19

Auto-Calibration

Auto-calibration is the process of determining internal camera parameters directly from multiple uncalibrated images. Once this is done, it is possible to compute a metric reconstruction from the images. Auto-calibration avoids the onerous task of calibrating cameras using special calibration objects. This gives great flexibility since, for example, a camera can be calibrated directly from an image sequence despite unknown motion and changes in some of the internal parameters.

The root of the method is that a camera moves rigidly, so the absolute conic is fixed under the motion. Conversely, then, if a unique fixed conic in 3-space can be determined in some way from the images, this identifies Ω_∞. As we have seen in earlier chapters, once Ω_∞ is identified, the metric geometry can be computed. An array of auto-calibration methods are available for this task of identifying Ω_∞.

This chapter has four main parts. First we lay out the algebraic structure of the auto-calibration problem, and show how the auto-calibration equations are generated from constraints on the internal or external parameters. Second, we describe several *direct* methods for auto-calibration which involve computing the absolute conic or its image. These include estimating the absolute dual quadric over many views, or the Kruppa equations from view pairs. Third, are *stratified* methods for auto-calibration which involve two steps – first solving for the plane at infinity, then using this to solve for the absolute conic. The fourth part covers a number of special configurations including a camera rotating about its centre, a camera undergoing planar motion, and the motion of a stereo rig.

19.1 Introduction

Auto- (or self-) calibration is the computation of metric properties of the cameras and/or the scene from a set of uncalibrated images. This differs from conventional calibration where the camera calibration matrix $\mathtt{K}$ is determined from the image of a known calibration grid (chapter 7) or properties of the scene, such as vanishing points of orthogonal directions (chapter 8). Instead, in auto-calibration the metric properties are determined directly from constraints on the internal and/or external parameters.

For example, suppose we have a set of images acquired by a camera with fixed internal parameters, and that a projective reconstruction is computed from point cor-

respondences across the image set. The reconstruction computes a projective camera matrix $\mathtt{P}^i$ for each view. Our constraint is that for the actual cameras the internal parameter matrix $\mathtt{K}$ is the same (but unknown) for each view. Now, each camera $\mathtt{P}^i$ of the projective reconstruction may be decomposed as $\mathtt{P}^i = \mathtt{K}^i[\mathtt{R}^i \mid \mathbf{t}^i]$ but in general the calibration matrix $\mathtt{K}^i$ will differ for each view. Thus the constraint will *not* be satisfied by the projective reconstruction.

However, we have the freedom to vary our projective reconstruction by transforming the camera matrices by a homography $\mathtt{H}$. Since the actual cameras have fixed internal parameters, there will exist a homography (or a family of homographies) such that the transformed cameras $\mathtt{P}^i\mathtt{H}$ do decompose as $\mathtt{P}^i\mathtt{H} = \mathtt{K}\mathtt{R}^i[\mathtt{I} \mid \mathbf{t}^i]$, with the same calibration matrix for each camera, so the reconstruction is consistent with the constraint. Provided there are sufficiently many views and the motion between the views is general (see later), then this consistency constrains $\mathtt{H}$ to the extent that the reconstruction transformed by $\mathtt{H}$ is within a similarity transformation of the actual cameras and scene, i.e. we achieve a metric reconstruction.

Although the particular constraints used to achieve a metric reconstruction may differ, this example illustrates the general approach:

(i) Obtain a projective reconstruction $\{\mathtt{P}^i, \mathbf{X}_j\}$.

(ii) Determine a rectifying homography $\mathtt{H}$ from auto-calibration constraints, and transform to a metric reconstruction $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$.

Various flavours of auto-calibration will be covered in the following sections. They differ in the constraints used, and the methods whereby the homography $\mathtt{H}$ is determined. The methods may be divided into two classes: those that directly determine $\mathtt{H}$; and those that are stratified, determining first the projective and then the affine components of $\mathtt{H}$. The advantage of the latter approach is that once an affine reconstruction is achieved, i.e. $\boldsymbol{\pi}_\infty$ is known, the solution for a metric reconstruction is linear.

If camera calibration rather than metric scene reconstruction is the goal, then it is not always necessary to compute an explicit projective reconstruction, and sometimes the camera calibration may be computed more directly than via a rectifying transformation. This is the case, for instance, if a camera rotates about its centre without translation, as is discussed in section 19.6.

19.2 Algebraic framework and problem statement

Suppose we have a projective reconstruction $\{\mathtt{P}^i, \mathbf{X}_j\}$; then based on constraints on the cameras' internal parameters or motion we wish to determine a rectifying homography $\mathtt{H}$ such that $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$ is a metric reconstruction.

