Long QUAN

Image-based Modeling

– The Techniques of Generating 3D Models from Multiple Images –
Notation

Unless otherwise stated, we use the following notation throughout the book.

- Scalars are lower-case letters (e.g., \(x, y, z\)) or lower-case Greek letters (e.g., \(\alpha, \lambda\)).
- Vectors are bold lower-case letters (e.g., \(\mathbf{x}, \mathbf{v}\)).
- Matrices or tensors are bold upper-case letters (e.g., \(\mathbf{A}, \mathbf{P}, \mathbf{T}\)).
- Polynomials and functions are lower-case letters (e.g., \(f(x, y, z) = 0\)).
- Geometry points or objects are upper-case letters (e.g., \(A, B, P\)).
- Sets are upper-case letters (e.g., \(I, V\)).
- An arbitrary number field is \(k\), the real numbers are \(\mathbb{R}\) and the complex numbers are \(\mathbb{C}\).
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>vi</td>
</tr>
<tr>
<td>Preface</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>x</td>
</tr>
<tr>
<td>Notation</td>
<td>xi</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td><strong>Part I Geometry: fundamentals of multi-view geometry</strong></td>
<td></td>
</tr>
<tr>
<td>2 Geometry prerequisite</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Projective geometry</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 The basic concepts</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Projective spaces and transformations</td>
<td>10</td>
</tr>
<tr>
<td>2.2.3 Affine and Euclidean specialization</td>
<td>16</td>
</tr>
<tr>
<td>2.3 Algebraic geometry</td>
<td>21</td>
</tr>
<tr>
<td>2.3.1 The simple methods</td>
<td>21</td>
</tr>
<tr>
<td>2.3.2 Ideals, varieties, and Gröbner bases</td>
<td>23</td>
</tr>
<tr>
<td>2.3.3 Solving polynomial equations with Gröbner bases</td>
<td>24</td>
</tr>
<tr>
<td>3 Multi-view geometry</td>
<td>29</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>30</td>
</tr>
<tr>
<td>3.2 The single-view geometry</td>
<td>30</td>
</tr>
<tr>
<td>3.2.1 What is a camera?</td>
<td>30</td>
</tr>
<tr>
<td>3.2.2 Where is the camera?</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 The DLT calibration</td>
<td>37</td>
</tr>
<tr>
<td>3.2.4 The three-point pose algorithm</td>
<td>39</td>
</tr>
<tr>
<td>3.3 The uncalibrated two-view geometry</td>
<td>42</td>
</tr>
<tr>
<td>3.3.1 The fundamental matrix</td>
<td>43</td>
</tr>
</tbody>
</table>
3.3.2 The seven-point algorithm ......................... 45
3.3.3 The eight-point linear algorithm ................. 46
3.4 The calibrated two-view geometry .................. 47
  3.4.1 The essential matrix ......................... 47
  3.4.2 The five-point algorithm ................... 49
3.5 The three-view geometry ............................ 53
  3.5.1 The trifocal tensor .......................... 54
  3.5.2 The six-point algorithm ................... 58
  3.5.3 The calibrated three views .................. 63
3.6 The N-view geometry ................................ 66
  3.6.1 The multi-linearities ....................... 66
  3.6.2 Auto-calibration ............................ 68
3.7 Discussions ......................................... 72
3.8 Bibliographic notes .................................. 72

Part II Computation: from pixels to 3D points

4 Feature point ........................................... 77
  4.1 Introduction ........................................ 78
  4.2 Points of interest ................................... 78
    4.2.1 Tracking features ............................ 78
    4.2.2 Matching corners ............................ 80
    4.2.3 Discussions .................................. 81
  4.3 Scale invariance ................................. 82
    4.3.1 Invariance and stability .................... 82
    4.3.2 Scale, blob and Laplacian .................. 82
    4.3.3 Recognizing SIFT ............................ 83
  4.4 Bibliographic notes .............................. 84

5 Structure from Motion ................................. 85
  5.1 Introduction ........................................ 86
    5.1.1 Least squares and bundle adjustment .......... 86
    5.1.2 Robust statistics and RANSAC ................. 88
  5.2 The standard sparse approach ..................... 90
    5.2.1 A sequence of images ....................... 92
    5.2.2 A collection of images ..................... 93
  5.3 The match propagation .............................. 94
    5.3.1 The best-first match propagation .......... 94
    5.3.2 The properties of match propagation ...... 97
    5.3.3 Discussions .................................. 101
  5.4 The quasi-dense approach ......................... 103
    5.4.1 The quasi-dense resampling .................. 103
    5.4.2 The quasi-dense SFM ......................... 104
    5.4.3 Results and discussions .................... 111
  5.5 Bibliographic notes .................................. 117
Part III Modeling: from 3D points to objects

6 Surface modeling .................................................. 121
   6.1 Introduction .................................................. 122
   6.2 Minimal surface functionals ................................. 123
   6.3 A unified functional ......................................... 124
   6.4 Level-set method ............................................ 124
   6.5 A bounded regularization method .......................... 125
   6.6 Implementation .............................................. 127
   6.7 Results and discussions .................................... 129
   6.8 Bibliographic notes .......................................... 136

7 Hair modeling ...................................................... 137
   7.1 Introduction .................................................. 138
   7.2 Hair volume determination .................................. 139
   7.3 Hair fiber recovery ........................................... 140
      7.3.1 Visibility determination ................................ 140
      7.3.2 Orientation consistency ................................ 141
      7.3.3 Orientation triangulation ................................ 141
   7.4 Implementation ............................................... 142
   7.5 Results and discussions .................................... 144
   7.6 Bibliographic notes .......................................... 148

8 Tree modeling ..................................................... 149
   8.1 Introduction .................................................. 150
   8.2 Branch recovery ............................................... 153
      8.2.1 Reconstruction of visible branches ..................... 153
      8.2.2 Synthesis of occluded branches ........................ 155
      8.2.3 Interactive editing ..................................... 157
   8.3 Leaf extraction and reconstruction ........................ 159
      8.3.1 Leaf texture segmentation ............................ 159
      8.3.2 Graph-based leaf extraction .......................... 162
      8.3.3 Model-based leaf reconstruction ...................... 165
   8.4 Results and discussions .................................... 167
   8.5 Bibliographic notes .......................................... 174

9 Façade modeling .................................................. 177
   9.1 Introduction .................................................. 178
   9.2 Façade initialization ......................................... 180
      9.2.1 Initial flat rectangle ................................... 181
      9.2.2 Texture composition ................................... 181
      9.2.3 Interactive refinement .................................. 183
   9.3 Façade decomposition ....................................... 184
      9.3.1 Hidden structure discovery ........................... 184
      9.3.2 Recursive subdivision .................................. 185
      9.3.3 Repetitive pattern representation .................... 186
9.3.4 Interactive subdivision refinement ............................................. 187
9.4 Façade augmentation ................................................................. 188
  9.4.1 Depth optimization ......................................................... 188
  9.4.2 Cost definition ................................................................. 190
  9.4.3 Interactive depth assignment ............................................ 190
9.5 Façade completion ................................................................. 192
9.6 Results and discussions ......................................................... 192
9.7 Bibliographic notes ............................................................... 197
10 Building modeling ................................................................. 199
  10.1 Introduction ........................................................................... 200
  10.2 Pre-processing ....................................................................... 201
  10.3 Building segmentation ......................................................... 203
    10.3.1 Supervised class recognition ........................................... 203
    10.3.2 Multi-view semantic segmentation .................................... 205
  10.4 Building partition ................................................................. 207
    10.4.1 Global vertical alignment ............................................... 208
    10.4.2 Block separator .............................................................. 208
    10.4.3 Local horizontal alignment ............................................ 209
  10.5 Façade modeling ................................................................. 210
    10.5.1 Inverse orthographic composition ................................... 211
    10.5.2 Structure analysis and regularization .................................. 213
    10.5.3 Repetitive pattern rediscovery ........................................ 216
    10.5.4 Boundary regularization .................................................. 217
  10.6 Post-processing ..................................................................... 218
  10.7 Results and discussions ......................................................... 219
  10.8 Bibliographic notes ............................................................... 224

List of algorithms ................................................................. 227
List of figures ................................................................. 227
References ................................................................. 237
Index ................................................................. 249
Chapter 2
Geometry prerequisite

This chapter revisits some basic concepts from projective geometry, algebraic geometry and computer algebra. For projective geometry, we present, first intuitively then more formally, the concepts of homogeneous coordinates and projective spaces. Then we introduce metric properties using a projective language, paving the way for camera modeling. For algebraic geometry and computer algebra, we primarily target the introduction of the Gröbner basis and eigenvalue methods for solving polynomial systems of vision geometry problems. The purpose is to provide a minimum for readers to be able to follow the book. More refined and detailed treatment of the topics can be found in the excellent textbooks [196] for projective geometry and [27, 26] for algebraic geometry and computer algebra.
2.1 Introduction

The mathematical foundation of 3D geometric modeling is geometry. We are familiar with Euclidean geometry and the methods of Cartesian coordinates integrated into algebra and calculus as part of our standard engineering curriculum. We are not well acquainted with the old subject of projective geometry, nor the new area of computer algebra as they are not part of our curriculum. Both of them, however, play fundamental roles in advancing the geometry of computer vision.

Projective geometry is native in describing camera geometry and central projection. Historically, projective geometry was motivated and developed in search of a mathematical foundation for the techniques of ‘perspective’ used by painters and architects. The success of projective geometry is due above all to the systematic introduction of points at infinity and imaginary points. Computer algebra has been developed for manipulating systems of polynomial equations since the 1960s. Many of the difficult vision geometry problems are now efficiently solved with the concepts and the tools from computer algebra.

2.2 Projective geometry

2.2.1 The basic concepts

Basic geometry concepts

Geometry concepts are built into our mathematical learning. The classical axiomatic approach to geometry proves theorems from a few geometric axioms or postulates. Since Descartes, we usually define a Euclidean coordinate frame; then, points are converted into coordinates and geometric figures are studied by means of algebraic equations. This is the analytical geometry we learned in high school.

Affine and vector space. In modern languages, a vector space is an affine space. For instance, an affine plane is the two-dimensional vector space $\mathbb{R}^2$. A point $P$ on the plane is identified with a vector $\vec{AP} = a\vec{AB} + b\vec{AC}$, where the three non-collinear points $A$, $B$, and $C$ specify a vector basis $\vec{AB}$ and $\vec{AC}$. The affine geometry reduces de facto to linear algebra. Theoretically, there is only a difference of point of view between the concept of a vector space and that of an affine space: each affine space becomes a vector space when we fix one point as the origin; and each vector space is an affine space when we ‘forget’ the ‘zero’. The affine coordinates $(a, b)$ of the point $P$ are the coefficients of linear combinations in the given basis.

Geometrically, the affine coordinates can also be viewed as the relative distances to the $x$-unit length $AB$ and the $y$-unit length $AC$, which are obtained by parallel-projecting to the non-rectangular frame with $A$ as the origin and $AB$ and $AC$ as the $x$ and $y$ axis, as illustrated in Figure 2.1. In affine geometry, there is no distance, no angle, only parallelism and ratios.
2.2 Projective geometry

Fig. 2.1 Left: affine coordinates as coefficients of linear combinations. Right: affine coordinates as ratios.

**Euclidean space.** If we define a scalar product or dot product ‘·’ in the vector space, we have the orthogonality expressed as the vanishing of the scalar product \( \mathbf{x} \cdot \mathbf{y} = 0 \) for two vectors \( \mathbf{x} \) and \( \mathbf{y} \). An affine or vector space equipped with a scalar product is an Euclidean space or a metric space in which we do Euclidean or metric geometry. For instance, the distance between two points \( A \) and \( B \) is the vector length \( (\mathbf{x} \cdot \mathbf{x})^{1/2} \) for \( \mathbf{x} = \overrightarrow{AB} \). The angle between \( AB \) and \( BC \) is the angle \( \arccos(\mathbf{x} \cdot \mathbf{y})/(\mathbf{x} \cdot \mathbf{x})(\mathbf{y} \cdot \mathbf{y}) \) between the two vectors \( \mathbf{x} = \overrightarrow{AB} \) and \( \mathbf{y} = \overrightarrow{BC} \). All these metric concepts are expressed by the scalar product.

**Intuitive projective concepts**

The elements of both affine and Euclidean spaces are geometry points that are all finite points in \( \mathbb{R}^n \), with or without a scalar product.

**Points at infinity.** On a plane, we know that two non-parallel lines intersect at a point, but two parallel lines cannot. Imagine that two parallel lines do meet at a point that is a special point we call a point at infinity for that group of parallel lines. Apparently, different groups of parallel lines meet at a different point at infinity. Simply adding these missing points at infinity to the finite point set of the plane \( \mathbb{R}^2 \) gives an extended plane we call a projective plane \( \mathbb{P}^2 \). More generally, a projective space of any dimension is simply an affine space plus some missing points at infinity:

\[
\mathbb{P}^n = \mathbb{R}^n + \{\text{points at infinity}\}.
\]

In projective geometry, metric properties do not exist, nor does the parallelism, but the collinearity and incidence are kept.

Different geometries speak differently of points at infinity. Euclidean geometry says that parallel lines exist, but never meet. Affine geometry says that parallel lines exist, but meet at a special point. Projective geometry says that any two lines, including parallel lines, always meet at a point. A point at infinity becomes a normal point in projective space. A point at infinity is visible as a finite vanishing point in a photograph of it!

**Homogeneous coordinates.** The difficulty we are facing is how to algebraically represent a point at infinity.
A point $P$ on a real Euclidean line usually has a real number $x \in \mathbb{R}$ as its coordinate, which is the distance to the origin. The point at infinity of the line could not be represented by the symbol $\infty$, which is merely a notation, it is not a number. Instead, we take a pair of real numbers $(x_1, x_2)$ and let the ratio be the usual coordinate such that

$$x = \frac{x_1}{x_2}, \text{ when } x_2 \neq 0.$$ 

The pair of numbers $(x_1, x_2)$ is the homogeneous coordinate of a point, and the number $x$ is the usual inhomogeneous coordinate of the point if it is not at infinity.

Intuitively, when $x_2 \to 0$, $x_1/x_2 \to \infty$. The point represented by homogeneous coordinates $(x_1, 0)$ is the point at infinity. Of course, the point at infinity has only homogeneous coordinates and does not have any inhomogeneous representation as we cannot divide by zero.

We see that the homogeneous representation of a point is not unique since $\lambda(x_1, x_2) \equiv (x_1, x_2)$ for any $\lambda \neq 0$. The point at infinity is $(1, 0)$ as $(1, 0) \equiv x_1(1, 0) \equiv (x_1, 0)$. The point $(0, 1)$ is the origin. But the representation $(0, 0)$ is invalid and does not represent any point. All pairs of such proportional numbers define an equivalent class by $\lambda(x_1, x_2) \equiv (x_1, x_2)$. Each representative of the class is a valid projective point, and the set of all these representatives forms a projective line.

### 2.2.2 Projective spaces and transformations

**Projective spaces**

**Definition.** Let $k$ be a number field, either real numbers $\mathbb{R}$ or complex numbers $\mathbb{C}$. We first define an equivalence relation for non-zero points $x = (x_1, \ldots, x_{n+1})^T \in k^{n+1} - \{0\}$ by setting $x \sim x'$ if there is a non-zero number $\lambda$ such that $x = \lambda x'$. The quotient space of the set of equivalence classes of the relation $\sim$ 

$$\mathbb{P}^n = (k^{n+1} - \{0\})/\sim$$

is a projective space of dimension $n$.

A projective space $\mathbb{P}^n(k)$ or simply $\mathbb{P}^n$ is then the non-zero equivalence classes determined by the relation $\sim$ on $k^{n+1}$. Any element $x = (x_1, \ldots, x_{n+1})^T$ of the equivalence class is called homogeneous coordinates of the point in $\mathbb{P}^n$.

This projective space from the quotient space of homogeneous coordinates is so far only a space that does not inherit algebraic structures from $k^{n+1}$. It is obviously not a vector space; it even does not have the zero! The only structure it inherits is the notion of linear dependence of points that encodes the collinearity! A point $P$
by \( \mathbf{x} \) is said to be linearly dependent on a set of points \( P_i \) by \( \mathbf{x} \), if there exist \( \lambda_i \), such that \( \mathbf{x} = \sum \lambda_i \mathbf{x}_i \).

Essential properties of a projective space follow immediately from the definition.

- A point is represented by homogeneous coordinates.
- The homogeneous coordinates of a given point are not unique, but are all multipliers of one another.
- The \((n + 1)\)-dimensional zero vector \( \mathbf{0} \) does not represent any point in any projective space.
- A line is a set of points linearly dependent upon two distinct points.
- A hyper-plane is a set of linearly dependent points described by \( \mathbf{u}^T \mathbf{x} = 0 \).
- Duality:
  \( \mathbf{u}^T \mathbf{x} = 0 \) can be viewed as \( \mathbf{x}^T \mathbf{u} = 0 \) by transposition.

Geometrically, \( \mathbf{u}^T \mathbf{x} = 0 \) being a set of coplanar points is dual to \( \mathbf{x}^T \mathbf{u} = 0 \) being a set of planes intersecting at the point \( \mathbf{x} \).

- The set of points at infinity in \( \mathbb{P}^n \) is a hyper-plane \( x_{n+1} = 0 \). Finite points are \( (\mathbf{x}_n, 1)^T \), and points at infinity are \( (\mathbf{x}_n, 0)^T \) in \( \mathbb{P}^n \).
- The relation between \( \mathbb{P}^n \) (not \( \mathbb{P}^{n+1} \!\)) and \( \mathbb{P}^m \): all points (finite) in \( \mathbb{P}^n \) are embed-
ded in \( \mathbb{P}^m \) by
  \[
  (x_1, \ldots, x_n)^T \mapsto (x_1, \ldots, x_n, 1)^T.
  \]

And the finite points of \( \mathbb{P}^n \), not at infinity so to exclude \( x_{n+1} = 0 \), are mapped back to \( \mathbb{P}^n \) by
  \[
  (x_1, \ldots, x_{n+1})^T \mapsto (x_1/x_{n+1}, \ldots, x_n/x_{n+1})^T.
  \]

Precisely, the points at infinity are those not reached by this injection.

**Projective transformations**

Among transformations between two projective spaces of same dimension \( \mathbb{P}^n \rightarrow \mathbb{P}^n \), let us take the simplest linear transformation in homogeneous coordinates, which is represented by a \((n + 1) \times (n + 1)\) matrix \( \mathbf{A}_{(n+1) \times (n+1)} \):

\[
\lambda \mathbf{x}_{n+1}' = \mathbf{A}_{(n+1) \times (n+1)} \mathbf{x}_{n+1}.
\]

The crucial fact is that this homogeneous linear transformation maps collinear points into collinear points! This can be verified easily by taking three points, one of which is linearly dependent on the other two, and transforming them with the above matrix.

