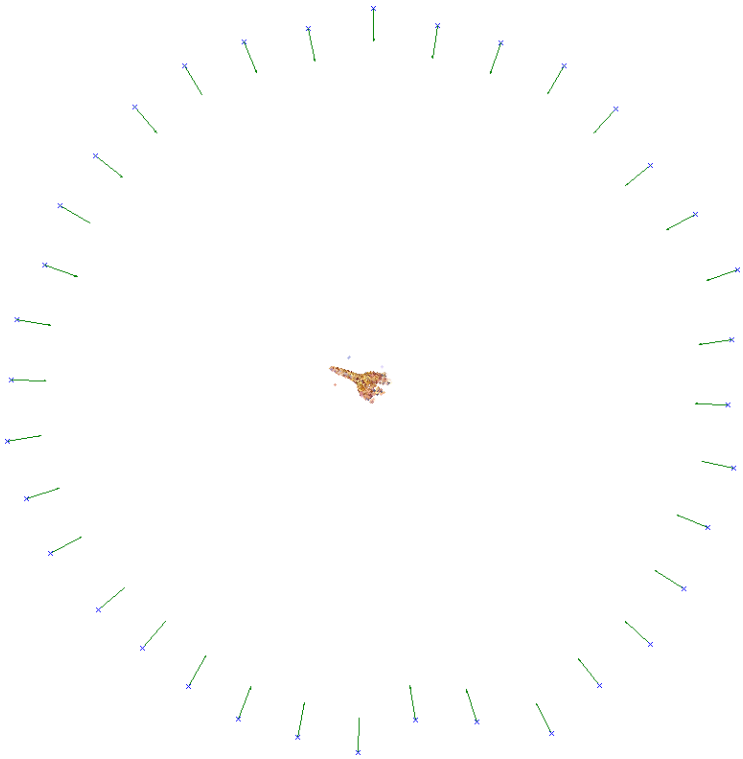


Figure 7.9

Figure 7.10: Reconstruction of the Oxford dinosaur.

Chapter 8

Discussion

In this thesis we have considered the problem of structure from motion for uncalibrated and unordered image collections. The focus has been on making auto-calibration robust by combining the estimates from many small auto-calibration problems. Originally the idea was to only perform the auto-calibration for image pairs, but the troublesome nature of the degenerate cases for pairwise auto-calibration led us to instead consider image triplets.

The auto-calibration problem is very sensitive to camera positions, and configurations which are close to degenerate often gives very inaccurate estimates. Thus it is of interest to discover when the cameras are in a degenerate configuration. For a pair of cameras we know (see Section 5.4.1) that the fundamental matrix can be used to decide if the configuration is degenerate. The method proposed for checking if a pair is close to degenerate only considers how close the fundamental matrix is to the degenerate form. This gives a quite crude estimate and perhaps a more thorough study could reveal a better method. Also for image triplets we have not considered how to decide if a configuration is degenerate.

For a degenerate pair we instead of focal length estimates get a condition on the focal lengths (see Section 5.4.1) which is on the form

$$G(f_1, f_2) = af_1^2 + bf_1^2 f_2^2 + cf_2^2 = 0. \tag{8.1}$$

In the focal length averaging we use the distance to the implicit curve $G(f_1, f_2) = 0$ as a cost function. We have not considered how noise in the coefficients (a, b, c) affect the cost function.

In Section 6.2.3 we have to decide which image triplets to perform auto-calibration for. The proposed method is to for each pair select the triplet with the most point correspondences. If the chosen triplet has already been used for another pair the second best triplet is chosen instead and so on. This method has the drawback that it depends on the order of the images. A possible improvement would be to instead consider which triplets would increase the connectivity of the camera graph the most.

In the proposed pipeline we estimate the rotations and the focal lengths at

the same time. The focal lengths and rotations are then averaged separately. Another approach might be to re-estimate the rotations using the resulting focal lengths from the focal length averaging. This was not done due to the excessive computational cost.

It can also be hard to distinguish when the calibration has failed since it is possible to end up in a projective reconstruction which has very small reprojection errors but is very far from the true metric scene. Checking that all points lie in front of the cameras can be used in some cases to decide if the reconstruction was successful. This is captured by the *chirality constraints*. These constraints can also be used when performing the auto-calibration. In [2] Chandraker et. al propose a non-linear optimization scheme for estimating the absolute quadric Q_∞^* which can enforce chirality.

In Section 5.4.2 a quasi-linear method for estimating the absolute quadric Q_∞^* is described. This method does not enforce the constraint that Q_∞^* should be a semi-definite matrix (i.e. all eigenvalues should have the same sign). While the method proposed by Chandraker et. al enforces this constraint it has the drawback of being a lot more computationally expensive. For the Lund cathedral reconstruction (see Section 7.2.1) 15.16% of the image triplets failed due to Q_∞^* not being definite.

Bibliography

[1] The MOSEK optimization software.

[2] M. Chandraker, S. Agarwal, F. Kahl, D. Nister, and D. Kriegman. Autocalibration via rank-constrained estimation of the absolute quadric. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[3] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[4] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd European Conf. on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, 1992.

[5] O. D. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proc. 2nd European Conf. on Computer Vision, Santa Margherita Ligure, Italy*, pages 321–334. Springer-Verlag, 1992.

[6] O.D. Faugeras and S. Maybanks. Motion from point matches: multiplicity of solutions. *Int. J. Computer vision*, 1990.

[7] R. Hartley. Projective reconstruction from line correspondences. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 903–907. IEEE Computer Society Press, 1994.

[8] R. Hartley. Lines and points in three views and the trifocal tensor. *Int. Journal of Computer Vision*, 22(2):125–140, March 1997.

[9] R. Hartley, R. Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *Proceedings IEEE Conf. on Computer vision and Pattern Recognition*, pages 761–764, 1992.

[10] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proceedings of the Second European Conference on Computer Vision*, ECCV '92, pages 579–587. Springer-Verlag, 1992.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[12] R.I. Hartley. Extraction of focal lengths from the fundamental matrix. Unpublished manuscript.

[13] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[14] R.I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Computer Vision, 1998. Sixth International Conference on*, 1998.

[15] D.Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 2009.

[16] Chen J. and B. Yuan. Metric 3d reconstruction from uncalibrated unordered images with hierarchical merging. In *IEEE 10th International Conference on Signal Processing (ICSP)*, 2010.

[17] F. Kahl, B. Triggs, and K. Astrom. Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 2000.

[18] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[19] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, November 2004.

[20] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *in Proceedings of the 17th Scandinavian conference on Image analysis, ser. SCIA11*, pages 524–535. Springer-Verlag, 2011.

[21] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *Workshop on 3D Structure from Multiple Images of Large-Scale Environments*. Springer-Verlag, 1998.

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, New York, NY, USA, 2006. ACM Press.

[23] G Taubin. Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics*, 13:3–42, 1994.

[24] B. Triggs. Autocalibration and the absolute quadric. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1997.

[25] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS. Springer Verlag, 2000.

[26] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.

[27] A. Zisserman and P.H.S. Torr. Robust parameterization and computation of the trifocal tensor. In *Proc. British Machine Vision Conference*, 1996.

Master's Theses in Mathematical Sciences 2013:E42
ISSN 1404-6342

LUTFMA-3252-2013

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/