We start from the true metric situation with calibrated cameras, and structure represented in a Euclidean world frame. Thus in actuality there are m cameras $\mathtt{P}^i_{\mathrm{M}}$ which project a 3D point $\mathbf{X}_{\mathrm{M}}$ to an image point $\mathbf{x}^i = \mathtt{P}^i_{\mathrm{M}}\mathbf{X}_{\mathrm{M}}$ in each view, where the subscript M indicates that the cameras are calibrated and the world frame is Euclidean. The cameras may be written as $\mathtt{P}^i_{\mathrm{M}} = \mathtt{K}^i[\mathtt{R}^i \mid \mathbf{t}^i]$ for $i = 1, \ldots, m$.

In a projective reconstruction we obtain cameras $\mathtt{P}^i$ which are related to $\mathtt{P}^i_{\mathrm{M}}$ by

$$\mathtt{P}^i_{\mathrm{M}} = \mathtt{P}^i\mathtt{H} \quad i = 1, \ldots, m \tag{19.1}$$

where $\mathtt{H}$ is an unknown 4×4 homography of 3-space. Our goal is to determine $\mathtt{H}$.

To be precise we are not concerned with the absolute rotation, translation and scale of the reconstruction, and we will now factor out this similarity component. We choose the world frame to coincide with the first camera, so that $\mathtt{R}^1 = \mathtt{I}$ and $\mathbf{t}^1 = \mathbf{0}$. Then $\mathtt{R}^i, \mathbf{t}^i$ specifies the Euclidean transformation between the i-th camera and the first, and $\mathtt{P}^1_{\mathrm{M}} = \mathtt{K}^1[\mathtt{I} \mid \mathbf{0}]$. Similarly, in the projective reconstruction we choose the usual canonical camera for the first view, so that $\mathtt{P}^1 = [\mathtt{I} \mid \mathbf{0}]$. Then writing $\mathtt{H}$ as

$$\mathtt{H} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & k \end{bmatrix}$$

the condition $\mathtt{P}^1_{\mathrm{M}} = \mathtt{P}^1\mathtt{H}$ from (19.1) becomes $[\mathtt{K}^1 \mid \mathbf{0}] = [\mathtt{I} \mid \mathbf{0}]\mathtt{H}$, which implies that $\mathtt{A} = \mathtt{K}^1$ and $\mathbf{t} = \mathbf{0}$. In addition, since $\mathtt{H}$ is non-singular, k must be non-zero, so we may assume $k = 1$ (this fixes the scale of the reconstruction). This shows that $\mathtt{H}$ is of the form

$$\mathtt{H} = \begin{bmatrix} \mathtt{K}^1 & \mathbf{0} \\ \mathbf{v}^\mathsf{T} & 1 \end{bmatrix}.$$

This has factored out the similarity component.

The vector $\mathbf{v}$, together with $\mathtt{K}^1$, specifies the plane at infinity in the projective reconstruction since the coordinates of $\boldsymbol{\pi}_\infty$ are

$$\boldsymbol{\pi}_\infty = \mathtt{H}^{-\mathsf{T}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} (\mathtt{K}^1)^{-\mathsf{T}} & -(\mathtt{K}^1)^{-\mathsf{T}}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -(\mathtt{K}^1)^{-\mathsf{T}}\mathbf{v} \\ 1 \end{pmatrix}.$$

We will write $\boldsymbol{\pi}_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ so that $\mathbf{p} = -(\mathtt{K}^1)^{-\mathsf{T}}\mathbf{v}$. To summarize so far we have shown:

Result 19.1. *A projective reconstruction* $\{\mathtt{P}^i, \mathbf{X}_j\}$ *in which* $\mathtt{P}^1 = [\mathtt{I} \mid \mathbf{0}]$ *can be transformed to a metric reconstruction* $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$ *by a matrix* $\mathtt{H}$ *of the form*

$$\mathtt{H} = \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ -\mathbf{p}^\mathsf{T}\mathtt{K} & 1 \end{bmatrix} \tag{19.2}$$

where $\mathtt{K}$ *is an upper triangular matrix. Furthermore,*

(i) $\mathtt{K} = \mathtt{K}^1$ *is the calibration matrix of the first camera.*
(ii) *The coordinates of the plane at infinity in the projective reconstruction are given by* $\boldsymbol{\pi}_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$.

Conversely, if the plane at infinity in the projective frame and the calibration matrix of the first camera are known, then the transformation $\mathtt{H}$ *that converts the projective to a metric reconstruction is given by (19.2).*

From this result it follows that to transform a projective to a metric reconstruction it is sufficient to specify 8 parameters – the three entries of $\mathbf{p}$ and five entries of $\mathtt{K}^1$. This agrees with a geometric counting argument. Finding metric structure is equivalent to specifying the plane at infinity and the absolute conic (three and five degrees of freedom respectively). In a metric reconstruction the calibration $\mathtt{K}^i$ of each camera, its rotation $\mathtt{R}^i$ relative to the first camera, and its translation $\mathbf{t}^i$ relative to the first camera up to a single common scaling, i.e. $\mathbf{t}^i \mapsto s\mathbf{t}^i$, are all determined.