The collinearity is the very defining property of projective geometry, thus this linear transformation in homogeneous coordinates is called a **projective transformation**, or a **collineation** or a **homography**.

A projective transformation has \((n + 1) \times (n + 1) - 1 = (n + 1)^2 - 1 = 2(n + 2)\) degrees of freedom, which can be determined from \( n + 2 \) points since each point contributes to two inhomogeneous equations. Not surprisingly, this is the
same number of points that define a projective basis in the given space. Obviously, the basis points in two spaces completely determine the transformation.

The set of projective transformations $A_{(n+1) \times (n+1)}$ forms a projective group that is the general linear group of dimension $n + 1$ denoted as $GL(n + 1, k)$.

### The 1D projective line

The 1D projective line is simple.

**Points.** The homogeneous coordinates of a point on a projective line are $x = (x_1, x_2)^T$. There is only one point at infinity $\lambda(1, 0)$ (not two!). Any de-homogenized ratio of $x_1/x_2$ or $x_2/x_1$ admits an invariant interpretation as a cross-ratio of four points or four numbers.

**Cross-ratios.** The cross-ratio of four numbers $a, b, c,$ and $d$ is

$$ (a, b, c, d) = \frac{a - c/b - c}{a - d/b - d}, $$

which is the ratio of the relative distances (and the relative distances are affine coordinates of a point). The cross-ratio is the fundamental projective invariant quantity like the distance for Euclidean geometry and the relative distance for affine geometry. Figure 2.2 shows the invariance of the cross-ratio of four points on a line:

$$ (A, B : C, D) = (A', B' : C', D'). $$

![Fig. 2.2 The invariance of a cross-ratio.](image)

A projective transformation or a homography on a line is given by

$$ \begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} $$

**The topology.** The projective line is closed as there is only one at infinity. It is topologically equivalent to a circle.
The 2D projective plane

The projective plane is the most convenient space to introduce concepts and describe properties.

Points.

- The homogeneous coordinates of a point on a projective plane are \( \mathbf{x} = (x_1, x_2, x_3)^T \).
- The points at infinity are those characterized by \((x_1, x_2, 0)^T\), or \(x_3 = 0\), which is the line at infinity. There is one point at infinity per direction.
- Three points (non-collinear) define a projective basis.

The coordinates of the basis points can be chosen arbitrarily as long as they are independent, the simplest possible ones are called the canonical or standard coordinates of the basis points, which are:

\[
(1, 0, 0), \ (0, 1, 0), \ (0, 0, 1) \text{ and } (1, 1, 1).
\]

Fig. 2.3 Definition of (inhomogeneous) projective coordinates \((\alpha, \beta)\) of a point \(P\) on a plane through four reference points \(A, B, C\) and \(D\).

Lines. Two distinct points are linearly independent. A line is a set of points \(\mathbf{x}\) linearly dependent on two distinct points \(\mathbf{x}_1\) and \(\mathbf{x}_2\), \(\mathbf{x} = \lambda \mathbf{x}_1 + \mu \mathbf{x}_2\), which is equivalent to \(|\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2| = 0\). This leads to the linear form \(\mathbf{l}^T \mathbf{x} = 0\) of the equation of the line in which

\[
\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2.
\]

The cross-product above is merely a notational device from the vanishing determinant. If two points \(\mathbf{x}_1\) and \(\mathbf{x}_2\) define a line by \(\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2\), dually, two lines \(\mathbf{l}_1\) and \(\mathbf{l}_2\) define a point

\[
\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2,
\]

which is the intersection point of the pencil of concurrent lines generated by \(\mathbf{l}_1\) and \(\mathbf{l}_2\):

\[
|\mathbf{l}, \mathbf{l}_1, \mathbf{l}_2| = 0.
\]

Example 1 A first line through two points \((0, 0)^T\) and \((0, 1)^T\) is \((0, 0, 1)^T \times (0, 1, 1)^T = (1, 0, 0)\), i.e., \(x = 0\). A second line through two points \((1, 0)^T\) and
Geometry prerequisite

\((1, 1)^T \times (1, 1, 1)^T = (1, 0, -1)\), i.e., \(x - 1 = 0\). These two lines intersect at the point \((0, 1, 0)^T \times (1, 0, -1)\), which is the point at infinity of the \(y\)-axis. This example illustrates that these basic geometry operations are much easier carried out with homogeneous coordinates than with Cartesian coordinates.

The set of all lines through a fixed point is called a pencil of lines, and the fixed point is the vertex of the pencil.

Conics. A curve described by a second-degree equation in the plane is a conic curve. It could be \(ax^2 + bxy + cy^2 + dx + ey + f t^2 = 0\) in inhomogeneous coordinates or in homogeneous and matrix form

\[x^T C x = 0,\]

where \(C\) is a \(3 \times 3\) homogeneous and symmetric matrix. There are five degrees of freedom for a conic matrix \(C\), so five points determine a unique conic.

The line tangent to a conic \(C\) at a given point \(x\) is \(\lambda l = C x\). The dual conic of a given point conic \(C\) is \(l^T C^{-1} l = 0\), which is a conic in line coordinates, or a line conic or a conic envelope.

Two points are said to be conjugate points with respect to a conic if the cross-ratio of these two points and the two additional points in which the line defined by the two points and the conic meet is in harmonic division or equals to -1.

For any point \(x\) we can associate a polar line \(l = C x\). Accordingly, the point \(x\) is called the pole of the polar line.

A conic is generated by the intersection point of corresponding lines of two homographic pencils of lines, and the vertices of the pencils are on the conic. This is Steiner’s theorem for projective generation of the conic.

The topology. The real projective plane \(\mathbb{P}^2(\mathbb{R})\) is topologically equivalent to the sphere in space with one disc removed and replaced by a Möbius strip. The Möbius strip has after all a single circle as the boundary, and all that we are asking is that the points of this boundary circle be identified with those of the boundary circle of the hole in the sphere. The resulting closed surface is not orientable, and is equivalent to the projective plane.

The 3D projective space

The 3D projective space can be extended from the 2D projective plane with more elaborate structures.

Points. The homogeneous coordinates of a point in a 3D projective space are \((x_1, x_2, x_3, x_4)\). The points at infinity \((x_1, x_2, x_3, 0)\) or \(x_4 = 0\) form a plane, the plane at infinity. There is a line at infinity for each pencil of parallel planes.

Planes. Three distinct points are linearly independent. A plane is a set of points \(x\) linearly dependent on three distinct points \(x_1, x_2\) and \(x_3\): \(x = \alpha x_1 + \beta x_2 + \gamma x_3\), which is

\[|x, x_1, x_2, x_3| = 0,\]
which is the linear form $u^T x = 0$ of the equation of the plane. The points and planes (not lines!) are dual in space. For instance, three points define a plane and three planes intersects at a point.

The set of all lines through a fixed line is called a pencil of planes, and the fixed line is the axis of the pencil.

**Lines.** A line is the set of points $x = a x_1 + b x_2$ linearly dependent on two distinct points $x_1$ and $x_2$. A space line has only 4 degrees of freedom. Count it properly! One way is to imagine that a line is defined through two points each moving on a fixed plane, and a point moving on a pre-fixed plane has only 2 degrees of freedom. The Grassmanian or Plücker coordinates of a line in space is to take the six $2 \times 2$ minors $p_{ij} = |ij|$ of the $2 \times 4$ matrix by stacking $x_1$ and $x_2$. The vanishing $\det(x, y, x, y) = 0$ gives the quadratic identity

$$\Omega_{pp} = p_{01} p_{23} + p_{02} p_{31} + p_{03} p_{12} = 0.$$ 

These minors are not independent. Among the six of them, there are only 6 - 1 (scale) - 1 (identity) = 4 degrees of freedom. Two lines $p$ and $q$ are coplanar or intersect if and only if $\Omega_{pq} = 0$.

**Quadrics.** The first analogue of the conic as a surface in 3D space is a quadric surface. Analogous to a conic $x^T C x = 0$, a surface of degree two is

$$x^T Q x = 0,$$

where $Q$ is a $4 \times 4$ homogeneous and symmetric matrix. A quadric has nine degrees of freedom, so nine points in general positions determine a quadric.

The dual quadric in plane coordinates is a plane quadric

$$u^T Q^{-1} u = 0.$$ 

A surface is ruled if through every point of the surface there is a straight line lying on the surface. A proper non-degenerate ruled quadric is the hyperboloid of one sheet. The degenerate rules quadrics might be cones and planes. Ruled surfaces are of particular importance for the study of critical configurations of vision algorithms.

A ruled surface is generated by the intersection line of corresponding planes of two homographic pencils of planes.

**Twisted cubics.** The second analogue of the conic as a curve in 3D space is a twisted cubic, which is represented through a non-singular $4 \times 4$ matrix $A$ and parameterized by $\theta$ as

$$(x_1, x_2, x_3, x_4)^T = A(\theta^3, \theta^2, \theta, 1)^T.$$ 

This is analogous to a conic $(x_1, x_2, x_3)^T = A(\theta^2, \theta, 1)^T$, with a non-singular $3 \times 3$ matrix $A$. A twisted cubic has twelve degrees of freedom, the sixteen entries of the
matrix minus one for the homogeneity and minus three for a 1D homography of the parameter $\theta$. Six points in general positions define a unique twisted cubic.

**The topology.** The real projective space $\mathbb{P}^3(\mathbb{R})$ is topologically equivalent to a sphere in four-dimensional space with the antipodal points identified, which is the topology of the rotation group $SO(3)$. It is orientable.

### 2.2.3 Affine and Euclidean specialization

We have seen how a projective space is extended from an affine space by adding the special points at infinity, which lose the special status in projective space. Now we will see how a projective space is specialized into an affine and a Euclidian space by singling out special points and lines, which will then enjoy the special status of invariance in the specialized spaces.

This is Klein’s view to regard projective geometry as an umbrella geometry under which affine, similarity, and Euclidean geometries all reside as sub-geometries of projective geometry. A geometry is associated with a group of transformations that leaves the properties of the geometry invariant.

**Projective to affine**

Affine geometry involves a specialization of one invariant line in 2D and one invariant plane in 3D. Affine specialization is linear.

**The line at infinity.** The line at infinity defines what is affine in geometry. The parallel lines meet at a point at infinity. The projective conic specializes as an ellipse, parabola, and hyperbola in the affine space depending on the number of the intersection points of a conic with the line at infinity.

![Fig. 2.4 Affine classification of a conic: an ellipse, a parabola, and a hyperbola is a conic intersecting the line at infinity at 0 point, 1 (double) point, and 2 points.](image)

**The 2D affine transformation.** Affine geometry is characterized by the group of transformations that leaves the points at infinity invariant. The invariance is global, not point by point, that is, a point at infinity is usually mapped into a different point at infinity, but it cannot be mapped into a finite point. Class matters!

Start from an arbitrary projective transformation
2.2 Projective geometry

\[ A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}. \]

Choose a line at infinity. For convenience, we take \( x_3 = 0 \). If it is left invariant by any transforms, then the transformed line should be \( x_3' = x_3 = 0 \). This constraint imposes \( a_{31} = a_{32} = a_{33} = 0 \), so an affine transformation is of the form:

\[ A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}. \]

It is straightforward to verify by inspection that all such matrices form a group of affine transformations, which is a sub-group of the group of projective transformations.

Now for all finite points \( x_3 \neq 0 \), we can de-homogenize the homogeneous coordinates \((x_1, x_2, x_3)^T\) by \( x = x_1/x_3 \) and \( y = x_2/x_3 \). Then, we obtain

\[ \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix}, \]

which appears in the familiar form of the affine transformation in inhomogeneous coordinates, which is a linear transformation plus a translation, valid only for finite points.

**The plane at infinity.** Each plane has one line at infinity. All lines at infinity from all planes form a plane at infinity in space.

**The 3D affine transformation.** A 3D affine transformation leaving invariant the plane at infinity \( x_4 = 0 \) becomes

\[ \begin{pmatrix} A_{3 \times 3} & a_{1 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix}. \]

**Affine to Euclidean**

Euclidean geometry involves a specialization of an invariant pair of points in 2D and an invariant conic in 3D. Euclidean specialization is quadratic.

**The circular points.** The circular points define what is Euclidean from projective and affine point of view. What are these points? Let’s first do the impossible of intersecting two concentric circles of different radii:

\[ x^2 + y^2 = 1, \]
\[ x^2 + y^2 = 4. \]

There is no intersection point if we stay in finite space characterized by the above inhomogeneous coordinates and equations. We first homogenize the equations through \( x \mapsto x_1/x_3 \) and \( y \mapsto x_2/x_3 \) to extend Euclidean space to projective space,
\[ x_1^2 + x_2^2 = x_3^2, \]
\[ x_1^2 + x_2^2 = 4x_3^2. \]

The subtraction of the two equations leads to \(3x_3^2 = 0\), so \(x_3 = 0\) and \(x_1^2 + x_2^2 = 0\), which gives \((x_1/x_2)^2 = -1\), or \(x_1/x_2 = \pm i\). Thus, the intersection points of the two concentric circles are a pair of points \((\pm i, 1, 0)\), which we call circular points and denote them by \(I\) and \(J\) with coordinates \(i\) and \(j\). They are called circular points as all circles always go through them! This can be easily verified by substituting the circular points back into an equation of any circle. We never see them in Euclidean space as they are complex points at infinity! A circle is determined by three points, as the other two points are the circular points.

**The orthogonality.** The circular points are an algebraic device to specify the metric properties of a Euclidean plane. The orthogonality of two lines is expressed as that the cross-ratio of the intersection points of the two lines with the line at infinity, and the circular points is in harmonic division:

\[(A, B; I, J) = -1.\]

Furthermore, the angle between the two lines can be measured through the cross-ratio by Laguerre’s formula,

\[\theta = \frac{i}{2} \ln(A, B; I, J).\]

**The congruency.** The congruency substitutes the homography as a one-dimensional Euclidean transformation. The two pencils are congruent if they are related by a rotation, so the angle between the corresponding points (or lines or planes) is constant.

For instance, projective generation of the conic and the quadric from homographic pencils of lines and planes becomes a kind of Euclidean generation of the circle and the orthogonal quadric from congruent pencils of lines and planes. In particular, a circle is generated by two congruent pencils of lines specified by the three points.

**The 2D Euclidean transformation.** (Similarity) Euclidean geometry is obtained by the group of transformations that leaves the circular points invariant, as a pair, so \((i, 1)\) might be mapped to \((-i, 1)\). The invariance

\[
\begin{pmatrix}
\pm i \\
1
\end{pmatrix}
= \mathbf{A}_{2 \times 2}
\begin{pmatrix}
\pm i \\
1
\end{pmatrix}
\]

constrains \(s^2 + c^2 = 1\) such that

\[
\begin{pmatrix}
x' \\
y'
\end{pmatrix}
= \rho
\begin{pmatrix}
c & s \\
-s & c
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
+
\begin{pmatrix}
a_{13} \\
a_{23}
\end{pmatrix},
\]

which is the familiar similarity transformation if we make the parameterization of \(c\) and \(s\) with an explicit angle \(\theta\) as \(c = \cos \theta\) and \(s = \sin \theta\). The \(\rho\) is a global scaling factor.
2.2 Projective geometry

The absolute conic. The pair of circular points \( i \) and \( j \), also called the absolute points, are in fact jointly described by the equation \( x_1^2 + x_2^2 = x_3 = 0 \). The absolute conic is its extension in 3D,

\[
x_3^T x_3 \equiv x_1^2 + x_2^2 + x_3^2 = 0,
\]

which is made up of all circular points of all planes. The absolute conic has no real points; it is purely imaginary. It is the algebraic device to specify the metric properties of the Euclidean space. Two planes are perpendicular if they meet the absolute conic in a pair of lines conjugate for the absolute conic. Two lines are perpendicular if they meet the absolute conic in a pair of points conjugate for the absolute conic.

The 3D Euclidean transformation. A similarity transformation leaves invariant, globally, the absolute conic. The invariance of the absolute conic,

\[
x_3^T A_{3 \times 3} x_3 = x_3^T A_{3 \times 3} A_{3 \times 3} x_3 = x_3^T x_3 \quad \text{with} \quad x_3' = A_{3 \times 3} x_3,
\]

constrains \( A_{3 \times 3} A_{3 \times 3} = I_{3 \times 3} \), which is the defining property for \( A_{3 \times 3} \) to be a rotation matrix \( R \)! Recall that a \( 3 \times 3 \) matrix \( R \) is an orthogonal matrix representing a 3D rotation if \( RR^T = I_{3 \times 3} \). It has three degrees of freedom that might be chosen for the three rotation angles around the three axes.

The similarity transformation now becomes

\[
\begin{pmatrix}
R_{3 \times 3} a_{1 \times 1} \\
0_{1 \times 3} \rho
\end{pmatrix}.
\]

If we fix the global scale \( \rho = 1 \), we obtain the familiar rigid transformation in homogeneous form:

\[
\begin{pmatrix}
x' \\
y' \\
z'
\end{pmatrix} =
\begin{pmatrix}
R_{3 \times 3} \\
t_{3 \times 1}
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
\]

or in usual inhomogeneous form

\[
\begin{pmatrix}
x' \\
y' \\
z'
\end{pmatrix} = R \begin{pmatrix}
x \\
y \\
z
\end{pmatrix} + t.
\]

The geometry of cameras

The purpose of reviewing the projective geometry is that the camera geometry is more intrinsically described by it. We will see in the coming chapter the following highlights.
• A pinhole camera without nonlinear optical distortions is a projective trans-formation from a projective space of dimension three $\mathbb{P}^3$ to a projective space of dimension two $\mathbb{P}^2$, which is a $3 \times 4$ matrix.

• The intrinsic parameters of the camera, describing the metric properties of the camera, are corresponding to the absolute conic, which specializes a projective geometry into a Euclidean geometry. If a camera is calibrated, then the absolute conic is fixed.

• The epipolar geometry of two cameras is that of two pencils of planes, which are in homographic relation if the cameras are not calibrated, and are in congruent relation if the cameras are calibrated.

Summary

The hierarchy of geometries in projective language is summarized in Table 2.1 and 2.2 for 2D planes and 3D spaces. The term ‘incidence’ should broadly include collinearity and tangency.

Notice that invariant elements of a given transformation are different from invariant elements of a given group of transformations. The invariants of a group specifies different geometries, while the invariants of a transformation merely characterize the type of transformation within the same group or geometry. For instance, the line at infinity is invariant for the affine group, while an eigenvector for a given transformation representing a point is a fixed point for the given transformation.

Table 2.1 Hierarchy of plane geometries.