We now develop the basic auto-calibration equations. We denote the cameras of the projective reconstruction as $\mathtt{P}^i = [\mathtt{A}^i \mid \mathbf{a}^i]$. Multiplying out (19.1) using (19.2) gives

$$\mathtt{K}^i\mathtt{R}^i = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\mathtt{K}^1 \quad \text{for } i = 2, \ldots, m \tag{19.3}$$

which may be rearranged as $\mathtt{R}^i = (\mathtt{K}^i)^{-1}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\mathtt{K}^1 \quad i = 2, \ldots, m$. Finally, the rotation $\mathtt{R}^i$ may be eliminated using $\mathtt{R}\mathtt{R}^\mathsf{T} = \mathtt{I}$, leaving

$$\mathtt{K}^i\mathtt{K}^{i\mathsf{T}} = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\mathtt{K}^1\mathtt{K}^{1\mathsf{T}}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^\mathsf{T}.$$

Note now that $\mathtt{K}^i\mathtt{K}^{i\mathsf{T}} = \boldsymbol{\omega}^*$, the dual image of the absolute conic (or DIAC) – see (8.11–*p*210). Making this substitution gives the basic equations for auto-calibration:

$$\begin{aligned} \boldsymbol{\omega}^{*i} &= \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\boldsymbol{\omega}^{*1}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^\mathsf{T} \\ \boldsymbol{\omega}^{i} &= \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^{-\mathsf{T}}\boldsymbol{\omega}^{1}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^{-1} \end{aligned} \tag{19.4}$$

the second equation being simply the inverse of the first, with $\boldsymbol{\omega}$ the image of the absolute conic (or IAC). These equations relate the unknown entries of $\boldsymbol{\omega}^{*i}$ or $\boldsymbol{\omega}^i$ $i = 1, \ldots, m$ and unknown parameters $\mathbf{p}$ with the *known* entries of the projective cameras $\mathtt{A}^i, \mathbf{a}^i$.

The art of auto-calibration is to use constraints on the $\mathtt{K}^i$, such as that one of the elements of $\mathtt{K}^i$ is zero, to generate equations on the eight parameters of $\mathbf{p}$ and $\mathtt{K}^1$ from (19.4). *All* auto-calibration methods are variations on solving these equations, and in the following sections we describe several of these methods. Generally methods proceed by computing $\boldsymbol{\omega}^i$ or $\boldsymbol{\omega}^{*i}$ first and extracting the values of the calibration matrices $\mathtt{K}^i$ from these – though iterative methods (such as bundle adjustment) may be parametrized by $\mathtt{K}^i$ directly. The equations (19.4) have a geometric interpretation as mappings on the absolute conic, and we will return to this in section 19.3 and section 19.5.2.

We start with a simple example to illustrate how equations on the eight parameters are generated from (19.4).

Example 19.2. Auto-calibration equations for identical $\mathtt{K}^i$

Suppose that all the cameras have the same internal parameters, so $\mathtt{K}^i = \mathtt{K}$, then (19.4) becomes

$$\mathtt{K}\mathtt{K}^\mathsf{T} = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\mathtt{K}\mathtt{K}^\mathsf{T}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^\mathsf{T} \quad i = 2, \ldots, m. \tag{19.5}$$

Each view $i = 2, \ldots, m$ provides an equation, and we can develop a counting argument

for the number of views required (in principle) in order to be able to determine the 8 unknowns. Each view other than the first imposes 5 constraints since each side is a 3×3 symmetric matrix (i.e. 6 independent elements) and the equation is homogeneous. Assuming these constraints are independent for each view, a solution is determined provided $5(m-1) \geq 8$. Consequently, provided $m \geq 3$ a solution is obtained, at least in principle. Clearly, if m is much larger than 3 the unknowns $\mathtt{K}$ and $\mathbf{p}$ are very over-determined. △

One could imagine using (19.5) as a basis for a direct estimation of the rectifying transformation $\mathtt{H}$. This may be framed as a parametrized minimization problem in which the eight parameters of (19.2) are allowed to vary with the purpose of minimizing a cost function on how well the equations (19.4) are satisfied or measuring closeness to metric structure. Of course, a method of obtaining an initial solution would also be required. In essence these two steps, initial solution and iterative minimization, are what will be investigated in the following sections – though under constraints less restrictive than identical internal parameters.

19.3 Calibration using the absolute dual quadric

The absolute dual quadric, $\mathtt{Q}_\infty^*$ is a degenerate dual (i.e. plane) quadric represented by a 4×4 homogeneous matrix of rank 3. Its importance here is that $\mathtt{Q}_\infty^*$ encodes both $\boldsymbol{\pi}_\infty$ and Ω_∞ in a very concise fashion, for instance $\boldsymbol{\pi}_\infty$ is the null-vector of $\mathtt{Q}_\infty^*$, and it has an algebraically simple image projection:

$$\boldsymbol{\omega}^* = \mathtt{P}\mathtt{Q}_\infty^*\mathtt{P}^\mathsf{T} \tag{19.6}$$

which is simply the projection (8.5–*p*201) of a (dual) quadric. In words, $\mathtt{Q}_\infty^*$ projects to the dual image of the absolute conic $\boldsymbol{\omega}^* = \mathtt{K}\mathtt{K}^\mathsf{T}$.

The idea of auto-calibration based on $\mathtt{Q}_\infty^*$ is to use (19.6) to transfer a constraint on $\boldsymbol{\omega}^*$ to a constraint on $\mathtt{Q}_\infty^*$ via the (known) camera matrix $\mathtt{P}^i$. In this manner the matrix representing $\mathtt{Q}_\infty^*$ may be determined in the projective reconstruction from constraints on $\mathtt{K}^i$, as will be seen below. Indeed, $\mathtt{Q}_\infty^*$ was introduced as a convenient representation for auto-calibration in [Triggs-97].

Once $\mathtt{Q}_\infty^*$ has been determined, then the rectifying homography (19.2) $\mathtt{H}$ that we seek is also determined as shown below. Thus we have a general framework for auto-calibration based on specifying constraints on $\mathtt{K}^i$ to determine $\mathtt{Q}_\infty^*$, and then from $\mathtt{Q}_\infty^*$ determining $\mathtt{H}$. This general approach is summarized in algorithm 19.1. In section 19.3.1 we will concentrate on the second step of this algorithm, estimation of $\mathtt{Q}_\infty^*$. We first fill in some details.

Simple properties of the absolute dual quadric. Section 3.7(*p*83) gives a full description of $\mathtt{Q}_\infty^*$. For the purposes of auto-calibration particularly important properties are summarized here. In a Euclidean frame $\mathtt{Q}_\infty^*$ has the canonical form

$$\tilde{\mathtt{I}} = \begin{bmatrix} \mathtt{I}_{3\times3} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}. \tag{19.7}$$

Objective

Given a set of matched points across several views and constraints on the calibration matrices $\mathtt{K}^i$, compute a metric reconstruction of the points and cameras.

Algorithm

(i) Compute a projective reconstruction from a set of views, resulting in camera matrices $\mathtt{P}^i$ and points $\mathbf{X}_j$.
(ii) Use (19.6) along with constraints on the form of the $\boldsymbol{\omega}^{*i}$ arising from $\mathtt{K}^i$ to estimate $\mathtt{Q}^*_\infty$.
(iii) Decompose $\mathtt{Q}^*_\infty$ as $\mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T}$, where $\tilde{\mathtt{I}}$ is the the matrix $\mathrm{diag}(1,1,1,0)$.
(iv) Apply $\mathtt{H}^{-1}$ to the points and $\mathtt{H}$ to the cameras to get a metric reconstruction.
(v) Use iterative least-squares minimization to improve the solution (see section 19.3.3).

Alternatively, the calibration matrix of each camera may be computed directly:

(i) Compute $\boldsymbol{\omega}^{*i}$ for all i using (19.6).
(ii) Compute the calibration matrix $\mathtt{K}^i$ from the equation $\boldsymbol{\omega}^* = \mathtt{K}\mathtt{K}^\mathsf{T}$ by Cholesky factorization.

Algorithm 19.1. *Auto-calibration based on* $\mathtt{Q}^*_\infty$.

In a projective coordinate frame $\mathtt{Q}^*_\infty$ has the form $\mathtt{Q}^*_\infty = \mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T}$, where $\tilde{\mathtt{I}}$ is the matrix in (19.7). This follows from the projective transformation rule (3.17–*p*74) for dual quadrics, $\mathtt{Q}^*_\infty \mapsto \mathtt{H}\mathtt{Q}^*_\infty\mathtt{H}^\mathsf{T}$. Consequently:

Result 19.3. *In an arbitrary projective frame, the dual absolute quadric is represented by a symmetric* 4×4 *matrix with the following properties.*

(i) *It is singular of rank 3, since* $\mathtt{Q}^*_\infty$ *is a degenerate conic.*
(ii) *Its null space is the vector representing the plane at infinity, since* $\mathtt{Q}^*_\infty\boldsymbol{\pi}_\infty = \mathbf{0}$.
(iii) *It is positive semi-definite (or negative – depending on the homogeneous scale).*

These properties are immediate for $\mathtt{Q}^*_\infty$ in its canonical form in a Euclidean frame, and easily extend to an arbitrary frame.

Extracting the rectifying homography from $\mathtt{Q}^*_\infty$. Given an estimated $\mathtt{Q}^*_\infty$ in a projective coordinate frame we wish to determine the homography $\mathtt{H}$. Extracting $\mathtt{H}$ is a simple matter of decomposing the expression as follows.