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Projective</th>
<th>Affine</th>
<th>Similarity</th>
<th>Euclidean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation group</td>
<td>$\begin{pmatrix} a_{11} &amp; a_{12} &amp; a_{13} \ a_{21} &amp; a_{22} &amp; a_{23} \ a_{31} &amp; a_{32} &amp; a_{33} \end{pmatrix}$</td>
<td>$\begin{pmatrix} a_{11} &amp; a_{12} &amp; a_{13} \ a_{21} &amp; a_{22} &amp; a_{23} \ 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} \cos \theta &amp; \sin \theta &amp; t_1 \ -\sin \theta &amp; \cos \theta &amp; t_2 \ 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} \cos \theta &amp; \sin \theta &amp; t_1 \ -\sin \theta &amp; \cos \theta &amp; t_2 \ 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
</tr>
<tr>
<td>D.o.f.</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Invariant properties</td>
<td>incidence (including collinearity and tangency)</td>
<td>incidence, parallelism</td>
<td>incidence, parallelism, orthogonality</td>
<td>incidence, parallelism, orthogonality</td>
</tr>
<tr>
<td>Invariant quantities</td>
<td>cross-ratio</td>
<td>ratio</td>
<td>ratio, angle</td>
<td>distance, angle</td>
</tr>
<tr>
<td>Invariant elements</td>
<td>—</td>
<td>line at infinity</td>
<td>line at infinity, circular points</td>
<td>line at infinity, circular points</td>
</tr>
<tr>
<td>Deformation of a unit square</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.3 Algebraic geometry

Many geometry problems lead to a system of polynomial equations of the form:

\[ f_1(x_1, \ldots, x_n) = 0, \]
\[ \ldots \]
\[ f_s(x_1, \ldots, x_n) = 0, \]

where \( f_i \) are polynomials in \( n \) variables with coefficients from the real number field. The goal is to solve this system of polynomial equations.

2.3.1 The simple methods

The fundamental theorem of algebra states that a polynomial equation of degree \( n \) in one variable

\[ f(x) = x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0 \]

has \( n \) solutions over the field \( \mathbb{C} \) of complex numbers.

- **The companion matrix.** Solving a polynomial in one variable \( f(x) = 0 \) is equivalent to computing the eigenvalues of its companion matrix

\[
\begin{pmatrix}
0 & 0 & \ldots & 0 & -c_0 \\
1 & 0 & \ldots & 0 & -c_1 \\
0 & 1 & \ldots & 0 & -c_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & -c_n \\
\end{pmatrix}
\]

Numerically, the power method for eigenvalue or the Newton-Raphson for finding all the roots of a polynomial in one variable could be used for effective solutions [172].
• **Bézout’s theorem.** For a polynomial system, if the degrees of the polynomials are \( n_1, n_2, \ldots, n_m \), then there are \( n_1 \times n_2 \times \ldots \times n_m \) solutions. It is exact if we count properly with complex points, points at infinity, and multiplicities of points.

• **The Sylvester resultant.** A system of polynomials could be solved by elimination to reduce the system to a polynomial in only one variable, similar to the approach for a system of linear equations. We mention two ways of eliminating variables. The first elimination method is the classical Sylvester resultant of two polynomials. The second method requires a few more advanced algebraic geometry concepts based on the Gröbner bases, which will be presented in the next paragraph.

Given two polynomials \( f, g \in k[x_1, \ldots, x_n] \),

\[
\begin{align*}
  f &= a_0x^l + \ldots + a_l, \\
  g &= b_0x^m + \ldots + b_m,
\end{align*}
\]

then the Sylvester matrix of \( f \) and \( g \) with respect to \( x \), denoted \( \text{Sylvester}(f, g, x) \), is

\[
\begin{pmatrix}
  a_0 & b_0 \\
  a_1 & b_1 & b_0 \\
  a_2 & a_1 & \ddots & \ddots & b_1 \\
  \vdots & \vdots & \ddots & \ddots & \ddots \\
  a_l & \ddots & \ddots & a_1 & b_m \\
  \vdots & \ddots & \ddots & \ddots & \ddots \\
  a_l & \ddots & \ddots & \ddots & \ddots & \vdots \\
  \vdots & \ddots & \ddots & \ddots & \ddots & a_l \\
  \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & b_m
\end{pmatrix}
\]

The resultant of \( f \) and \( g \) with respect to \( x \) is the determinant of the Sylvester matrix, i.e.,

\[
\text{Resultant}(f, g, x) = \text{Det}(\text{Sylvester}(f, g, x)),
\]

which is free of the variable \( x \) so that the variable \( x \) is ‘eliminated’. In Maple, we can use the command ‘resultant()’ to compute the resultant for instance.

**Example 2** Let \( f = xy - 1 \) and \( g = x^2 + y^2 - 4 \).

\[
\text{Resultant}(f, g, x) = \begin{vmatrix}
  y & 0 & 1 \\
  -1 & y & 0 \\
  0 & -1 & y^2 - 4
\end{vmatrix} = y^4 - 4y^2 + 1,
\]

which eliminates the variable \( x \).
2.3 algebraic geometry

2.3.2 Ideals, varieties, and Gröbner bases

A monomial is a finite product of variables \( x_1^{\alpha_1} \cdots x_n^{\alpha_n} \) with non-negative exponents \( \alpha_1, \ldots, \alpha_n \). A polynomial \( f \) in \( x_1, \ldots, x_n \) with coefficients in a number field \( k \) (either \( \mathbb{R} \) or \( \mathbb{C} \)) is a finite linear combination of monomials. The sum and product of two polynomials are again a polynomial. The set of all polynomials in \( x_1, \ldots, x_n \) with coefficients in \( k \), denoted by \( k[x_1, \ldots, x_n] \), is a ring, or a polynomial ring, which is also an infinite-dimensional vector space.

**Definition.** An ideal \( I \) is a subset of the ring \( k[x_1, \ldots, x_n] \) satisfying

1. \( 0 \in I \).
2. If \( f, g \in I \), then \( f + g \in I \).
3. If \( f \in I \) and \( h \in k[x_1, \ldots, x_n] \), then \( hf \in I \).

If \( f_1, \ldots, f_s \in k[x_1, \ldots, x_n] \), then \( I = \langle f_1, \ldots, f_s \rangle \) is an ideal of \( k[x_1, \ldots, x_n] \), generated by \( f_1, \ldots, f_s \).

The ideal \( I \) generated by \( f_i = 0 \) consists of all polynomial consequences of the equations \( f_i = 0 \). Intuitively, the solutions of the ideal is the same as those of the generating polynomials.

**Definition.** The variety \( V(f_1, \ldots, f_s) \in k^n \) is the set of all solutions of the system of equations \( f_1 = f_2 = \ldots = f_s = 0 \),

\[
V = \left\{ (a_1, \ldots, a_n) \in k^n : f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq s \right\}.
\]

A variety is determined by the ideal, but not by the equations that can be rearranged. A variety can be an affine variety in \( k^n \) defined by inhomogeneous polynomials or a projective variety in \( \mathbb{P}^n \) defined by homogeneous polynomials. In algebraic geometry terminology, finding the finite number of solutions to polynomial equations \( f_i = 0 \) is equivalent to finding the points of the variety \( V(I) \) for an ideal \( I \) generated by \( f_i \).

The generating polynomials \( f_1, \ldots, f_s \) for an ideal \( I \) is a basis of \( I \). The Hilbert basis theorem states that every ideal is finitely generated. But there are many different bases for an ideal. A Gröbner basis or standard basis is an ideal basis that has special interesting properties.

If we want to solve the ideal membership of a polynomial \( f \in k[x_1, \ldots, x_n] \) regarding an ideal generated by a set of polynomials \( F = \{ f_1, \ldots, f_s \} \), then intuitively we may divide \( f \) by \( F \). This means that \( f \) is expressed in the form:

\[
f = a_1 f_1 + \ldots + a_s f_s + r,
\]

where the quotients \( a_1, \ldots, a_s \) and the remainder \( r \) are in \( k[x_1, \ldots, x_n] \). If \( f \) is ‘divisible’, i.e., the remainder is zero, then \( f \) belongs to the ideal generated by \( F \).

But to characterize the remainder properly, it is necessary to introduce an ordering for the monomials. There are many choices of monomial orders.

**Definition.** We first fix an ordering on the variables, \( x_1 > x_2 > \ldots > x_n \). The lexicographic order is analogous to the ordering of words used in dictionaries, that
Geometry prerequisite

is $x^\alpha >_{\text{lex}} x^\beta$ if the left-most nonzero entry in the difference $\alpha - \beta$ is positive. The graded lexicographic order is first ordered by the total degree, then breaks ties using the lexicographic order. The graded reverse lexicographic order is first ordered by the total degree, then breaks ties in the way that $x^\alpha >_{\text{grevlex}} x^\beta$ if $\sum_i \alpha_i > \sum_i \beta_i$, or if $\sum_i \alpha_i = \sum_i \beta_i$, and the right-most nonzero entry in the difference $\alpha - \beta$ is negative. The graded reverse lexicographic order is less intuitive, but is more efficient for computation.

**Example 3** If $x > y > z$, we have the following ordering of the given monomials.

Lexicographic: $x^3 > x^2 z^2 > xy^2 z > z^2$,

Graded lexicographic: $x^2 z^2 > xy^2 z > x^3 > z^2$,

Graded reverse lexicographic: $xy^2 z > x^2 z^2 > x^3 > z^2$.

**Definition.** Given a monomial order, a finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal $I$ is a Gröbner basis if

$$\langle \text{LT}(g_1), \ldots, \text{LT}(g_t) \rangle = \langle \text{LT}(I) \rangle,$$

where $\text{LT}(f)$ is the leading term of a polynomial $f$. That is, a set of polynomials is a Gröbner basis if and only if the leading term of any element of $I$ is divisible by one of the $\text{LT}(g_i)$.

If $G$ is a Gröbner basis of $I = \langle f_1, \ldots, f_s \rangle$, the remainder $f^G$ of a polynomial $f$ is uniquely determined when we divide by $G$, and

$$f \in I \text{ if and only if } f^G = 0.$$

### 2.3.3 Solving polynomial equations with Gröbner bases

There are two approaches to solving polynomial equations via Gröbner bases. The first approach uses a lexicographic Gröbner basis to obtain a polynomial in one variable. The second uses a non-lexicographic Gröbner basis to convert the system into an eigenvalue problem.

**By elimination**

A lexicographic Gröbner basis for a polynomial system is to Gaussian elimination for linear system. One of the generators is a polynomial in one variable only. So once we have a lexicographic Gröbner basis, we solve it by using any of the numerical root-finding methods for one-variable polynomials, then back-substituting it to find solutions of the whole system. This is conceptually a simple and universal method that generalizes the usual techniques used to solve systems of linear equations.

**Example 4** For the system of equations
2.3 Algebraic geometry

\begin{align*}
  x^2 + y^2 + z^2 &= 4, \\
  x^2 + 2y^2 &= 5, \\
  xz &= 1,
\end{align*}

a lexicographic Gröbner basis for the system (for instance, use the command ‘gbasis’ in Maple to compute) is

\begin{align*}
  x + 2z^2 - 3z &= 0, \\
  y^2 - z^2 - 1 &= 0, \\
  2z^4 - 3z^2 + 1 &= 0.
\end{align*}

The last equation has only one variable $z$.

The main challenge is that the computation of a Gröbner basis with symbolic coefficients in the system of polynomials is difficult. A non-lexicographic basis is often easier to compute than a lexicographic one. This motivates the following eigenvalue methods using a non-lexicographic Gröbner basis.

**Via eigenvalues**

This approach exploits the structure of the polynomial ring $k[x_1, \ldots, x_n]$ modulo an ideal $I$ from a non-lexicographic Gröbner basis $G$.

**Definition.** The quotient $k[x_1, \ldots, x_n]/I$ is the set of equivalent classes:

$$k[x_1, \ldots, x_n]/I = \{ [f] : f \in k[x_1, \ldots, x_n] \},$$

and the equivalent class of $f$ modulo $I$ or the coset of $f$ is

$$[f] = f + I = \{ f + h : h \in I \}.$$

The key is that $[f] = [g]$ implies $f - g \in I$.

The quotient $k[x_1, \ldots, x_n]/I$ is a ring as we can add and multiply any of its two elements. Moreover, we can multiply by constants, so it is a vector space over $k$ as well. This vector space is denoted by $\mathbb{A} = k[x_1, \ldots, x_n]/I$, which is finite-dimensional while the polynomial ring $k[x_1, \ldots, x_n]$ as a vector space is infinite-dimensional. Given a Gröbner basis $G$ of $I$, if we divide $f$ by $G$, then

$$f = h_1g_1 + \ldots + h_tg_t + \overline{f^G}.$$

The remainder $\overline{f^G}$ can be taken as a standard representative of $[f]$. There is a one-to-one correspondence between remainders and equivalent classes of the quotient ring. Furthermore, the remainder $\overline{f^G}$ is a linear combination of the monomials $x^\alpha \notin \langle \text{LT}(I) \rangle$ in the vector space. This set of monomials

$$B = \{ x^\alpha : x^\alpha \notin \langle \text{LT}(I) \rangle \},$$
which are linearly independent, can therefore be regarded as a basis of the vector space \( \mathbb{A} \).

The crucial fact is that when a system has only a finite number of \( n \) solutions, the vector space \( \mathbb{A} = \mathbb{C}[x_1, \ldots, x_n]/I \) has dimension \( n \).

**Definition.** Given a polynomial \( f \in \mathbb{C}[x_1, \ldots, x_n] \), we can use the multiplication of a polynomial \( g \in \mathbb{A} \) by \( f \) to define a map from the vector space \( \mathbb{A} \) to itself,

\[
[g] \rightarrow [g] \cdot [f] = [g \cdot f] \in \mathbb{A}.
\]

This mapping is linear and can be represented by an \( n \times n \) matrix \( A_f \) in the monomial basis \( B \) if \( n = \text{Dim}(\mathbb{A}) \):

\[ A_f : \mathbb{A} \rightarrow \mathbb{A}. \]

This matrix can be constructed from a Gröbner basis. Given a Gröbner basis \( G \) of \( I \). We first obtain the monomial basis \( B \) of \( \mathbb{A} \) as the non-leading terms of \( G \).

Then we multiply each basis monomial of \( B \) by \( f \), and compute the remainder of the product modulo \( G \) (we can use the command ‘normalf()’ in Maple for instance). The vector of coefficients in the basis \( B \) of the remainder is a column of the matrix \( A_f \).

**Example 5** The set

\[
G = \{ x^2 + 3xy/2 + y^2/2 - 3x/2 - 3y/2, xy^2 - x, y^3 - y \}
\]

is a Gröbner basis in graded reverse lexicographic order for \( x > y \). Viewed as a vector space, it can be rearranged into matrix form:

\[
Gx = \begin{bmatrix}
1 & 0 & 0 & 1/2 & 3/2 & -3/2 & -3/2 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x^2 \\
xy^2 \\
y^3 \\
y^2 \\
xy \\
y \\
x \\
1
\end{bmatrix}.
\]

(2.1)

The leading monomials are \( \{ x^2, xy^2, y^3 \} \). The non-leading monomials are \( B = \{ y^2, xy, y, x, 1 \} \), which is a basis for \( \mathbb{A} \). We compute the product of each monomial in \( B \) and \( x \) modulo \( G \),

\begin{align*}
y^2 \mapsto y^2 \cdot x, & \quad y^2 \cdot x^G = x = (0, 0, 0, 1, 0)b, \\
xy \mapsto xy \cdot x, & \quad xy \cdot x^G = \frac{3}{2}y^2 + \frac{3}{2}xy - \frac{1}{2}y - \frac{3}{2}x = \left( \frac{3}{2}, \frac{3}{2}, -\frac{3}{2}, -\frac{3}{2}, 0 \right)b, \\
y \mapsto y \cdot x, & \quad y \cdot x^G = xy = (0, 1, 0, 0, 0)b, \\
x \mapsto x \cdot x, & \quad x \cdot x^G = -\frac{3}{2}y^2 - \frac{3}{2}xy + \frac{3}{2}y + \frac{3}{2}x = \left( -\frac{3}{2}, -\frac{3}{2}, \frac{3}{2}, \frac{3}{2}, 0 \right)b, \\
1 \mapsto 1 \cdot x, & \quad 1 \cdot x^G = x = (0, 0, 0, 1, 0)b,
\end{align*}
where \( \mathbf{b} = (y^2, xy, y, x, 1)^T \). Each vector of coefficients is a column of the matrix \( \mathbf{A}_x \) for the multiplication operator by \( x \), thus we obtain

\[
\mathbf{A}_x = \begin{pmatrix}
0 & 3/2 & 0 & -1/2 & 0 \\
0 & 3/2 & 1 & -3/2 & 0 \\
0 & -1/2 & 0 & 3/2 & 0 \\
1 & -3/2 & 0 & 3/2 & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}.
\]

Notice that some entries of \( \mathbf{A}_x \) have already appeared in the 3 \times 5 sub-matrix of the matrix \( \mathbf{G} \) in Eq. 2.1. In practice, \( \mathbf{A}_x \) is constructed by rearranging \( \mathbf{G} \).

The conclusion is that the eigenvalues of \( \mathbf{A}_f \) are the values of \( f \) evaluated at the solution points of the variety \( V(I) \). Moreover, the right eigenvectors of \( \mathbf{A}_f \) or the eigenvectors of the transpose \( \mathbf{A}_f^T \) are the monomial basis evaluated at the solution points. The solution points are trivially included or reconstructed from the right eigenvectors. If \( f = x_i \), the eigenvalues of \( \mathbf{A}_{x_i} \) are the \( x_i \)-coordinates of the solution points of \( V(I) \). The choice of \( f \) should guarantee that \( f \) is evaluated distinctly on different solution points. It often suffices to take \( f = x_1 \) or a linear combination of variables if necessary. This matrix is the extension of the companion matrix for a polynomial of a single variable to a system of polynomials. More details and proofs could be found in [26, 27].
Chapter 3
Multi-view geometry

This chapter studies the geometry of a single view, two views, three views and $N$ views. We will distinguish an uncalibrated setting from a calibrated one if necessary. The focus is on finding the algebraic solutions to the geometry problems with the minimal data. The entire multi-view geometry is condensed into the three fundamental algorithms: the five-point algorithm for two calibrated views, the six-point algorithm for three uncalibrated views and the seven-point algorithm for two uncalibrated views. Adding calibration and auto-calibration, this chapter is summarized into seven algorithms, which are the computational engine for the geometry of multiple views in ‘structure from motion’ of the next chapter.
3.1 Introduction

A camera projects a 3D point in space to a 2D image point. A ‘good’ camera projects a straight line to a straight line. The geometry of a single camera specifies this 2D and 3D relationship while preserving the straightness of lines. We can ‘triangulate’ the 3D point in space if it is observed in at least two views much like do our two eyes! The geometry of two or more views studies the triangulation of a space point from multiple views. Intuitively, there are fundamentally two geometric constraints known for long time. The first is the collinearity constraint of a space point, its projection and the camera center for one view. The second is the coplanarity constraint of a space point, its two projections, and the camera centers for two views.