Result 19.4. *If* $\mathtt{Q}^*_\infty$ *is decomposed as* $\mathtt{Q}^*_\infty = \mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T}$ *(see notation above), then* $\mathtt{H}^{-1}$ *is a 3D (point) homography that takes the projective coordinate frame to a Euclidean frame.*

Note that a camera is transformed by the inverse of the transformation applied to points, so $\mathtt{H}$ is the correct matrix to apply to a camera to give $\mathtt{P}_\mathrm{M} = \mathtt{P}\mathtt{H}$. Thus $\mathtt{H}$ is the rectifying transformation to apply to cameras. A decomposition of $\mathtt{Q}^*_\infty$ as $\mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T}$ may be easily computed from its eigenvalue decomposition (see section A4.2(*p*580) for Jacobi's algorithm for this).

Equivalence to auto-calibration equations. Equations (19.6) which describe the image projection of $\mathtt{Q}^*_\infty$ are simply a geometric representation of the auto-calibration equations (19.4) as will now be demonstrated.

The forms of $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$ and $\boldsymbol{\omega}^* = \boldsymbol{\omega}^{-1} = \mathtt{K}\mathtt{K}^\mathsf{T}$ for a camera with calibration matrix $\mathtt{K}$ as in (6.10–*p*157) are

$$\boldsymbol{\omega}^* = \begin{bmatrix} \alpha_x^2 + s^2 + x_0^2 & s\alpha_y + x_0 y_0 & x_0 \\ s\alpha_y + x_0 y_0 & \alpha_y^2 + y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix} \tag{19.9}$$

and

$$\boldsymbol{\omega} = \tfrac{1}{\alpha_x^2 \alpha_y^2} \begin{bmatrix} \alpha_y^2 & -s\alpha_y & -x_0\alpha_y^2 + y_0 s\alpha_y \\ -s\alpha_y & \alpha_x^2 + s^2 & \alpha_y s x_0 - \alpha_x^2 y_0 - s^2 y_0 \\ -x_0\alpha_y^2 + y_0 s\alpha_y & \alpha_y s x_0 - \alpha_x^2 y_0 - s^2 y_0 & \alpha_x^2\alpha_y^2 + \alpha_x^2 y_0^2 + (\alpha_y x_0 - s y_0)^2 \end{bmatrix} \tag{19.10}$$

If the skew is zero, i.e. $s = 0$, then the expressions simplify to

$$\boldsymbol{\omega}^* = \begin{bmatrix} \alpha_x^2 + x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & \alpha_y^2 + y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix} \tag{19.11}$$

and

$$\boldsymbol{\omega} = \frac{1}{\alpha_x^2 \alpha_y^2} \begin{bmatrix} \alpha_y^2 & 0 & -\alpha_y^2 x_0 \\ 0 & \alpha_x^2 & -\alpha_x^2 y_0 \\ -\alpha_y^2 x_0 & -\alpha_x^2 y_0 & \alpha_x^2\alpha_y^2 + \alpha_y^2 x_0^2 + \alpha_x^2 y_0^2 \end{bmatrix} \tag{19.12}$$

Table 19.1. *The image of the absolute conic, $\boldsymbol{\omega}$, and dual image of the absolute conic, $\boldsymbol{\omega}^*$, written in terms of the camera internal parameters.*

We have seen that in a projective frame $\mathtt{Q}_\infty^*$ has the form $\mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T}$. The projective reconstruction is related to the metric reconstruction by (19.2), so in detail

$$\mathtt{Q}_\infty^* = \mathtt{H}\tilde{\mathtt{I}}\mathtt{H}^\mathsf{T} = \begin{bmatrix} \mathtt{K}^1\mathtt{K}^{1\mathsf{T}} & -\mathtt{K}^1\mathtt{K}^{1\mathsf{T}}\mathbf{p} \\ -\mathbf{p}^\mathsf{T}\mathtt{K}^1\mathtt{K}^{1\mathsf{T}} & \mathbf{p}^\mathsf{T}\mathtt{K}^1\mathtt{K}^{1\mathsf{T}}\mathbf{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}^{*1} & -\boldsymbol{\omega}^{*1}\mathbf{p} \\ -\mathbf{p}^\mathsf{T}\boldsymbol{\omega}^{*1} & \mathbf{p}^\mathsf{T}\boldsymbol{\omega}^{*1}\mathbf{p} \end{bmatrix}. \tag{19.8}$$

On applying (19.6) with $\mathtt{P}^i = [\mathtt{A}^i \mid \mathbf{a}^i]$ we obtain once again the auto-calibration equations (19.4)

$$\boldsymbol{\omega}^{*i} = \mathtt{P}^i \mathtt{Q}_\infty^* \mathtt{P}^{i\mathsf{T}} = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right) \boldsymbol{\omega}^{*1} \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)^\mathsf{T}.$$

This is a geometric interpretation of (19.4) – $\mathtt{Q}_\infty^*$ is a fixed quadric under the Euclidean motion of the camera, and the DIAC $\boldsymbol{\omega}^{*i}$ is the image of $\mathtt{Q}_\infty^*$ in each view.