These properties have often been presented in a Euclidean setting associated with a calibrated camera framework. But the properties of collinearity and coplanarity are fundamental incidence properties that do not require any metric measurements. They are better expressed in a projective geometry setting associated with an uncalibrated camera framework. That has been the motivated development of 3D computer vision in the 1990s [47, 80], and it has been enormously successful. With hindsight, the projective structure per se is not the goal as it might have been suggested then, but it is an intrinsic representation to encode the crucial correspondence information across images, which does not require any metric information of the camera as long as the camera is pin-hole and the scene is rigid and static. A projective setting simplifies parameterization by justifying more of them, but most of the algorithms suffer from degeneracy in the presence of the coplanarity structure while the calibrated versions do not. It is wrong not to use the partially known parameters of the camera to come back to a Euclidean setting [88], if the information is available. Calibrated setting using Euclidean geometry or uncalibrated one using projective geometry is not an ideological combat, but is more of computational convenience encountered at different stages of the entire reconstruction pipeline, and each complements the other. The literature on the subject is abundant. In the end, only the seven-point algorithm and the six-point algorithm for two and three uncalibrated views and the five-point algorithm for two calibrated views are the known algorithms sufficient to characterize the whole vision geometry. The classical calibration and pose estimation still play an important role in the calibrated setting, the modern auto-calibration is more conceptual than practical. The goal is to synthesize and summarize the vision geometry in seven algorithms for computational purposes.

3.2 The single-view geometry

3.2.1 What is a camera?

If cameras are free of nonlinear optical distortions, straight lines are projected onto straight lines. This is the ideal pinhole camera model. We first develop the model
in a Euclidean setting, then in a projective setting. Both of them lead to the same conclusion.

A Euclidean setting

**Camera coordinate frame.** We first specify a Cartesian coordinate frame $x_cy_cz_c = o_c$, centered at the projection center and aligned with the optical axis of a pinhole camera, as illustrated in Figure 3.1. The image plane or the retina is located at the distance $f$ away from the projection center. This coordinate frame is called the **camera-centered coordinate frame**.

The pinhole camera is mathematically a central projection, which is

$$\frac{x}{f} = \frac{x_c}{z_c}, \text{ and } \frac{y}{f} = \frac{y_c}{z_c},$$

by similar triangles.

![Fig. 3.1 The central projection in the camera-centered coordinate frame.](image)

It is only the play of the matrix for the above central projection to become

$$\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}.$$ 

The remaining efforts are only a matter of making the appropriate coordinate changes to the observed pixels, expressed in a different 2D image coordinate frame; and to the object space, measured in a different 3D coordinate frame.

**Image coordinate frame.** The image plane $xy = p$ is a 2D plane involving only $x_c$ and $y_c$, and $p$. The point $p$ is the intersection point $(0, 0, f)$ of the optical axis $z_c$ with the image plane at $f$. It is called the **principal point**. Since image pixels are observed and measured within the image plane from a corner, which is a 2D
coordinate frame specified by $uv - o$. The pixel space $uv$ may not be orthogonal, so

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_u & a & u_0 \\ b & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$ 

The point $p = (u_0, v_0)$ specifies the offset between the two frames in pixels (the $uv$ units). The $\alpha_u$ and $\alpha_v$ convert the focal length in $x$ and $y$ units of $mm$ for instance to $u$ and $v$ units of pixels in the horizontal and vertical directions that might be different. The ratio of the difference in size in the two directions $\alpha_u/\alpha_v$ is called the aspect ratio. The possible non-perpendicularity of the $uv$ axes is the skew, accounting for the parameter $a$. The parameter $b$ may have accounted for a rotation between the two frames aligning $u$ and $x$, but we will see that this rotation is not independent and could be absorbed later in an external rotation of the 3D frame $xyz - c$. This finalizes the most general form of the upper triangular matrix of the five intrinsic parameters:

$$K = \begin{pmatrix} \alpha_u & a & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$ 

There is no reason for the pixel in $uv$ to not be square! This is true for most modern CCD cameras. Thus, there is no skew so that $a = 0$, and no aspect ratio, so that $\alpha_u = \alpha_v = f$. The matrix of the intrinsic parameters reduces to the three-parameter form

$$K = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$ 

In camera manufacturing, the best effort is made to align and to center the optical axis with the retina sensor. It is reasonable to assume that the $(u_0, v_0)$ is at the center of the image plane for non-metric applications or as a good initial guess of the true position. For metric applications with metric cameras in photogrammetry, all five intrinsic parameters need to be carefully readjusted through the calibration process, as do the non-linear distortion parameters.
3.2 The single-view geometry

World coordinate frame. A 3D point or an object is first measured in its own 3D Euclidian coordinate frame, which is called the world coordinate frame \( w \). The transformation between the world coordinate frame and the camera-centered frame is a rigid transformation in 3D space:

\[
\begin{pmatrix}
  x_c \\
  y_c \\
  z_c \\
  1
\end{pmatrix} =
\begin{pmatrix}
  R & t \\
  0 & 1
\end{pmatrix}
\begin{pmatrix}
  x_w \\
  y_w \\
  z_w \\
  1
\end{pmatrix}.
\]

Now we can describe the whole projection for a point \((x_w, y_w, z_w)^T\) in a world coordinate frame onto an image point \((u, v)^T\) in pixels:

\[
\lambda
\begin{pmatrix}
  u \\
  v \\
  1
\end{pmatrix} =
K
\begin{pmatrix}
  I_{3 \times 3} & 0 \\
  0 & 1
\end{pmatrix}
\begin{pmatrix}
  R & t \\
  0 & 1
\end{pmatrix}
\begin{pmatrix}
  x_w \\
  y_w \\
  z_w \\
  1
\end{pmatrix}.
\]

A projective setting

More abstractly, the physical pin-hole camera is nothing but a mathematical central projection from \( P^3 \) to \( P^2 \). Such a projection \((x, y, z, t)^T \mapsto (u, v, w)^T\) preserves the collinearity, i.e. lines in space are projected onto lines in the image. It is a linear transformation from \( P^3 \) to \( P^2 \), so it is is represented by a \( 3 \times 4 \) matrix \( P \):

\[
\lambda
\begin{pmatrix}
  u \\
  v \\
  w
\end{pmatrix} =
P_{3 \times 4}
\begin{pmatrix}
  x \\
  y \\
  z \\
  t
\end{pmatrix}.
\]

This \( 3 \times 4 \) projection matrix is the most general camera model without considering nonlinear optical distortion. The study of the camera now becomes the study of this \( 3 \times 4 \) projection matrix.

Properties of the projection matrix.

- It has eleven degrees of freedom as the twelve entries of the matrix are homogeneous, so there are at most eleven independent parameters describing a camera.
- The rank \( P_{3 \times 3} \) is at most three as it has only three rows.
- It follows from the rank-three constraint that the one-dimensional kernel of the projection matrix is the projection center or the camera center \( c \), i.e. \( P_{3 \times 4} c = 0 \).

If the projection center is not at infinity, then \( c = (o, 1)^T \). Hence,

\[
 o = \left( -P_{3 \times 3}^{-1} P_{3 \times 4} \right)_1.
\]
The camera center being the kernel of the projection matrix is not affected by any projective transformation in space.

- Look at the row vectors of the matrix

\[ P = \begin{pmatrix} u^T \\ v^T \\ w^T \end{pmatrix} \]

Each row vector is a four vector, so it can be interpreted as a plane. The \( u^T x = 0 \) is the plane going through the camera center and the \( u \)-coordinate axis of the image plane. The \( v^T x = 0 \) is the plane going through the camera center and the \( v \)-coordinate axis of the image plane. The \( w^T x = 0 \) is the plane going through the camera center and parallel to the image plane. Again, the intersection of these three planes is the camera center \( c \), the kernel of the projection matrix.

- The plane \( w^T x = 0 \) is called the principal plane. It is a plane going through the camera center and parallel to the image plane, so the points on the principal plane \( w^T x = 0 \) project onto the \( w = 0 \), which is the line at infinity of the image plane. Two parallel planes intersect at the line of infinity that is common to the whole family of the parallel planes.

- Look at the column vectors of the

\[ P = (p_1, p_2, p_3, p_4) \]

The first column vector \( p_1 \) is obtained by projecting the point \((1, 0, 0, 0)^T\) by \( P \). It is thus the image of the \( x \) axis direction. Similarly, \( p_2 \) is the image of the \( y \) direction, and \( p_3 \) is the image of the \( z \) direction. \( p_4 \) is the image of the origin \((0, 0, 0, 1)^T\).

- It can be determined or calibrated from six 3D-2D point correspondences.

**Decomposition of \( P \).** Using the QR decomposition theorem (RQ to be more exact), the \( 3 \times 3 \) submatrix of the projection matrix becomes \( KR \) where \( K \) is the upper triangular matrix and \( R \) is an ortho-normalized rotation matrix such that \( P = K(R, t) \). The eleven parameters of \( P \) are thus restructured into five in \( K \), and three in \( R \), and three in \( t \). This comes to exactly the same number of parameters or the degrees of freedom of the matrix as the intrinsic and extrinsic compositions of the projection matrix assembled from a usual Euclidean point of view in the previous section. This confirms that the \( 3 \times 4 \) projection matrix is the most general description of a pinhole camera if we do not take the nonlinear optical distortions into account.

**The image of the absolute conic.** Take a picture of the absolute conic \( x_3^2 x_4 = x_4 = 0 \) by the camera \( P \). The image of the absolute conic is

\[ u^T (KK^T)^{-1} u = 0, \]

which is the conic of the intrinsic parameters \( K \). For a four-parameter camera without skew, the image of the absolute conic is
3.2 The single-view geometry

\[
\left( \frac{u - u_0}{a_u} \right)^2 + \left( \frac{v - v_0}{a_v} \right)^2 = 1^2,
\]

which is a pure imaginary ellipse.

3.2.2 Where is the camera?

We start with a geometric method that constructs the camera center from point correspondences, which is equivalent to camera calibration.

Given a set of six point correspondences \( \{A, B, C, D, E, F\} \) in space and \( \{a, b, c, d, e, f\} \) in an image plane, we want to recover the unknown camera center \( O \). The configuration is illustrated in Figure 3.4. In this section, we use lower-case letters for 2D points in an image plane and upper-case letters for 3D points in space.

Consider the pencil of five planes \( OAB, OAC, OAD, OAE, \) and \( OAF \), which go through the axis \( OA \) and one of the points \( B, C, D, E, F \), respectively. The lines of intersection of these planes with the image plane are the \( ab, ac, ad, ae, \) and \( af \). We can then obtain the cross-ratio of any pencil of four planes by taking measures of the cross-ratios of the corresponding pencil of lines. For instance,

\[
\{OAB, OAC; OAD, OAE\} = \{ab, ac; ad, ae\}.
\]

Take a known plane \( P \), for instance, the plane \( BCD \) defined by the three points \( B, C, \) and \( D \). The plane \( P \) intersects the pencil of planes with the axis \( OA \) in the pencil of lines with the vertex \( a' \), which is the point of intersection of the line \( OA \) with the plane \( P \). The lines \( AB, AC, AD, AE, AF \) intersect the plane \( P \) in the known points \( B', C', D', E', \) and \( F' \), respectively.

Recall that Chasles’s theorem states that for four points \( A, B, C, D \) in a projective plane, with no three of them collinear, the locus of the vertex of a pencil of lines passing through these four points and having a given cross ratio is a conic (see Figure 3.3. Reciprocally, if \( M \) lies on a conic passing through \( A, B, C \) and \( D \), the cross-ratio of four lines \( MA, MB, MC, MD \) is a constant, and is independent of the point \( M \) on the conic.

![Fig. 3.3 A Chasles conic defined by a constant cross-ratio of a pencil of lines.](image-url)
The cross-ratio of the pencil of lines \((A'B', A'C', A'D', A'E')\) can be measured so the point \(A'\) lies on the conic determined by \(B', C', D', E'\) in the plane \(P\) by Chasles's theorem. We can do the same construction while replacing the point \(F'\) by the point \(E'\). Thus, \(A'\) lies on a second conic determined by \(B', C', D', F'\) on the plane \(P\) (see Figure 3.4).

![Figure 3.4 Determining the view line associated with a known point A.](image)

As a result, the point \(A'\) lies at the same time on the two conics that already have three common points \(B', C',\) and \(D'\). Thus, the point \(A'\) must be the fourth intersection point of these two conics. The two conics are known and the three of their intersection points are also known, the remaining fourth unknown intersection point is unique. We can therefore reconstruct the viewing line \(AA'\). The same construction can be applied to any of the other points \(B, C, D, E,\) and \(F\). Finally, the camera center \(O\) is the intersecting point of the bundle of lines \(AA', BB', CC', DD', EE',\) and \(FF'\).

**The critical configurations**

The critical configurations to a given method of a problem are those configurations that fail to solve the problem with the method or give multiple solutions to the problem.
• When the six points in space are coplanar, this constructive method does not hold. Clearly, for instance, the cross-ratio of the pencil of lines \((ab, ac; ad, ae)\) cannot be measured anymore as the point \(a\) is confused with \(b, c, d,\) and \(e\) if \(A\) is coplanar with \(B, C, D,\) and \(E.\)

• Moreover, when the six points and the camera center lie on a twisted cubic, both its non-degenerate and degenerate forms, the method fails. This is due to the fact that a twisted cubic in space is projected onto a plane as a conic if the projection center is on the twisted cubic. The lines that project the twisted cubic lie on a quadric cone, which is a degenerate ruled quadric. For instance, there will be only one conic for any chosen plane. The conic through the points \(A', B', C', D', E'\) and the conic through the points \(A', B', C', D', F'\) are the same conic, it will be impossible to construct the intersection of these two conics to obtain the point \(A'.\)

Remarks

• This constructive technique was presented by Tripp [236] in the planar case. The 3D extensions have been presented in [211, 144].

• This technique is not computational, but the construction is simple and elegant. More importantly, it has inspired the development of vision geometry using projective geometry. This is de facto an uncalibrated method in which only the camera center matters while the other camera parameters are bypassed.

3.2.3 The DLT calibration

Given a set of point correspondences \(x_i \leftrightarrow u_i\) between the 3D reference points \(x_i\) and 2D image points \(u_i.\) The camera calibration consists of determining the projection matrix of the camera \(P,\) which contains both the intrinsic and extrinsic parameters of the camera. The non-linear distortions are not considered for the time being. They could be carried out beforehand and jointly optimized in the final optimization.

From the projection equation of each point \(\lambda u_i = Px_i,\) we have two linear equations in the entries \(p_{ij}\) by taking the ratios of the first component and the third one, and the second components and the third one to eliminate the unknown \(\lambda:\)

\[
\begin{pmatrix}
x_i & y_i & z_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i z_i & -u_i \\
0 & 0 & 0 & x_i & y_i & z_i & 1 & -v_i x_i & -v_i y_i & -v_i z_i & -v_i
\end{pmatrix}
\begin{pmatrix}
p_{12}
\end{pmatrix}
= 0.
\]

where we pack the unknown entries \(p_{ij}\) into the twelve vector \(p.\) For \(n\) given point correspondences, we have a homogeneous linear system of equations

\[
A_{2n \times 12} p_{12} = 0.
\]
As the unknowns are homogeneous, there are only eleven degrees of freedom. The dehomogenization could be one of the following constraints:

- \( p_{34} = 1 \), which transforms the homogeneous system into an inhomogeneous system \( Ax = b \).
- \( \|p_{12}\| = 1 \), which is directly solved by the SVD.
- \( p_{21}^2 + p_{32}^2 + p_{33}^2 = 1 \), which becomes a constraint linear system [52]. This constraint proposed by Faugeras and Toscani has the advantage of preserving the rigidity in the decomposition.

As soon as we have \( 5 \frac{1}{2} \) points, or six points if we do not have a half point, the system is solved. The most convenient way might be through a SVD decomposition of the matrix \( A_{2N \times 12} \) with the unknown vector \( x \) normalized. This gives a good initial estimate of the camera matrix if additional data normalization advocated by Hartley in [76] is carried out.

**Algorithm 1 (The DLT calibration)** Given at least six 3D-2D point correspondences \( x_i \leftrightarrow u_i \) for \( i = 1, \ldots, n \) and \( n \geq 6 \), compute the intrinsic parameters \( K \) of the camera and the rotation \( R \) and the translation \( t \) of the camera with respect to the points \( x_i \).

1. Compute a 2D similarity transformation \( T_u \) such that the points \( u_i \) are translated to its centroid and re-scaled so that the average distance equals to \( \sqrt{2} \). Do the same to compute a 3D similarity transformation \( T_x \) for \( x_i \) so that the average distance equals to \( \sqrt{3} \).
2. Apply \( \tilde{u}_i = T_u u_i \) and \( \tilde{x}_i = T_x x_i \).
3. Form the \( \tilde{A}_{2n \times 12} \) matrix with normalized points \( \tilde{u}_i \) and \( \tilde{x}_i \).
4. Solve for \( \tilde{p}_{12} \) by taking the singular vector corresponding to the smallest singular value of \( \tilde{A}_{2n \times 12} \).
5. Convert \( \tilde{p}_{12} \) into the matrix \( \tilde{P} \).
6. Undo the normalization by \( P = T_x^{-1} \tilde{P} T_u \).
7. Decompose \( P \) to obtain the intrinsic parameters in \( K \) and the extrinsic parameters \( R \) and \( t \).

The solution is unique.

**The critical configurations**

- When the points are coplanar, the calibration method fails because of the rank deficiency of the matrix \( A_{2n \times 12} \).
- When the points and the camera center lie on a twisted cubic, including both non-degenerate and degenerate forms of the twisted cubic, the calibration is not unique [16]. The geometric proof is constructed in the previous section.
3.2 The single-view geometry

Remarks

- The half point redundancy of the six points with respect to the strict minimum of 5.5 points in the calibration finds the ramification in the six-point algorithm in Section 3.5.2.
- A more elaborate calibration procedure usually includes a nonlinear optimization after the DLT solutions with additional nonlinear distortion parameters. The methods of using a planar calibration pattern introduced in [254, 215] are also practical.

3.2.4 The three-point pose algorithm

Given a set of point correspondences \( x_i \leftrightarrow u_i \) between the 3D reference points \( x_i \) and 2D image points \( u_i \), and also given the intrinsic parameters of the camera \( K \). The pose estimation consists of determining the position and orientation of the calibrated camera with respect to the known reference points. The camera pose is called space resection in photogrammetry. The difference between the camera calibration and the camera pose is that the intrinsic parameters of the camera needs to be estimated for the calibration and is known for the pose.

The distance constraint

An uncalibrated image points in pixels \( u_i \) and its calibrated counterpart \( \bar{x}_i \) is related by the known calibration matrix \( K \) such that \( u_i = K \bar{x}_i \). The calibrated point \( \bar{x}_i = K^{-1}u_i \) is a three vector representing a 3D direction in the camera-center coordinate frame. For convenience, we assume the direction vector is normalized to a unit vector such that \( \bar{x} \equiv \bar{x}/\|\bar{x}\| \). A 3D point corresponding to the back-projection of an image point/direction \( \bar{x} \) is determined by a depth \( \lambda \) as \( \lambda \bar{x} \). The depth \( \lambda \) is the camera-point distance.

In summary, \( u \) is an image point in pixels; \( \bar{x} \) is the direction vector of an image point for a calibrated camera; \( x' = \lambda \bar{x} \) is a space point corresponding to the image point \( u \) in the camera-centered coordinate frame; and \( x \) is the space point corresponding to the image point \( u \) in the world coordinate frame.

The distance between two 3D points represented by the vectors \( p \) and \( q \) is given by the cosine rule:

\[
\|p - q\|^2 = \|p\|^2 + \|q\|^2 - 2p^Tq.
\]

Applying this to the normalized direction vectors representing the 3D points in the camera frame, and using the fact that \( \|x_p\| = 1 \), gives:

\[
d_{pq}^2 = \lambda_p^2 + \lambda_q^2 - c_{pq} \lambda_p \lambda_q,
\]
where \( c_{pq} = 2x_i^T x_j = 2 \cos(\theta_{pq}) \) is a known constant from the image points, and \( d_{pq} \) is the known distance between the space points.

**The three points**

If we are given three points, we have three pairs of points and three quadratic equations

\[
\begin{align*}
 f_{12}(\lambda_1, \lambda_2) &= 0, \\
 f_{13}(\lambda_1, \lambda_3) &= 0, \\
 f_{23}(\lambda_2, \lambda_3) &= 0
\end{align*}
\]

of the form

\[
 f_{ij}(\lambda_i, \lambda_j) \equiv \lambda_i^2 + \lambda_j^2 - 2\lambda_i\lambda_j \cos \theta_{ij} - d_{ij}^2 = 0
\]

for the three unknown distances \( \lambda_1, \lambda_2, \lambda_3 \).

The polynomial system has a Bézout bound of \( 2 \times 2 \times 2 = 8 \) solutions. But the quadratic equations do not have linear terms, so \( \lambda_i \leftrightarrow -\lambda_i \) preserves the form and the eight solutions should appear in four pairs. There are many ad hoc elimination techniques in the literature [206, 54] to effectively obtain a polynomial of degree four. Our favorite one is the classical Sylvester resultant, which first eliminates \( \lambda_3 \) from \( f_{13}(\lambda_1, \lambda_3) \) and \( f_{23}(\lambda_2, \lambda_3) \) to obtain a new polynomial \( h(\lambda_1, \lambda_2) \) in \( \lambda_1 \) and \( \lambda_2 \). Then, we can further eliminate \( \lambda_2 \) from \( f_{12}(\lambda_1, \lambda_2) \) and \( h(\lambda_1, \lambda_2) \), and obtain a polynomial in \( \lambda_1 \) of degree eight with only even terms. We let \( x = \lambda_1^2 \), the polynomial of degree four is of the following form:

\[
a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0,
\]
3.2 The single-view geometry

where the coefficients are given in [182] for instances.

The equation has at most four solutions for \( x \) and can be solved in closed-form. Since \( \lambda_1 \) is positive, \( \lambda_1 = \sqrt{x} \). Then \( \lambda_2 \) and \( \lambda_3 \) are uniquely determined from \( \lambda_1 \).

The absolute orientation

The camera-point distances \( \lambda_i \) are then converted into the camera-centered 3D co-
ordinates, \( x'_{ij} = \lambda_i K^{-1} u_i \), of the reference points in space. We are now given a set of
 corresponding 3D points, \( x'_i \leftrightarrow x_i \). We would like to compute a rigid transformation
up to a scale, or a similarity transformation \( R, t \), and \( s \) such that the set of points \( x_i \)
maps to the set of points \( x'_i \),

\[ x'_i = sR x_i + t. \]

This is also called the absolute orientation in the photogrammetry literature.

As there are totally seven degrees of freedom for the similarity transformation, we need \( 2 \frac{7}{3} \) points or at least three points if we do not have a third of a point. It is elementary Euclidian geometry to find a closed-form solution from three points: the rotation maps the normal of the plane determined by the given three points, the scale is the ratio of the vector lengths relative to the centroid, and the translation is the de-rotated centroid. If more than three points are available, the best least-square rotation is obtained in closed-form using quaternions [45, 87]. The determination of the translation and the scale follow immediately from the rotation.

Algorithm 2 (The three-point pose) Given the calibration matrix \( K \) of the camera and three 3D-2D point correspondences \( x_i \leftrightarrow u_i \) for \( i = 1, \ldots, 3 \), compute the rotation \( R \) and translation \( t \) of the camera with respect to the points \( x_i \).

1. Convert 3D points \( x_i \) in coordinates into pair-wise distances \( d_{ij} \). Convert 2D
image points \( u_i \) into the pair-wise angular measures \( \cos \theta_{ij} \) with the calibration
matrix \( K \).
2. Compute the coefficients of the fourth degree polynomial in \( x \) from the quadratic
equations.
3. Solve the equation in closed-form. For each solution of \( x \), get all the camera-
point distances \( \lambda_i \).
4. Convert back the distances \( \lambda_i \) into the 3D points \( x'_i \).
5. Estimate the similarity transformation, the scale, the rotation and the translation,
between the two sets of 3D points \( x_i \) and \( x'_i \).

There are at most four solutions to \( R \) and \( t \).

The critical configurations

There are usually multiple solutions to the pose from the minimum of three points. All critical configurations for which multiple distinct or coinciding (unstable) solutions occur are known in [225, 251].
There is no coplanarity case per se, since three points always define a plane. Better, any additional point coplanar with the three points will give a unique solution to the pose.

If the three points and the camera center lie on a specific twisted cubic or its degenerate forms, the ambiguity of multiple solutions cannot be resolved, regardless of any additional points lying on the same cubic. The solution will be unique if a fourth point not lying on the specific cubic is introduced. The specific twisted cubics are called horopters, which are spatial cubic curves going through the camera center and lying on a circular cylinder known as the ‘dangerous cylinder of space resection’ in photogrammetry.

Remarks

The fourth degree polynomial derived from the resultants is different from that derived in [54]. Many variants of the polynomial are reviewed in [73] following different orders of substitutions and eliminations.

Four points or more not lying on the critical curves will be sufficient for a unique solution, which can be directly obtained by solving a linear system in [182].

Methods for the camera pose using line segments instead of points as image features have also been developed, mostly in computer vision. Dhome et al. [36] and Chen [23] developed algebraic solutions for three-line algorithms, and Lowe [128] used the Newton-Raphson method for any number of line segments. Liu, Huang and Faugeras [123] combined points and line segments into the same pose estimation procedure.

Historically, the robust method RANSAC that we will discuss in Section 5.1.2 is proposed within the context of the pose estimation [54].

3.3 The uncalibrated two-view geometry

The study of the geometry of two views is fundamental as the two views are the strict minimum for us to be able to ‘triangulate’ the lost third dimension of the image points. It is often called stereo vision when we take the images at different viewpoints simultaneously, or motion estimation if the two images are taken sequentially.

The general approach is to understand the geometric constraints for the image points in different views that come from the same physical point in space. These constraints can then be used to find the point correspondences and the triangulation.
3.3 The uncalibrated two-view geometry

3.3.1 The fundamental matrix

Geometrically, given an image point in the first view, this point first back-projects into a line in space, then this line reprojects onto the second view as a line along which all potential corresponding points of the given point in the first view are located. (cf. Figure 3.6). Equivalently, this is to say that the corresponding points in two views and the point in space are on the same plane! This is the coplanarity constraint, from which all the geometries of two views are derived.

A point in one view generates a line in the other view, which we call the epipolar line. The geometry is called the epipolar geometry. The line connecting the two camera centers intersect with each image plane at a point, which we call the epipole. In other words, the epipole in one view is the image of the other camera center. All epipolar lines pass through the epipole, and form a pencil of lines.

Algebraically, without loss of generality, we can always choose the simplest coordinate representation by fixing the coordinate frame with that of the first view, so the projection matrix for the first view is $P_{3 \times 4} = (I, 0)$ and the second becomes a general $P'_{3 \times 4} = (A, a)$. Also with us is a point correspondence $u \leftrightarrow u'$ in two views of a point $x$ in space.

The back-projection line of the point $u$ is defined by the camera center $c = (0, 1)^T$ and the direction $x_\infty = (I^{-1}u, 0)^T$, the point at infinity of this line. The images of these two points in the second view are respectively $e' = (A, a)c = a$ and $u'_\infty = (A, a)x_\infty = Au$. The epipolar line is then given by $l' = e' \times u'_\infty = a \times Au$. Using the anti-symmetrical matrix $[a]_\times$ associated with the vector $a$ to represent the vector product, if we define

$$F = [a]_\times A,$$

which we call the fundamental matrix, then we have

$$l' = Fu.$$
As the point \( u' \) lies on the line \( l' \), it verifies \( u'^T l' = 0 \). We come to the fundamental relationship of the epipolar geometry between two image point correspondence \( u \) and \( u' \):

\[
u'^T F u = 0.
\]

**Properties of \( F \)**

- **Rank constraint.** The fundamental matrix \( F \) is singular and has rank two because of its multiplicative component of an anti-symmetric matrix that has rank two. Geometrically, the set of all epipolar lines passes through the same common point—the epipole, they form a pencil of dimension one.

- **Epipolar geometry.** The kernel of \( F \) is the epipole in the first image plane. To obtain the other epipole, it suffices to transpose \( F \), that is \( F^T e' = 0 \). As if \( u'^T F u = 0 \) is the epipolar geometry between the first and the second view, transposing the equation gives the same in exchange of the views. The epipolar line in the second image plane is \( Fu \), and is \( F^T u' \) in the first image plane.

- **Degrees of freedom.** The degrees of freedom of \( F \) is seven.
  - Algebraically, \( F \) has \( 3 \times 3 = 9 \) homogeneous elements, which makes up only eight degrees of freedom. In addition, it is singular, so only seven degrees of freedom remain.
  - Geometrically, each epipole accounts for two degrees of freedom, which makes four for the two epipoles. The two pencils of corresponding epipolar lines are in homographic correspondence of dimension one, which accounts for three degrees of freedom. The epipoles and the pencils make up the total of seven degrees of freedom for \( F \).
  - Systematically, for a system of two uncalibrated views, each view or camera has 11 degrees of freedom, so the two views amount to 22 degrees of freedom. The entire uncalibrated system is defined up to a projective transformation that accounts for \( 4 \times 4 - 1 \) degrees of freedom, so the total degree of freedom of a system of two uncalibrated views is seven from \( 2 \times 11 - (4 \times 4 - 1) = 7 \). The crucial fact is that the fundamental matrix having exactly seven degrees of freedom is a perfect *minimal* parameterization of the two uncalibrated views!

- **Projective reconstruction.** The fundamental matrix or the epipolar geometry is equivalent to the determination of the two projection matrices of the two views up to a projective transformation, which is equivalent to a projective reconstruction of the two uncalibrated views [42, 79]. Given \( F \), one possible choice for the pair of projective projection matrices is

\[
(I, 0), \quad (\langle e' \rangle, F, e'),
\]

which can be verified by the definition of \( F \). Then, each image point correspondence can be reconstructed up to a projective transformation from the two projective projection matrices.
3.3 The uncalibrated two-view geometry

Remarks

The notion of projective reconstruction, originated by Faugeras and Hartley in [42, 79], changed the scope of shape representation, and initiated the systematic study of vision geometry in a projective setting. Together with the affine shape representation introduced by Koenderink in [103], this projective stratum complements the spectrum of shape representations. Computationally, it is still the determination of the epipolar geometry encoded by the fundamental matrix. With hindsight, the point correspondence information of rigid objects in different views encoded by the projective structure via fundamental matrices has a profound impact on solving the fundamental structure from motion.

3.3.2 The seven-point algorithm

The fundamental matrix has seven degrees of freedom. Since each point correspondence generates one constraint, so we expect that seven point correspondences suffice to obtain a solution.

Each image point correspondence in two views \( u \leftrightarrow u' \) generates one equation in the unknown entries of \( F \) from \( u'^T F u = 0 \),

\[
(u'u, u'v, u'v', u'v, v' u, v' v, v' v, u, v, 1)f_9 = 0,
\]

where \( f_9 = (f_1, \ldots, f_9)^T \) is the nine vector of the entries of the fundamental matrix \( F \). For seven point correspondences, we obtain a linear homogeneous system of seven equations,

\[
A_{7 \times 9} f_9 = 0,
\]

which is not yet sufficient to obtain a unique solution. But a one-dimensional family of solutions parametrized by \( x/t \) is given by \( f_9 = xa + tb \), where \( a \) and \( b \) are the two right singular vectors corresponding to the two smallest singular values of the \( A_{7 \times 9} \). We now add the rank two constraint of \( F \) which is the vanishing of

\[
\text{Det}(F(x, t)) = 0.
\]

By expanding this determinant, we obtain a cubic equation in \( x \) and \( t \),

\[
ax^3 + bx^2 t + cxt^2 + dt^3 = 0.
\]

Each solution to \( x/t \) from the cubic equation gives one solution to \( f_9 \), and therefore one solution to the fundamental matrix \( F \). The cubic equation can have either one or three real solutions.

Algorithm 3 (The seven-point fundamental matrix) Given seven image point correspondences \( u_i \leftrightarrow u'_i \) for \( i = 1, \ldots, 7 \), compute the fundamental matrix \( F \) between the two views.
1. Form the $A_{7 \times 9}$ matrix.
2. Construct a one-dimensional family of solutions to $f_0$ by linearly combining the two singular vectors $a$ and $b$ corresponding to the two smallest singular values of $A$.
3. Solve the cubic equation in closed-form with coefficients obtained from $a$ and $b$.
4. Obtain a solution vector $f$ from each real solution of the cubic equation.
5. Convert $f_0$ into the matrix form $F$.

There are at most three possible solutions to $F$.

The critical configurations

- When the seven points are coplanar, there will be an infinite number of solutions to the fundamental matrix, regardless of any additional points coplanar with the seven points.
- The seven points and the two camera centers make up a set of nine points, which determines a quadric surface. If the quadric surface is a proper quadric but not a ruled quadric, there will be only one solution. If the quadric surface is a ruled quadric surface, for instance, a hyperboloid of one sheet, there will be three solutions. The ambiguity of multiple solutions cannot be resolved regardless of any additional points belonging to the same ruled quadric. The solution will be unique if an eighth point not belonging to the ruled quadric is introduced [106, 137].

Remarks

This formulation and solution, which appeared in several papers in the early 1990s in computer vision, is simple, elegant and practical. It is equivalent to the original Sturm’s method using algebraic geometry back in 1869 in [216], which was re-introduced into the vision community by Faugeras and Maybank [48].

3.3.3 The eight-point linear algorithm

If we add one more point correspondence and ignore the rank constraint on the matrix $F$, the solution can be simply obtained from a pure linear system $A_{8 \times 9}f_0 = 0$. The excitement is that this eight-point algorithm is remarkably simple. It has been known to photogrammetrists, a more recent revival could be found in [124, 44]. Not until recently, this linear algorithm suffered from the numerical instability. Hartley in [76] proposed a data normalization that makes the algorithm numerically more stable.
Algorithm 4 (The linear eight-point fundamental matrix) Given at least eight image point correspondences \( u_i \leftrightarrow u'_i \) for \( i = 1, \ldots, n \), and \( n \geq 8 \), compute the fundamental matrix \( F \) between the two views.

1. Transform the image points in each image plane by \( \tilde{u} = Au \) and \( \tilde{u}' = A'u' \) such that the image point cloud in each image plane is centered at the centroid and re-scaled to have the average distance to the centroid equal to \( \sqrt{2} \).
2. Apply the linear method \( \tilde{A}_{8 \times 9} \tilde{f}_9 = 0 \) to the transformed points.
3. Take the right singular vector corresponding to the smallest singular value as the solution vector \( \tilde{f}_9 \).
4. Convert \( \tilde{f}_9 \) to the matrix form \( \tilde{F} \).
5. Enforce the rank two constraint by setting the smallest singular value of \( \tilde{F} \) to be zero and recomposing the matrix \( \tilde{F}_2 \).
6. Undo the normalization to obtain \( F = A' \tilde{F}_2 A \).

The solution is unique.

Remarks

- The simplicity of this linear eight-point algorithm is attractive as it is a linear method for both the minimum eight points and redundant \( n \) points, but it is sub-optimal. The recommended optimal method in Chapter 5 is first to initialize the solution by running the RANSAC on the seven-point algorithm, then to optimize numerically.
- The idea of taking advantage of data redundancy to ‘linearize’ algorithms has been explored in other vision geometry problems [208, 122, 167, 182].

3.4 The calibrated two-view geometry

When the two views are calibrated, there remains only the relative orientation between the two views. The relative orientation has five degrees of freedom. Three for the rotation, and two for the direction of the translational vector. Unfortunately the magnitude of the translation is unrecoverable. The geometry of the calibrated views is characterized by the essential matrix bearing more constraints than its counterpart, the fundamental matrix, in the uncalibrated case.

3.4.1 The essential matrix

For the two calibrated cameras in a Euclidean setting, without loss of generality, we can center and align the world coordinate frame with the first camera, the second camera is relatively oriented by a rotation \( R \) and displaced by a translation vector \( t \).
This amounts to assigning the projection matrices for the two cameras to

\[(I, 0)\) and \((R, t)\),

which act on the calibrated image points \(x\) and \(x'\). The calibrated image points and the original pixel points are related by \(u = Kx\) and \(u' = K'x'\) through the given intrinsic parameters \(K\) and \(K'\) for the two cameras of the two views.

Formally, the projection equations are exactly the same as for the uncalibrated case. If we define

\[E \equiv [t]_x R = TR,\]

the same derivation based on the coplanarity constraint (the vectors \(x\), \(x'\), and \(t\)) gives

\[x'^TEx = 0\]

for the calibrated image points \(x\) and \(x'\). The matrix \(E\) is called the essential matrix of the two calibrated cameras.