19.3.1 Linear solutions for $\mathtt{Q}_\infty^*$ from a set of images

The objective here is to estimate $\mathtt{Q}_\infty^*$ in a projective reconstruction directly from constraints on the internal parameters. We will start by describing three cases for which a linear solution may be obtained. It is convenient at this point to summarize the forms of the DIAC and also the IAC. Refer to table 19.1.

Specifying linear constraints on $\mathtt{Q}_\infty^*$. Linear constraints on $\mathtt{Q}_\infty^*$ may be obtained if the principal point is known. Assume that this point is known, then we may change

the image coordinate system so that the origin coincides with the principal point. Then $x_0 = 0, y_0 = 0$, and from table 19.1 the DIAC becomes

$$\boldsymbol{\omega}^* = \begin{bmatrix} \alpha_x^2 + s^2 & s\alpha_y & 0 \\ s\alpha_y & \alpha_y^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19.13}$$

The linear equations on $\mathtt{Q}_\infty^*$ are then generated from the zero entries in (19.13) by applying the projection equation (19.6) $\boldsymbol{\omega}^* = \mathtt{P}\mathtt{Q}_\infty^*\mathtt{P}^\mathsf{T}$ to each view i. For example the two equations

$$(\mathtt{P}^i\mathtt{Q}_\infty^*\mathtt{P}^{i\mathsf{T}})_{13} = 0 \;\text{ and }\; (\mathtt{P}^i\mathtt{Q}_\infty^*\mathtt{P}^{i\mathsf{T}})_{23} = 0 \tag{19.14}$$

follow immediately from ${\boldsymbol{\omega}^*}^i_{13} = {\boldsymbol{\omega}^*}^i_{23} = 0$.

If there are additional constraints on $\mathtt{K}^i$ which result in further relationship between the entries of $\boldsymbol{\omega}^*$, then these may provide additional linear equations. For instance, an assumption that skew is zero means that the (1,2)-entries of (19.13) vanish, which provides one more linear equation on the entries of $\mathtt{Q}_\infty^*$ similar to (19.14). Known aspect ratio provides a further constraint. Table 19.2 summarizes the possible constraints that may be used.

Condition	constraint	type	# constraints
zero skew	$\boldsymbol{\omega}^*_{12}\boldsymbol{\omega}^*_{33} = \boldsymbol{\omega}^*_{13}\boldsymbol{\omega}^*_{23}$	quadratic	m
principal point (p.p.) at origin	$\boldsymbol{\omega}^*_{13} = \boldsymbol{\omega}^*_{23} = 0$	linear	$2m$
zero skew (p.p. at origin)	$\boldsymbol{\omega}^*_{12} = 0$	linear	m
fixed (unknown) aspect ratio (zero skew and p.p. at origin)	$\frac{{\boldsymbol{\omega}^*}^i_{11}}{{\boldsymbol{\omega}^*}^i_{22}} = \frac{{\boldsymbol{\omega}^*}^j_{11}}{{\boldsymbol{\omega}^*}^j_{22}}$	quadratic	$m-1$
known aspect ratio $r = \alpha_y/\alpha_x$ (zero skew and p.p. at origin)	$r^2\boldsymbol{\omega}^*_{11} = \boldsymbol{\omega}^*_{22}$	linear	m

Table 19.2. **Auto-calibration constraints derived from the DIAC**. *The number of constraints column gives the total number of constraints over m views, assuming the constraint is true for each view. Each additional item of information generates additional equations. For example, if the principal point is known and skew is zero then there are 3 constraints per view.*

Linear solution. Since it is symmetric, $\mathtt{Q}_\infty^*$ may be parametrized linearly by 10 homogeneous parameters, namely the 10 diagonal and above-diagonal entries. These 10 entries may be represented by a 10-vector $\mathbf{x}$. In the usual manner the linear equations on $\mathtt{Q}_\infty^*$ may be assembled into a matrix equation of the form $\mathtt{A}\mathbf{x} = \mathbf{0}$, and a least-squares solution for $\mathbf{x}$ obtained via the SVD. For example, the two equations (19.14) provide two rows of the matrix from each view. From five images a total of 10 equations are obtained (assuming only that principal point is known), and a linear solution is possible. From four images eight equations are generated. In the same way as with the

computation of the fundamental matrix from seven points there is a 2-parameter family of solutions. The condition $\det \mathtt{Q}^*_\infty = 0$ gives a fourth-degree equation and so up to four solutions for $\mathtt{Q}^*_\infty$.