**Relation between \(E\) and \(F\)**

By substituting \(u' = K'x'\) and \(u = Kx\) into the fundamental matrix equation \(u^TFu = 0\), we obtain \(x'^T(K'^TFK)x = 0\), that is,

\[E = K'^TFK.\]

The essential matrix \(E\) inherits all the properties of the \(F\). It is rank two, thus \(\text{Det}(E) = 0\) and \(E^Tt = Et' = 0\).

**The Demazure constraints**

Moreover, the essential matrix \(E\) is the product of an antisymmetric matrix and a rotation matrix \(R\) such that \(RR^T = I\). The rotation matrix differs from an arbitrary \(3 \times 3\) matrix \(A\) for the uncalibrated fundamental matrix \(F\). It is straightforward to see that \(EE^T = TRR^T = -T^2 = (t^Tt)I - tt^T\), so \(t^Tt\) and \(-EE^T = tt^T\).

As \(t^2_1 + t^2_2 + t^2_3 = t^Tt = 1/2\text{Trace}(EE^T)\), we multiply on the right by \(E\) and \(tt^TE = t(E^Tt)^T = 0\), and obtain

\[D(E) \equiv EE^T - \frac{1}{2}\text{Trace}(EE^T)E = 0,\]

where the nine elements of the matrix give nine polynomial equations of degree three in the entries of the essential matrix [33, 41].

Together with the rank two constraint, the decomposability of \(E\) into the product of an antisymmetric and a rotation matrix is now equivalent to
which are ten homogeneous polynomial equations of degree three on the essential matrix $E$. We call these ten polynomials the Demazure constraints. These ten equations are linearly independent, and one can verify by brute force [33]. But obviously, they are not algebraically independent. Indeed it is a simple matter to verify that the vanishing determinant equation is a polynomial consequence of the first nine equations $D(E) = 0$ related to the trace.

The Huang-Faugeras Constraint

Since the singular values of $E$ are the eigenvalues of $EE^T$, the nine constraints $D(E) = 0$ are equivalent to the Huang-Faugeras constraint that a matrix $E$ is an essential matrix if and only if the two non-zero singular values of the rank two matrix $E$ are equal [89]. Thus, a real 3 by 3 matrix $E$ is an essential matrix if and only if it has one singular value equal to zero and the other two singular values equal to each other.

It is straightforward to verify that the two characterizations of the essential matrix, either by the Demazure constraints or by the Huang-Faugeras constraint are algebraically equivalent to the two polynomials $\text{Det}(E) = 0$ and $\frac{1}{2}\text{Trace}(EE^T)^2 - \text{Trace}((EE^T)^2) = 0$.

3.4.2 The five-point algorithm

There are five degrees of freedom for the essential matrix $E$. Five pairs of corresponding image points are sufficient for at most ten solutions. Demazure answered this question by using the algebraic geometry to characterize the variety of all the essential matrices $E$ embedded in $P^8$. The Demazure variety is defined by the ten polynomial equations and five linear equations:

$$
D(E) = 0, \\
\text{Det}(E) = 0, \\
x_i^TEx_i = 0, \text{ for } i = 1, \ldots, 5.
$$

Demazure established that the variety of essential matrices has dimension five, and the degree of the variety is ten. Therefore there are at most ten distinct essential matrices from five pairs of corresponding image points. This corresponds to the old results by Kruppa using a different approach. One of the intersection points should be counted twice as a double tangent point, so the eleven solutions of Kruppa should be ten as pointed out by Faugeras and Maybank in [48].

The way Demazure characterizes the essential matrix plays a key role in the modern methods of solving this system of polynomial equations for the given five
points [233, 156, 212]. They all use the same characterizing polynomials of the variety of essential matrices in Eq. 3.4.2, in particular, the linear independency of the ten algebraically redundant polynomial equations, and the subspace of co-dimension five characterized by the five linear equations.

**Solving equations**

The idea is to reduce the polynomial equations in nine homogenous unknowns $e_{ij}$ of the essential matrix to four homogenous unknowns from the five linear equations given by $x_i^T E x_i = 0$, for $i = 1, \ldots, 5$, which is a $5 \times 9$ linear homogeneous system. The four-dimensional null space of the system spans the solution space of $E$ now parametrized by four remaining unknowns $x, y, z,$ and $w$. Each of the nine components are

$$ e_{ij} = a_{ij}x + b_{ij}y + c_{ij}z + d_{ij}w, $$

where the coefficients come from the computed null space of the $5 \times 9$ system.

After this, we have ten cubic polynomials in $x, y,$ and $z$ by substituting $e_{ij}$ with $x, y, z$ and setting $w = 1$. Triggs [233] used the sparse resultant method to solve the system. Nistér [156] carried out *ad hoc* eliminations of these ten polynomials to first exhibit an explicit polynomial of degree ten in one variable, thus leading to a direct solution of the problem.

Stewénius et al. [212] continued with a more systematic elimination. They chose a graded lexicographic order, with total degree first and lexicographic order second, for the twenty monomials of the system,

$$(x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3, x^2, xy, xz, y^2, yz, z^2, x, y, z, 1) = x$$

with $x > y > z$. We divide $x$ into two parts $l$ and $b$ such that $x = (l, b)$. The vector $l$ contains the first ten monomials

$$ l = (x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3), $$

whose elements are of degree three in $x, y$ and $z$, and are the leading terms of the ten polynomials. The vector $b$ contains the last ten monomials

$$ b = (x^2, xy, xz, y^2, yz, z^2, x, y, z, 1), $$

whose elements are of degree two or less in $x, y,$ and $z$.

Now we write the ten cubic polynomials in matrix form:

$$ A_{10 \times 20} x = 0. $$

Since these equations are generally linearly independent, so are the matrix rows. Gauss-Jordan elimination reduces $A$ to $A = (l, b)$. 


3.4 The calibrated two-view geometry

\[(I_{10 \times 10}, B_{10 \times 10}) \begin{pmatrix} 1 \\ b \end{pmatrix} = 0.\]

where \(I\) is a \(10 \times 10\) identity matrix and \(B\) is a \(10 \times 10\) matrix.

The crucial observation of Stewénius et al. is that, knowing that there are ten solutions, the dimension of the vector space \(A = \mathbb{C}[x, y, z]/I\) is ten, where \(I\) is the ideal generated by the ten cubic polynomials. Thus, there should be exactly ten basis monomials for the vector space \(A\). It can be verified that the ten monomials in \(b\) are effectively the only ten non-leading monomials, none of which is divisible by any of the leading terms in \(I\). At the same time, these ten polynomials after Gauss-Jordan reduction must be a Gröbner basis for the graded lexicographic order. The fact that the Gauss-Jordan reduced form of the ten polynomial equations is a Gröbner basis for the graded lexicographic order has important consequences—the system can be solved via eigenvalues.

A polynomial \(f \in \mathbb{C}[x, y, z]\) defines a linear mapping represented by a \(10 \times 10\) matrix \(A_f\) (called the action matrix in [212]) by multiplication operator from \(A = \mathbb{C}[x, y, z]/I\) to itself. We assume that the \(x\)-coordinates of the solutions are distinct, so we choose to take \(f = x\). Each basis monomial from \(B\) is multiplied by \(x\), then the normalized product modulo \(G\) is expanded in the basis \(B\) to contribute one column to the matrix \(A_x\). By inspecting \(B_{10 \times 10}\), we have

\[
A_x = \begin{pmatrix}
-b_1 & -b_2 & -b_3 & -b_5 & -b_6 & -b_8 & 1_1 & 1_2 & 1_4 & 1_7
\end{pmatrix},
\]

where \(b_i\) is the \(i\)-th column vector of \(B_{10 \times 10}\) and the number one’s index is its row position in the matrix. Finally, the eigenvectors of \(A_x^T\) are the basis monomials evaluated at the solution points, which can be trivially reconstructed from the basis monomials.

**Rotation and translation from \(E\)**

Once we obtained an essential matrix \(E\) from \(e_{ij}\) via \(x, y, z\). We let the singular value decomposition of the \(E\) be \(E = UD\operatorname{Diag}(1, 1, 0)V^T\), such that \(\det(U) > 0\) and \(\det(V) > 0\); and we let

\[
D = \begin{pmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

If the last column vector of \(U\) is \(u_3\), \(t\) is either \(u_3\) or \(-u_3\); the rotation

\[
R = UD^TV^T \text{ or } UD^TV^T.
\]

Of the four possible solutions, the one that triangulates a 3D point in front of both cameras is the correct solution.
Algorithm 5 (The five-point relative orientation) Given five image point correspondences \( u_i \leftrightarrow u'_i \) for \( i = 1, \ldots, 5 \), compute the rotation and the translation between the two views.

1. Form the \( 5 \times 9 \) linear system.
2. Compute the four-dimensional null space of the linear system by SVD.
3. Obtain the coefficients of \( e_{ij} \) parametrized by three parameters \( x, y, z \) from the above null space.
4. Form the \( A_{10 \times 20} \) matrix from the ten cubic polynomial constraints.
5. Obtain the reduced matrix \( B_{10 \times 10} \) from \( A_{10 \times 20} \) by Gauss-Jordan elimination.
6. Form the multiplication matrix \( A_x \) from \( B_{10 \times 10} \).
7. Compute the eigenvectors of \( A_x^T \).
8. Each eigenvector is a solution to \( b \), and
\[
x = b_7/b_{10}, \quad y = b_8/b_{10}, \quad \text{and} \quad z = b_9/b_{10}.
\]
9. Compute \( e_{ij} \) from \( x, y, z \) and convert \( e_{ij} \) into the matrix \( E \).
10. Decompose the \( E \) into the \( R \) and \( t \) by SVD.
11. Choose the \( R \) and \( t \) that reconstruct a 3D point in front of the cameras.

There are at most ten solutions to \( R \) and \( t \).

The critical configurations

- One important advantage of the calibrated five-point algorithm is that it does not suffer from the singularity of coplanar points. Better, when the five points are coplanar, the solution is unique if we enforce the reconstructed points to be in front of the cameras.
- If the five points and the two camera centers lie on a specific ruled quadric or its degenerate forms, the ambiguity of multiple solutions cannot be resolved, regardless of any additional points belonging to the same quadric. The solution will be unique if a sixth point not belonging to the specific quadric is introduced. The specific ruled quadrics are called orthogonal ruled quadrics, which are generated by the intersection line of corresponding planes of a pair of congruent pencils of planes [106, 251]. The ‘congruent’ pencils means that the angle between the corresponding planes of the pencils is constant. The ‘orthogonal’ ruled quadric means that cross sections of parallel planes orthogonal to the specific line of the two camera centers are circles. One simple example is a circular cylinder with the line of the two camera centers on the cylinder and parallel to the axis, illustrated in Figure 3.7. If we split the line of the two camera centers into two lines, move them apart but keep each line going through only one camera center, the circular cylinder deforms into an orthogonal hyperboloid of one sheet.
3.5 The three-view geometry

The study of the geometry of the three views naturally extends that of the two views [199, 209, 78, 178, 229, 18, 19, 145]. Though two views are the minimum for depth, the point correspondence between the two views is not unique, a point corresponds only to a line, not a point. Intuitively, if we consider that a set of three views is three pairs of two views, then it generally suffices to have the epipolar geometry between the first and the third, and between the second and the third so that the corresponding point in the third image is completely determined. (see Figure 3.8).
The three-view geometry is fundamental in that it is the minimum number of views for which the image point correspondence ambiguity can be removed.

### 3.5.1 The trifocal tensor

The geometry constraints have been derived for points as the consequences of the collinearity for the single view or the coplanarity for the two views. When a third view is introduced, it is interesting to notice that the line is more natural than the point. We start with lines and end with points.

#### Line transfer

Geometrically, a line \( l \) in a view \( P \) defines a plane \( p = P^T l \) that is the back-projection plane of the line or the viewing plane of the line. It is known that there does not exist any geometric constraint for a pair of corresponding lines in two views, as the two back-projected planes of the two lines in the two views are always and generally intersecting on a line.

Given a triplet of corresponding lines in three views,

\[
l \leftrightarrow l' \leftrightarrow l''.
\]

The basic geometry is that these three planes should intersect on a line rather than a point at which the three planes usually meet. It is equivalent to
3.5 The three-view geometry

\[
\text{Rank} \begin{pmatrix} p_1^T \\ p_2^T \\ p_3^T \end{pmatrix} = 2.
\]

Algebraically, we take the projection matrices of the three views to be \((I, 0), (A, a), (B, b)\) as we have the freedom to arbitrarily fix the first one. The matrices can further be decomposed into vectors

\[
A = \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix}, \quad B = \begin{pmatrix} b_1^T \\ b_2^T \\ b_3^T \end{pmatrix}.
\]

The three planes can then be written as

\[
p_1 = (l_0), \quad p_2 = (A^T l'), \quad \text{and} \quad p_3 = (B^T l'').
\]

The rank two constraint says that the vectors representing the planes are linearly dependent. Notice that one entry is zero, so the linear combination coefficients are obtained from the last row, so we obtain

\[
l = \alpha(A^T l') + \beta(B^T l'') = a^T l' \begin{pmatrix} \alpha \end{pmatrix} + b^T l'' \begin{pmatrix} \beta \end{pmatrix}.
\]

Introducing \(T_k = a_k b^T - a b_k^T\), then \(l_k = l^T T_k l'\). We obtain

\[
l = \begin{pmatrix} l^T T_1 l' \\ l^T T_2 l' \\ l^T T_3 l'' \end{pmatrix}.
\]

This is the line transfer equation as it transfers or predicts the line \(l\) in the first view from the given lines \(l'\) and \(l''\) in the other two views.

The three matrices \(T_k\) actually form a tensor \(T^{ijk}\). The line transfer equation is

\[
l_k = l^T T^{ijk} \]

in tensorial notation.

**Point transfer**

Geometrically, the point transfer was depicted previously in Figure 3.8 in three views as the intersection point of the two epipolar lines. Algebraically, we take a triplet of corresponding points in three views

\[
u \leftrightarrow u' \leftrightarrow u''
\]

Then, for a point \(x = (x, y, z, t)\) in space, we obtain \(x = (u, t)\) from the first view. Projecting \((u, t)\) onto the second view, we have:

\[
X'u' = (A, a) \begin{pmatrix} u' \\ t \end{pmatrix}.
\]
Fig. 3.9 The geometry of three views for line correspondence.

which gives two possibilities for the scalar $t$: $t_1 = (a_1^T u - u'a_3^T u)/(u'a_3 - a_1)$ and $t_2 = (a_2^T u - v'a_3^T u)/(v'a_3 - a_2)$. Using one of two possible values of $t$, we project the space point, now a known point, onto the third view. By re-arranging the terms and introducing the vectors $t_{ij} = a_i b_j - b_j a_i$, we obtain two systems of equations with the two possible values of $t$:

$$\lambda''^n u'' = \begin{pmatrix} u't_{31} - t_{11}' & u't_{32} - t_{12}' & u't_{33} - t_{13}' \\ u't_{32} - t_{12}' & u't_{33} - t_{13}' & u't_{33} - t_{23}' \\ u't_{33} - t_{13}' & u't_{33} - t_{23}' & u't_{33} - t_{33}' \end{pmatrix} u$$

These two sets of three equations are homogeneous. By eliminating the scale factor, we obtain four equations in nine vectors $t_{ij}$.

It is interesting to observe that by continuing this game of indexes of the vectors $t_{ij}$, and denoting the $k$-th element of the vector $t_{ij}$ with a third index $k$, the set of the elements indexed by $i$, $j$ and $k$ forms a tensor $T_{ik}^j$ where $i$, $j$, and $k$ vary from 1 to 3. More interestingly, this tensor is exactly the same as that introduced for the study of the lines! We call this tensor the trifocal tensor, which may play the same role for three views as that of the fundamental matrix for two views.

If we use tensorial notation $\epsilon_{abc}$ that is 0 when $a$, $b$, and $c$ are distinct, +1 when $abc$ is an even permutation of 123, and -1 when $abc$ is an odd permutation of 123. The point transfer equation is nicely summarized in

$$u'(u'^i \epsilon_{jac})u'^k \epsilon_{kbd} T_{ik}^{ab} = o_{cd}.$$
The properties of the trifocal tensor $T$

- **Degree of freedom.** We can count the degree of freedom of a given system. For a set of two uncalibrated views, each view or camera has 11 degrees of freedom, so the two views amount to 22 degrees of freedom. The entire uncalibrated system is defined up to a projective transformation that accounts for $4 \times 4 - 1 = 15$ degrees of freedom, so the total degrees of freedom of a system of two uncalibrated views are seven from $2 \times 11 - (4 \times 4 - 1) = 7$. The fundamental matrix having exactly 7 degrees of freedom is a perfect parameterization of the two uncalibrated views that has exactly 7 degrees of freedom (cf. page 44).

For a set of three uncalibrated views, using the same counting arguments, the degrees of freedom should be $3 \times 11 - (4 \times 4 - 1) = 18$. This counting argument suggests that there are $27 - 1 = 26$ inhomogeneous entries for the tensor $T$, there should be $27 - 1 - 18 = 8$ independent algebraic constraints among these entries. Some of them are simple, for instance, the rank of each $3 \times 3$ matrix of the tensor is two—similar to the fundamental matrix. The others are known [80, 47] but are complex and difficult to be integrated into an efficient numerical schema.

- **Linear algorithms.** For each image point correspondence $u \leftrightarrow u' \leftrightarrow u''$, there are many trilinear equations by the trifocal tensor. Four of them (for instance those we mentioned above) are linearly independent for an image point correspondence, and two are linearly independent for an image line correspondence in three views. This immediately suggests that seven point correspondences are sufficient to linearly solve for the trifocal tensor $T$, and thirteen line correspondences are sufficient as well [209, 245]. More generally, and also new to the three views, any mixture of $n$ points and $m$ lines will be sufficient as long as $4n + 2m \geq 26$.

- **Relations between $F$ and $T$.** Now, we answer the question raised at the beginning of this section. Geometrically, both the trifocal tensors and the fundamental matrices describe the 'transfer' of the image correspondence in the first two views to the third view. Are they the same? Indeed, they are equivalent for generic view configurations and generic points [177]. For instance, when the three camera centers of the three views are aligned, clearly, the epipolar geometries degenerate so it is impossible to intersect the epipolar lines to transfer to the third view, but the trifocal tensor survives! More general discussion on the N-view case is given in Section 3.6.

**Remarks**

- Despite many nice mathematical properties of the trifocal tensor, it is an over-parametrization of the geometry of three views. Its direct estimation is not yet settled. The apparently attractive linear algorithms (for instance, the seven-point linear algorithm) still suffer from the instability, and the nonlinear optimization methods suffer from the complexity of integrating all algebraic constraints.
The geometry of the three uncalibrated views can be solved with a minimum of six point correspondences in the following section by solving a cubic equation. It is straightforward to convert the projection matrices computed with the six-point algorithm to the trifocal tensor if that conversion is of any necessity.

### 3.5.2 The six-point algorithm

#### The number of invariants

Given a set of points or any geometric configuration, the number of invariants is, roughly speaking, the difference between the dimension of the configuration and the dimension of the transformation group that acts on the configuration, if the dimension of the isotropy group of the configuration is null (cf. [71, 152]). For a set of six points in \( \mathbb{P}^3 \), there are three \((3 \times 6 - (16 - 1) = 3)\) absolute invariants under the action of general linear group \( GL(3) \) in \( \mathbb{P}^3 \).

#### The canonical representation

**The six points in space.** For a set of six points \( \{P_i, i = 1, \ldots, 6\} \) in space, there are three invariants. One of the simplest ways to consider the invariants is to use the canonical projective coordinates, then the three inhomogeneous projective coordinates of any sixth point with respect to a projective basis defined by any five of them characterize the set of the six points.

Any five of the six points, with no three of them collinear and no four of them coplanar, can be assigned to the canonical projective coordinates as follows

\[
(1, 0, 0, 0)^T, (0, 1, 0, 0)^T, (0, 0, 1, 0)^T, (0, 0, 0, 1)^T \quad \text{and} \quad (1, 1, 1, 1)^T.
\]

This choice of basis uniquely determines a space projective transformation \( A_{4 \times 4} \), which transforms the original five points into this canonical basis. The sixth point is transformed to \((x, y, z, t)^T\) by \( A_{4 \times 4} \).

Thus, \( x : y : z : t \) gives three independent absolute invariants of six points and equivalently a projective reconstruction of the six points.

**The six image points.** For a set of six image points \( \{p_i, i = 1, \ldots, 6\} \), usually measured in inhomogeneous coordinates \((\bar{u}_i, \bar{v}_i, 1)^T, i = 1, \ldots, 6\). Take any four of them, with no three of them collinear, and assign them with the canonical projective coordinates in \( \mathbb{P}^2 \):

\[
(1, 0, 0)^T, (0, 1, 0)^T, (0, 0, 1)^T, \quad \text{and} \quad (1, 1, 1)^T.
\]

This choice of basis uniquely determines a plane projective transformation \( A_{3 \times 3} \), which transforms the fifth \((\bar{u}_5, \bar{v}_5, 1)^T\) and sixth point \((\bar{u}_6, \bar{v}_6, 1)^T\) to \((\bar{u}_5, \bar{v}_5, \bar{w}_5)^T\) and \((\bar{u}_6, \bar{v}_6, \bar{w}_6)^T\) respectively.
Thus, \( \bar{u}_5 : \bar{v}_5 : \bar{w}_5 \) and \( \bar{u}_6 : \bar{v}_6 : \bar{w}_6 \) give four independent absolute invariants of six image points.

**Projection between space and image**

Remember that if we are given six point correspondence in space and image, we can calibrate the camera, i.e. estimate the parameters of the projection matrix \( P \) if these points are known. Now the points in space are unknown, and we want to eliminate the camera parameters. Remember also that theoretically we only need 5\( \frac{1}{2} \) points for the DLT calibration. The redundancy introduced by this half point is the key in the following development.

**Elimination of camera parameters.** For a set of six point correspondences \( u_i \leftrightarrow x_i \), for \( i = 1, \ldots, 6 \), established as

\[
\begin{align*}
(1, 0, 0)^T & \leftrightarrow (1, 0, 0, 0)^T, \\
(0, 1, 0)^T & \leftrightarrow (0, 1, 0, 0)^T, \\
(0, 0, 1)^T & \leftrightarrow (0, 0, 1, 0)^T, \\
(1, 1, 1)^T & \leftrightarrow (0, 0, 1, 0)^T, \\
(\bar{u}_5, \bar{v}_5, \bar{w}_5)^T & \leftrightarrow (1, 1, 1, 1)^T, \\
(\bar{u}_6, \bar{v}_6, \bar{w}_6)^T & \leftrightarrow (x, y, z, t)^T.
\end{align*}
\]

The six point correspondences lead to \( 12 = 2 \times 6 \) equations from (??) for the eleven unknowns of the camera \( P_{3 \times 4} \), as we assumed that the camera was uncalibrated. There will still remain one \( (1 = 12 - 11) \) independent equation after eliminating all eleven unknown camera parameters \( p_{ij} \).

Substituting the canonical projective coordinates of the first four point correspondences into the projection equation reduces to a three-parameter family of the camera matrix \( P \):

\[
P = \begin{pmatrix}
\alpha & \lambda \\
\beta & \lambda \\
\gamma & \lambda
\end{pmatrix},
\]

where \( \alpha = p_{11}, \beta = p_{22}, \gamma = p_{33}, \) and \( \lambda = p_{34} \).

Using the projective coordinates of the fifth point further reduces to a one-parameter family of the camera matrix, homogeneous in \( \mu \) and \( \nu \):

\[
P = \begin{pmatrix}
\bar{u}_5 \mu - \nu \\
\bar{v}_5 \mu - \nu \\
\bar{w}_5 \mu - \nu
\end{pmatrix}
\]

Finally, adding the sixth point, all the camera parameters \( \mu \) and \( \nu \) (the original \( p_{ij} \)) are eliminated, we obtain the following homogeneous equation in the unknowns \( x, y, z, t \) and the known \( \{(\bar{u}_i, \bar{v}_i, \bar{w}_i), \) for \( i = 5, 6\} \).
\[ \bar{w}_6(\bar{u}_5 - \bar{v}_5)xy + \bar{v}_6(\bar{w}_5 - \bar{u}_5)xz + \\
\bar{u}_5(\bar{v}_6 - \bar{w}_6)zt + \bar{v}_5(\bar{u}_6 - \bar{v}_6)yz + \\
\bar{w}_5(\bar{u}_6 - \bar{u}_6)yt + \bar{u}_5(\bar{v}_6 - \bar{w}_5)zt = 0. \] (3.3)

**Invariant interpretation.** The above Equation 3.3 can be arranged and interpreted as an invariant relationship between the invariants of \( P^3 \) and \( P^2 \).

If \( i_j \) and \( \xi_j \) denote respectively

\[ i_1 = \bar{w}_6(\bar{u}_5 - \bar{v}_5), i_2 = \bar{v}_6(\bar{w}_5 - \bar{u}_5), i_3 = \bar{u}_5(\bar{v}_6 - \bar{w}_6), i_4 = \bar{u}_6(\bar{v}_5 - \bar{w}_5), i_5 = \bar{v}_5(\bar{u}_6 - \bar{v}_6), i_6 = \bar{w}_5(\bar{u}_6 - \bar{v}_6), \]

and

\[ \xi_1 = xy, \xi_2 = xz, \xi_3 = xt, \]
\[ \xi_4 = yz, \xi_5 = yt, \xi_6 = zt. \]

By the counting arguments, there are only four invariants for a set of six points in \( P^2 \), so the homogeneous invariants \( \{i_j, j = 1, \ldots, 6\} \) are not independent, but are subject to one \((1 = 5 - 4)\) additional constraint

\[ i_1 + i_2 + i_3 + i_4 + i_5 + i_6 = 0. \]

The verification is straightforward from the definition of \( i_j \). There are only three independent invariants for a set of six points in space, so the invariants \( \{\xi_j, j = 1, \ldots, 6\} \) are subject to two additional constraints which are

\[ \frac{\xi_1}{\xi_2} = \frac{\xi_5}{\xi_6} \text{ and } \frac{\xi_2}{\xi_3} = \frac{\xi_4}{\xi_5}, \]

by inspection of the definition of \( \xi_j \).

The invariant relation 3.3 is then simply expressed as a bilinear homogeneous relation,

\[ i_1\xi_1 + i_2\xi_2 + i_3\xi_3 + i_4\xi_4 + i_5\xi_5 + i_6\xi_6 = 0, \] (3.4)

which is independent of any camera parameters, therefore can be used for any uncalibrated images.

**Invariant computation from three images**

There are only three independent invariants for a set of six points. With each view gives one invariant relation, therefore we can hope to solve for these three invariants if three views of the set of six points are available. The three homogeneous quadratic equations in \( x, y, z, t \) from three views can be written as follows.
Each equation represents a quadratic surface of rank three. The quadratic form does not have any square terms $x^2$, $y^2$, $z^2$, and $t^2$. It means that the quadratic surface goes through the vertices of the reference tetrahedron whose coordinates are canonical $(0, 0, 0, 1)^T$, $(0, 0, 1, 0)^T$, $(0, 1, 0, 0)^T$, and $(1, 0, 0, 0)^T$. This is easily verified by substituting these points into the equations. In addition, as the coefficients of the quadratic form $i^{(k)}$ are subject to $\sum_{j=1}^{6} i_{j} = 0$, all the equations necessarily pass through the unit point $(1, 1, 1, 1)^T$ as well.

According to Bezout’s theorem, three quadratic surfaces meet in $2 \times 2 \times 2 = 8$ points. Since they already pass through the five known points, so only $8 - 5 = 3$ common points remain. Thus, the maximum number of solutions for $x : y : z : t$ is only three. Indeed, by successively computing resultants to eliminate variables, we first obtain homogeneous polynomials in $x$, $y$ and $t$ of degree three, $g_1(x, y, t)$ and $g_2(x, y, t)$ by eliminating $z$ between $f_1$ and $f_3$ and between $f_2$ and $f_3$. Then, by eliminating $y$ between $g_1$ and $g_2$, we obtain a homogeneous polynomial in $x$, $t$ of degree eight, which can be factorized, as expected, into

$$xt(x - t)(b_1x^2 + b_2xt + b_3t^2)(a_1x^3 + a_2x^2t + a_3xt^2 + a_4t^3).$$

It is evident that the linear factors and the quadratic factor lead to trivial solutions. Thus, the only nontrivial solutions for $x/t$ are those of the cubic equation,

$$a_1x^3 + a_2x^2t + a_3xt^2 + a_4t^3 = 0. \quad (3.5)$$

The implicit expressions for $a_i$ can be easily obtained with Maple (for instance, at www.cs.ust.hk/quan/publications/proccubicxyz). The cubic equation may be solved algebraically by Cardano’s formula, either for $x/t$ or $t/x$.

**Projective reconstruction**

The determination of the projective coordinates of the sixth point completely determines the geometry of three cameras up to a projective transformation. All geometry quantities related to the three views follow trivially.

- Given the projective coordinates of the sixth point $(x, y, z, t)$, the projection matrix $P_i$ of each camera can be computed using Equation 3.2.
- Given the projection matrices, any point correspondence in the three views can be reconstructed in $\mathbb{P}^3$ up to a collineation.
- Given the projection matrices, it is also straightforward to compute any pair-wise fundamental matrices and any trifocal tensors.
Algorithm 6 (The six-point projective reconstruction) Given six image point correspondences \( u_i \leftrightarrow u'_i \leftrightarrow u''_i \) for \( i = 1, \ldots, 6 \) in three views, compute the projective projection matrices \( P, P', \) and \( P'' \) of the two views.

1. Compute a plane homography in each view to transform the points \( u_i, u'_i \) and \( u''_i \) respectively into the canonical basis.
2. Compute the \( i \) th in each view from the transformed image points.
3. Obtain the coefficients of the cubic equations (The coefficients can be downloaded from www.cs.ust.hk/quan/publications/).
4. Solve the cubic equation 3.5 in closed-form.
5. For each solution of \( x \), obtain \( (x, y, z, t) \).
6. Compute the projection matrix \( P \) for each view from \( (x, y, z, t) \).

There are at most three solutions to \( P, P' \) and \( P'' \).

The critical configurations

- When the six points are coplanar, there will be an infinite number of solutions, regardless of any additional points that are coplanar with the six points.
- The six points define a twisted cubic. If any one of three camera centers lie on the twisted cubic, the solution is ambiguous regardless of the existence of any additional points lying on the cubic.
- If the six and more points and the three camera centers lie on a six-parameter family of a curve of degree four, the solution is ambiguous [139, 80].

Remarks

- The trifocal tensor over-parameterizes the geometry of the three views. The six points give a minimal parameterization, thus the six-point algorithm is the choice for the estimation of the three uncalibrated views. The readers should not be afraid of directly plugging the lengthy coefficients from a symbolic computation like Maple into any programming language like C [178, 119].
- It is no accident that the six points algorithm in three views and the seven points algorithm in two views both lead to the solution of a cubic equation. There is in fact an intrinsic duality between these two configurations elucidated by Carlsson in [18, 19]. For a reduced camera

\[
P = \begin{pmatrix} \alpha & \lambda \\ \beta & \lambda \\ \gamma & \lambda \end{pmatrix}
\]

with the canonical representation for the reference tetrahedron, it is interesting to see that the camera center is \( o = (1/\alpha, 1/\beta, 1/\gamma, -1/\lambda) \) and
3.5 The three-view geometry

\[
\begin{pmatrix}
\alpha & \lambda \\
\beta & \lambda \\
\gamma & \lambda
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z \\
t
\end{pmatrix}
=
\begin{pmatrix}
x t \\
y t \\
z t \\
\frac{1}{\lambda}
\end{pmatrix}
\begin{pmatrix}
\alpha \\
\beta \\
\gamma \\
\lambda
\end{pmatrix},
\]

in which we can exchange the role of the camera center \( o \) and a space point \( x \). So the geometry of \( n \) points in \( v \) views is equivalent to that of \( n - 1 \) points in \( v + 1 \) views. In particular, the six points in three views is equivalent to the seven points in two views.

3.5.3 The calibrated three views

We start with some counting arguments to obtain the minimal configurations of four points in three calibrated views, then we propose a minimal parameterization to the system. But the problem is still open due to a lack of efficient algorithms.

The counting arguments

For 3D reconstruction from image points, each image point gives two constraints, each 3D point introduces three degrees of freedom and each camera pose introduces six degrees of freedom, but there are seven degrees of freedom in the 3D coordinate system (six for the Euclidean coordinate frame and one for the scene scale). So a system of \( n \) points visible in \( m \) calibrated images yields \( 2mn \) constraints in \( 3n + 6m - 7 \) unknowns. To have at most finitely many solutions, we therefore need:

\[
2mn \geq 3n + 6m - 7.
\] (3.6)

Minimal cases are given by an equality here, so we look for integer solutions for \( m \) and \( n \) satisfying:

\[
n = 3 + \frac{2}{2m - 3}.
\]

For \( m = 2 \), \( n = 5 \), a minimum of five points is required for a two-image relative orientation and Euclidean reconstruction.

For any \( m \geq 3 \), \( n \) is between 3 and 4, thus at least four points are required for \( m \geq 3 \)-view Euclidean reconstruction from unknown space points. In fact, four points in three images suffice to fix the 3D structure, after which just three points are needed in each subsequent image to fix the camera pose (the standard three point pose problem [206]). So at least four points are always required for Euclidean reconstruction, and of the minimal \( m \geq 3 \) cases, the four points and three views problem is the most interesting.

Note that for four points in three views, Eq. (3.6) becomes \( 2mn = 24 \geq 3n + 6m - 7 = 23 \). The counting suggests that the problem is over-specified.
An over-specified polynomial system generically has no solutions, but here (in the noiseless case) we know that there is at least one (the physical one). It is tempting to conclude that the solution is unique. In an appropriate formulation this does in fact turn out to be the case, but it needs to be proved rigorously. For example in the two image case, the ‘twisted’ partner of the physical solution persists no matter how many points are used, so the system becomes redundant but always has two solutions. The issue is general. Owing to the redundancy, for arbitrary image points there is no solution at all. To have at least one solution, the image points must satisfy some (here one, unknown and very complicated) polynomial constraints saying that they are possible projections of a possible 3D geometry. When these constraints on the constraints (i.e. on the image points) are correctly incorporated, the constraint counting argument inevitably gives an exactly specified system, not an over-specified one. To find out how many roots actually occur, the only reliable method is detailed polynomial calculations.

The basic Euclidean constraint

An uncalibrated image point in pixels $u_i$ and its calibrated counterpart $\bar{x}_i$ is related by the known calibration matrix $K$ such that $u_i = K\bar{x}_i$. The calibrated point $\bar{x}_i = K^{-1}u_i$ is a three vector representing a 3D direction in the camera-center coordinate frame. For convenience, we assume the direction vector is normalized to a unit vector such that $\bar{x} \equiv \bar{x}/\|\bar{x}\|$. A 3D point corresponding to the back-projection of an image point/direction $\bar{x}$ is determined by a depth $\lambda$ as $\lambda \bar{x}$. The depth $\lambda$ is the camera-point distance.

The distance between two 3D points represented by 3-vectors $p$ and $q$ is given by the cosine rule:

$$\|p - q\|^2 = \|p\|^2 + \|q\|^2 - 2p^Tq$$

Applying this to the normalized direction vectors representing the 3D points in the camera frame, and using the fact that $\|\bar{x}\| = 1$, gives:

$$\lambda_p^2 + \lambda_q^2 - c_{pq}\lambda_p\lambda_q = \delta_{pq}^2$$

where $c_{pq} = 2x_i^Tx_j = 2\cos(\theta_{pq})$ is a known constant from the image points, and $\delta_{pq}$ is the unknown distance between the space points.

These cosine-rule constraints are exactly the same as in calibrated camera pose from known 3D points in Section 3.2.4 and [206, 54], except that here the inter-point distances $\delta_{pq}$, which are known quantities as $d_{pq}$ in the pose estimation, are unknowns that must be eliminated.

The four-point configuration

A set of four 3D points has 6 independent Euclidean invariants — $3 \times 4 = 12$ degrees of freedom, modulo the six degrees of freedom of a Euclidean transformation —
and it is convenient to take these to be the \( \binom{4}{2} = 6 \) inter-point distances \( \delta_{pq} \), i.e. the edge lengths of the tetrahedron in Figure 3.10. This structure parametrization is very convenient here, as the \( \delta_{pq} \) appear explicitly in the cosine-rule polynomials. However, it would be less convenient if there were more than 4 points, as the inter-point distances would not all be independent.

Fig. 3.10 The configuration of four non-coplanar points in space.

**The polynomial system.** For \( n \) images of 4 points, we obtain a system of \( 6n \) homogeneous cosine-rule polynomials in \( 4n \) unknowns \( \lambda_{ip} \) and \( 6 \) unknowns \( \delta_{ij} \):

\[
f(\lambda_{ip}, \lambda_{iq}, \delta_{pq}) = 0, \quad i = 1, \ldots, m, \quad p < q = 1, \ldots, 4
\]

The unknown inter-point distances \( \delta_{ij} \) can be eliminated by equating cosine-rule polynomials from different images

\[
\lambda_{ip}^2 + \lambda_{iq}^2 - c_{pq}^i \lambda_{ip} \lambda_{iq} = \lambda_{jp}^2 + \lambda_{jq}^2 - c_{pq}^j \lambda_{jp} \lambda_{jq} \quad (= \delta_{pq}).
\]

leaving a system of \( 6(m - 1) \) homogeneous quadratics in \( 4m \) homogeneous unknowns \( \lambda_{ip} \).