Example 19.5. Linear solution for variable focal length
Suppose the camera is calibrated apart from the focal length – the principal point is known, the aspect ratio is unity (if it is not the equations can be transformed so that it is unity from the known value), and skew is zero – the focal length is unknown and may vary between views. In this case from table 19.2 there are four linear constraints on $\mathtt{Q}^*_\infty$ available from each view. In the case of two views there are eight linear constraints and up to four solutions are obtained using the condition $\det \mathtt{Q}^*_\infty = 0$. If $m \geq 3$ a unique linear solution exists. △

More will be said about determination of focal lengths in this minimal case in example 19.8.

19.3.2 Non-linear solutions for $\mathtt{Q}^*_\infty$

We now describe various non-linear equations that can be obtained from the form of (19.6). It has been seen that each element of $\boldsymbol{\omega}^{*i} = \mathtt{P}^i \mathtt{Q}^*_\infty \mathtt{P}^{i\mathsf{T}}$ is expressible as a linear expression in terms of the parameters of $\mathtt{Q}^*_\infty$. It follows that any relationship between the entries of the various $\boldsymbol{\omega}^{*i}$ translates into an equation involving the entries of $\mathtt{Q}^*_\infty$. In particular, linear or quadratic relationships between entries of $\boldsymbol{\omega}^*$ generate respectively linear or quadratic relationships between entries of $\mathtt{Q}^*_\infty$. Given sufficiently many such equations, we may solve for $\mathtt{Q}^*_\infty$.

Constant internal parameters. If the internal parameters of all cameras are the same, then $\boldsymbol{\omega}^{*i} = \boldsymbol{\omega}^{*j}$ for all i and j, which expands to $\mathtt{P}^i \mathtt{Q}^*_\infty \mathtt{P}^{i\mathsf{T}} = \mathtt{P}^j \mathtt{Q}^*_\infty \mathtt{P}^{j\mathsf{T}}$. However, since these are homogeneous quantities, the equality holds only up to an unknown scale. A set of five equations are generated:

$$\boldsymbol{\omega}^{*i}_{11}/\boldsymbol{\omega}^{*j}_{11} = \boldsymbol{\omega}^{*i}_{12}/\boldsymbol{\omega}^{*j}_{12} = \boldsymbol{\omega}^{*i}_{13}/\boldsymbol{\omega}^{*j}_{13} = \boldsymbol{\omega}^{*i}_{22}/\boldsymbol{\omega}^{*j}_{22} = \boldsymbol{\omega}^{*i}_{23}/\boldsymbol{\omega}^{*j}_{23} = \boldsymbol{\omega}^{*i}_{33}/\boldsymbol{\omega}^{*j}_{33}.$$

This gives a set of quadratic equations in the entries of $\mathtt{Q}^*_\infty$. Given three views, a total of 10 equations result, which may be solved to find $\mathtt{Q}^*_\infty$.

Calibration assuming zero skew. Under the assumption of zero skew in each of the cameras, the form of the DIAC is simplified, as given in (19.11). In particular, we obtain the following constraints between the entries of $\boldsymbol{\omega}^*$ in the zero-skew case

$$\boldsymbol{\omega}^*_{12}\boldsymbol{\omega}^*_{33} = \boldsymbol{\omega}^*_{13}\boldsymbol{\omega}^*_{23}. \tag{19.15}$$

This gives a single quadratic equation in the entries of $\mathtt{Q}^*_\infty$. From a set of m views we obtain m quadratics. However there is also one extra equation $\det \mathtt{Q}^*_\infty = 0$ derived from the fact that the absolute dual quadric is degenerate. Since $\mathtt{Q}^*_\infty$ has 10 homogeneous linear parameters, it may be computed (at least in principle) from 8 views.

These different calibration constraints are also summarized in table 19.2.

19.3.3 Iterative methods

As we have seen on many occasions throughout this book there is a choice between minimizing an algebraic or geometric error. In this case a suitable algebraic error is provided by (19.4) In previous cases, such as (4.1–*p*89), the unknown scale factor has been eliminated by forming a cross product. Here the scale factor can be eliminated by using a matrix norm. The cost function is

$$\sum_i \|\mathtt{K}^i\mathtt{K}^{i\mathsf{T}} - \mathtt{P}^i\mathtt{Q}^*_\infty\mathtt{P}^{i\mathsf{T}}\|^2_{\mathrm{F}} \tag{19.16}$$

where $\|\mathtt{M}\|_{\mathrm{F}}$ is the Frobenius norm of $\mathtt{M}$, and $\mathtt{K}^i\mathtt{K}^{i\mathsf{T}}$ and $\mathtt{P}^i\mathtt{Q}^*_\infty\mathtt{P}^{i\mathsf{T}}$ are both normalized to have unit Frobenius norm. The cost function is parametrized by the (at most eight) unknown elements of $\mathtt{Q}^*_\infty$, and the unknown elements of each $\boldsymbol{\omega}^{*i} = \mathtt{K}^i\mathtt{K}^{i\mathsf{T}}$. It is possible to use the expansion (19.8) to parametrize the absolute dual quadric. For example, in the case of example 19.5 where the focal length is the only unknown per view, (19.16) would be minimized over $m+3$ parameters. These are the focal length f^i of each view, and the three components of $\mathbf{p}$. Note that this parametrization ensures that $\mathtt{Q}^*_\infty$ has rank 3 throughout the minimization.