For \( m = 3 \) images we obtain 18 homogeneous polynomials \( f(\lambda_{ip}, \lambda_{iq}, \delta_{pq}) = 0 \) in 18 unknowns \( \lambda_{ip}, \delta_{ij} \), or equivalently 12 homogeneous polynomials in 12 unknowns \( \lambda_{ip} \). De-homogenizing (removing the overall 3D scale factor) leaves just 11 inhomogeneous unknowns, so as expected, equation counting suggests that the system is slightly redundant.

**The simulation method using random rational numbers.** The polynomial system has only finitely many solutions if the dimension of the variety is zero, an infinite number of solutions for positive dimension, and no consistent solutions for negative dimension. Generally, we expect to have only a finite number of solutions for a well-defined geometric problem when the dimension of the variety is zero. Combining elimination from lexicographic Gröbner bases with numerical root-finding for one-variable polynomials conceptually gives a general polynomial solver. But it is often impossible simply because Gröbner bases cannot be computed with limited computer resources. This is true for our polynomial system \( g(\lambda_{ip}, \lambda_{iq}, \lambda_{jp}, \lambda_{jq}) \) with parametric coefficients \( c_{pq}^i \). We choose the approach offered by Macaulay (http://www.math.uiuc.edu/Macaulay2) among other computer
algebra systems, which allows the computation with coefficients in modular arithmetic (a finite prime field \( k = \mathbb{Z}/(p) \)) to speed up computation and minimize memory requirements. Currently, by using this simulation method on random rational simulations, currently we can establish the following results.

- Euclidean reconstruction from four point correspondences that come from unknown coplanar points has a unique double solution in three views, but is generally unique in \( n > 3 \) views.
- Euclidean reconstruction from four point correspondences that come from known coplanar points is generally unique in \( n \geq 3 \) views.

Remarks

The minimal Euclidean reconstruction in multiple views is still inconclusive due to its algebraic complexities. Longuet-Higgins [125] described an iterative method of finding the solutions to the case of four points in three perspective images by starting from the solution obtained in the simplified approximate three scaled orthographic images. Holt and Netravali [84] proved that “there is, in general, a unique solution for the relative orientation . . . However, multiple solutions are possible, and in rare cases, even when the four feature points are not coplanar. [84]” They used some results from algebraic geometry to draw general conclusions regarding the number of solutions by considering a single example. This is certainly one step further to show the general uniqueness of the solutions, but still many questions remain unanswered and efficient algorithms do not exist for this problem as yet.

3.6 The N-view geometry

3.6.1 The multi-linearities

A natural question to ask is whether there exist similar constraints to the fundamental matrix and the trifocal tensor for the case of more than three views. In fact, it is easy to see that everything comes from the projection equation, so all these constraints are pure algebraic consequences of these projection equations.

The formulation

Based on [231, 49], for \( n \) views, we can easily re-write the projection equations into the following form. Given a space point \( \mathbf{x} \) projected into \( n \) views:
\[ \lambda_1 u_1 = P_1 x, \]
\[ \lambda_2 u_2 = P_2 x, \]
\[ \vdots \]
\[ \lambda_n u_n = P_n x. \]

We can then pack these equations into matrix form:

\[ \begin{bmatrix} P_1 u_1 \\ P_2 u_2 \\ \vdots \\ P_n u_n \end{bmatrix} \begin{bmatrix} x \\ -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_n \end{bmatrix} = \mathbf{0}, \]

where the vector \( (x, -\lambda, -\lambda', \ldots, -\lambda^{(n)})^T \) cannot be zero, so the rank of the matrix \( \mathbf{M} \) can not exceed \( n + 4 \). This implies that all its minors of \( (n + 4) \times (n + 4) \) have to vanish. The expansion of all these minors gives all the geometric constraints that we could imagine among multiple views.

**The multi-linear constraints**

The minors could be formed in different combinations. When the minors are formed by the elements involving only two different views, the expansion of these minors yields the geometric constraints for these two views. We call them the *bilinear constraints*, which are de facto the epipolar constraints or the fundamental matrix. Likewise, when the minors are formed by the elements involving only three different views, the expansion of these minors yields the constraints for these three views. We call them the *trilinear constraints*, which are de facto the trifocal tensor. When the minors are formed by the elements involving four different views, the expansion of these minors yields the constraints for these four views. We call them the quadrilinear constraints or quadrifocal constraints. Since \( \mathbf{M} \) has \( n + 4 \) columns and \( 3n \) rows, the minors cannot have more than four projection matrices involved in the formation. Therefore, there is no constraints for more than four different views.

**The algebraic relations**

The algebraic relations among these multi-linear constraints have been investigated in [75, 231, 49, 177].

- It is easy to see that the quadrilinear constraints are not algebraically independent, and that they break up into trilinear and bilinear constraints. Furthermore, these quadrilinear constraints are redundant due to the intrinsic Grassmannian quadratic relations among the minors.
The relation between the trilinear and bilinear is more subtle. The key is that we should single out the degenerate configurations of views and points. For generic view configurations and generic points, all multi-linear constraints may algebraically be reduced to the algebraically independent bilinear constraints. In other words, all matching constraints are contained in the ideal generated only by the bilinear constraints for generic views and points. As a consequence, \(2n - 3\) algebraically independent binlinearities from pairs of views completely describe the algebraic/geometric structure of \(n\) uncalibrated views for generic views and points. For degenerate points of generic views, each type of constraint reduces differently. The exact reduced form of the matching constraints are also made explicit by computer algebra.

Remarks

The formulation allows us to have a global picture of all geometric constraints for multiple views. These multi-linear constraints generally over-parametrize the geometry of \(n\) views, the numerical exploration of these constraints is not yet elucidated.

3.6.2 Auto-calibration

The idea of auto-calibration is to upgrade a computed projective reconstruction to a Euclidean reconstruction by taking advantages of the intrinsic constraints on the intrinsic parameters of the cameras in multiple views. The idea was introduced by Maybank and Faugeras [138] using the Kruppa’s equation with a minimal parameterization.

Basic equations

Given a projective reconstruction of 3D points \(x_i\) and camera matrices \(P_j\). By the very definition of a projective reconstruction from uncalibrated cameras, it is defined up to a space projective transformation because of

\[
u_i = Px = (PH)(H^{-1}x)
\]

for any projective transformation \(H\).

In case of auto-calibration, we seek a specific \(4 \times 4\) transformation \(H\) such that, if \(x_i \mapsto H^{-1}x_i\), the camera matrices should be brought to the calibrated Euclidian ones as

\[
P_jH = \lambda_jK_j(R_j, t_j).
\]

The auto-calibration directly looks for the unknown intrinsic parameters \(K_j\), and indirectly the rectifying transformation matrix \(H\). The \(K_j\) and \(H\) are not indepen-
We first try to eliminate $R_j$ and $t_j$. We may take the $3 \times 3$ part of the $3 \times 4$ matrices, and use the orthogonality constraint $RR^T = I$ to obtain

$$(P_jH)_{3 \times 3}(P_jH)^T = \lambda_j K_j K_j^T.$$ 

Matrix arrangements lead to

$$P_j(H \begin{bmatrix} I_3 & 0 \\ 0 & 0 \end{bmatrix} H^T)P_j^T = \lambda K_j K_j^T.$$ 

The fundamental auto-calibration equation, written for each view, is therefore

$$P_jQP_j^T = \lambda_j C_j,$$ 

(3.7)

where

$$Q \equiv H \begin{bmatrix} I_3 & 0 \\ 0 & 0 \end{bmatrix} H^T$$ and $C_j \equiv K_j K_j^T$. 

(3.8)

So the auto-calibration is to compute the unknowns $Q$ and $C_j$ from the known $P_j$ via the five equations in Equation 3.7 for each view.

Once $Q$ and $C_j$ are computed, the $H$ and $K_j$ can be computed. $K_j$ is obtained from Choleski decomposition of $C_j$, and $H$ is computed as $A\text{Diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \lambda_3^{1/2}, 1)$ from the eigen-decomposition of $Q = A\text{Diag}(\lambda_1, \lambda_2, \lambda_3, 0)A$.

### Geometric interpretations

We give a geometric interpretation for the above $Q$ and $C_j$. Let the vector $u = (u_1, u_2, u_3, u_4)^T$ denote the dual plane coordinate in space, not an image point $u = (u, v)$.

A similarity Euclidian transformation up to a scale is a projective transformation that leaves the so-called absolute conic invariant. The absolute conic is a virtual conic on the plane at infinity. It is given by $x_1^2 + x_2^2 + x_3^2 = 0 = x_4$ as a point locus. The dual form of the absolute conic, viewed as an envelop of planes in plane coordinate, is

$$u_1^2 + u_2^2 + u_3^2 = u^T \text{Diag}(1, 1, 1, 0)u = 0.$$ 

We call the dual of the absolute conic the dual absolute quadric. It is a degenerate quadric envelop. The unique null space of this rank three matrix is the plane at infinity. The canonical form of the dual absolute quadric in a Euclidian frame is

$$\text{Diag}(1, 1, 1, 0) = \begin{bmatrix} I_{3 \times 3} & 0 \\ 0 & 0 \end{bmatrix}.$$ 

We see that the representation of the dual absolute quadric is easier to manipulate than that of the absolute conic as it is a simple matrix with the implicit rank constraint within.

From the above definition of $Q$ in 3.8, we have
\[ \mathbf{HQH}^T = \begin{pmatrix} 1_{3 \times 3} & 0 \\ 0 & 0 \end{pmatrix}. \]

Immediately, the \( \mathbf{Q} \) can be interpreted as the dual absolute quadric in a projective frame before the rectification. The \( \mathbf{H} \) is the rectifying homography that brings the dual absolute quadric in an arbitrary projective basis to the canonical Euclidean basis.

Using the plane-line projection relation \( \mathbf{u} = \mathbf{P}^T \mathbf{l} \), a dual quadric \( \mathbf{u}^T \mathbf{Q} \mathbf{u} = 0 \) is projected onto \( (\mathbf{P}^T \mathbf{l})^T \mathbf{Q} (\mathbf{P}^T \mathbf{l}) = 0 \), which is \( \mathbf{l}^T (\mathbf{PQ} \mathbf{P}^T) \mathbf{l} = 0. \)

The equation describes an image conic envelop in dual line coordinates with the \( 3 \times 3 \) symmetric matrix \( \mathbf{PQP}^T \).

Finally, the general auto-calibration equation 3.7 admits a simple geometric interpretation, \textit{i.e.}, the absolute quadric \( \mathbf{Q} \) is projected onto the dual of the image of the absolute conic \( \mathbf{C} \).

### Auto-calibration algorithms

We first use the counting arguments to stress the practical difficulties of auto-calibration, then we give two common and practical methods of auto-calibration.

**The degrees of freedom.** A rank-three quadric \( \mathbf{Q} \) has eight degrees of freedom and a conic \( \mathbf{C} \) has five degrees of freedom. But \( \mathbf{Q} \) and \( \mathbf{C} \) or \( \mathbf{H} \) and \( \mathbf{K} \) are not independent, they are related through the first reference camera matrix \( \mathbf{K}_1 = (\mathbf{I}, \mathbf{0}) \):

\[ \mathbf{H} = \begin{pmatrix} \mathbf{K}_1 \\ \mathbf{v}^T \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \]

and the vector \( \mathbf{v} \) introduces only three additional degrees of freedom. It is easy to see that it is impossible for auto-calibration if we do not impose any additional constraints on \( \mathbf{K}_1 \). The art of auto-calibration makes different assumptions on \( \mathbf{K}_j \) and simplifies the parameterization of \( \mathbf{H} \) so as to make the system more tractable.

**The constancy of intrinsic parameters.** We assume constant but unknown intrinsic parameters \( \mathbf{K}_j \equiv \mathbf{K} \). This was the original assumption used to introduce the concept of auto-calibration. The polynomial system can be iteratively solved starting from three views, and an initial solution to \( \mathbf{K} \) is usually easy to provide.

In this case, we can keep eliminating the additional three independent parameters in \( \mathbf{H} \) or \( \mathbf{Q} \) to have a quadratic polynomial system with only five unknowns from \( \mathbf{K} \). This is equivalent to the so-called \textit{Kruppa equations} introduced to computer vision by Faugeras and Maybank. The Kruppa equations use the minimal parameterization, but are prone to singular cases \[214\]. The redundant absolute quadric parameterization does not suffer from the same singular cases as the Kruppa equations do.

**The variable focal length.** The only reasonable scenario is that each camera has only one unknown focal length. In this case, the equations become linear, observed
3.6 The N-view geometry

by Pollefeys et al. in [169] based on the absolute quadric introduced by Triggs [232]. A small caveat is that we should first transform the known principal point \((u_0, v_0)\) to \((0, 0)\) such that the \(K\) becomes diagonal \(\text{Diag}(f, f, 1)\), then \(C_j = \text{Diag}(f_j^2, f_j^2, 1)\) and

\[
Q = \begin{pmatrix}
\text{Diag}(f_1^2, f_1^2, 1) & a \\
\text{a}^T & ||a||^2
\end{pmatrix}.
\]

Using the four equations from \(c_{11} = c_{22}\), and \(c_{12} = c_{13} = c_{23} = 0\) leads to a linear equation system in the unknowns \(f_j^2, a = (a_1, a_2, a_3)^T\), and \(||a||^2\). This indeed turns out to be a practical method of solving a linear system. We may further to impose \(f_j = f\) in the linear system with less unknowns.

Algorithm 7 (The linear auto-calibration)

Given projective projection matrices \(P_j\) for \(j = 1, \ldots, n\) views, compute the rectification matrix \(H\) and the unknown focal lengths \(f_j\) for \(j = 1, \ldots, n\).

1. Transform each \(K_j\) into \(\text{Diag}(f_j^2, f_j^2, 1)\) with the known \((u_0, v_0)\) for each camera.
2. Form the linear system for each view in the unknowns \(f_j^2, a_1, a_2, a_3, ||a||^2\).
3. Solve the linear system for these unknowns, the focal lengths \(f_j\) and the matrix \(Q\).
4. Decompose the \(Q\) into the \(H\).

The critical configurations

The singularities of auto-calibration has been studied by Sturm in [214]. The critical configurations for auto-calibration are the configurations of the cameras, not those of the points.

- If the principal axes of the \(n\) views meet at a point at infinity, the auto-calibration is impossible. It is actually a translating camera with parallel principal axes. Only affine reconstruction is possible.
- If the principal axes of the \(n\) views meet at a finite point, the linear auto-calibration algorithm is ambiguous.

The auto-calibration is unstable when the configuration of the camera is close to these critical configurations.

Remarks

- The more general formulation using the absolute quadric is due to Triggs [232]. The notion of auto-calibration is conceptually attractive. Given the facts that the intrinsic parameters of the camera are correlated each other, few independent equations exist for each view, and the lack of statistical optimization criterion. It makes the numerical estimation of the auto-calibration unsatisfactory and less robust when compared to the projective reconstruction for instance.
• In practice, most of the intrinsic parameters are well known in advance. For instances, the principal point is in the center of the image plane, there is no skew and the pixel is square. These prior specifications of the intrinsic parameters are not worse than the results obtained by an auto-calibration algorithm for many of the non-metric applications of reconstruction.

There might be only the focal length that is worth auto-calibrating! Even the recommended linear auto-calibration method allows a variable unknown focal length. More robust results for the fixed unknown focal length are reported in [119].

3.7 Discussions

To calibrate or not to calibrate? That has been the subject of discussions for a while. For a pure uncalibrated framework, the seven-point algorithm for two views and the six-point algorithm for three views are the workhorses to be followed by a global bundle-adjustment for the projective structure. Such a system has been developed in [119]. It is remarkable that no any knowledge on cameras is required as long as the images are overlapping and free of nonlinear distortions. Nevertheless, all uncalibrated methods suffer from the singular coplanar scenes, while the calibrated methods do not. There is a big advantage of working in a calibrated framework. The intrinsic parameters are (new digital cameras record all parameters!) often sufficient for non-metric applications. The final bundle-adjustment is compulsory. These arguments support a calibrated framework in which the five-point relative orientation algorithm and the three-point pose algorithm are the workhorses, as the combination of these two gives a working algorithm for the three calibrated views. Nevertheless, the seven-point algorithm is almost a must to be used for finding correspondences at the very beginning of the entire pipeline. If a quick prototyping is necessary, the eight-point in lieu of the seven-point could be used for its implementation simplicity. The open question is an efficient algorithm for the case of three calibrated views with as few as four points.

3.8 Bibliographic notes

The study of the camera geometry has a long history and was originated in photogrammetry [206, 251], in particular for the classical tasks of pose and calibration. The 'DLT' methods have been proposed in [206, 218, 65]. The methods were further improved by Faugeras and Toscani [52] using a different constraint on the projection matrix. Lenz and Tsai [110] proposed both linear and nonlinear calibration methods. The three-point pose algorithm can be traced back to 1841 by the photogrammetrists [73]. Many variants [54, 57, 73] of the basic three-point algorithm have been developed ever since.
The two-view geometry in the uncalibrated projective framework, summarized in the seven-point algorithm, has been revived in the 1990s [48, 42, 146, 145, 43, 130, 131, 79, 76, 228, 255]. The materials are now standard, and can be found in standard textbooks [43, 47, 80, 59, 132]. Nistér and Stévenus’s efforts in making the five-point algorithm workable is a major contribution as it is the probably the most fundamental component of vision geometry. Remarkably, these methods are based on the characterizing polynomials presented by Demazure [33] for the proof of the solutions. The trifocal tensor in the calibrated case for points and lines are from [78, 199, 209, 245], and its introduction into the uncalibrated framework and the development are due to [199, 78]. The six-point algorithm is adapted from [178] by Quan, which first appeared in a conference version [176].

The systematic consideration of multi-linearities and their relations have been investigated in [231, 83, 50]. The auto calibration was originated in [138] by Maybank and Faugeras, but the introduction of the absolute quadric by Triggs in [232] led to the more simple and practical methods presented in [169]. There is also a large body of literature on special auto-calibration methods with special constraints.

Approximative camera models such as orthographic, weak perspective and affine cameras have been considered. A good review can be found in [100, 198, 180]. There have also been methods for specific motions such as planar motion and circular motion [2, 97, 96, 51]. They were purposely not mentioned as they are useful only for specific applications, and are absent in any general structure from motion systems.