Since the above cost function has no particular geometric meaning, it is advisable to follow this up with a complete bundle adjustment. In fact, given a good initial linear estimate one can proceed directly to bundle adjustment. There is no difficulty in incorporating assumptions on calibration parameters into a full bundle adjustment as described in section 18.1(*p*434).

Example 19.6. Metric reconstruction for general motion
Figure 19.1(a-c) shows views of an Indian temple acquired by a hand held camera. A projective reconstruction is computed from image point correspondences as described in section 18.6, and a metric reconstruction obtained using algorithm 19.1 under the constraint of constant camera parameters with known principal point. The computed cameras and 3D point cloud are shown in figure 19.1(d) and (e). △

19.3.4 A counting argument

The constraints we have seen have been of two types: a parameter has a known value; or a parameter is fixed across views but its value is unknown. The actual constraints that apply depend on the physical circumstances of the image production from acquisition by the camera, through digitization and cropping, to the final image. For example, for an image sequence in which the lens is zoomed it might be the case that the skew and aspect ratio are fixed (but unknown), but that the focal length and principal point vary through the sequence. Often it is the case that the pixels are square or have a known aspect ratio, so that both the skew (which is zero) and aspect ratio are known.

We will now consider the number of constraints that are required to determine a metric reconstruction fully.

The number of parameters that must be computed to perform calibration is 8. This is equal to the number of essential parameters of the absolute dual quadric, including the scale ambiguity and rank-3 constraint. Consider m views and suppose that k of the

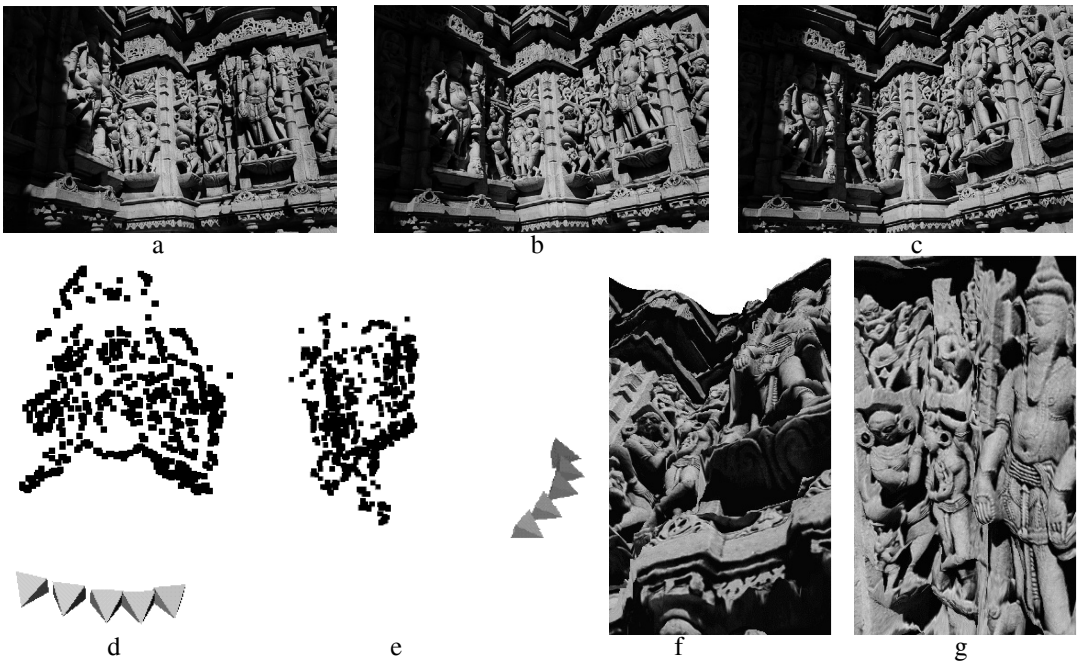

Fig. 19.1. **Metric reconstruction for general motion**. *(a)-(c) 3 views (of 5) acquired by a hand held camera. (d) and (e) Two views of a metric reconstruction computed from interest points matches over the five views. The cameras are represented by pyramids with apex at the computed camera centre. (f) and (g) two views of a texture mapped 3D model computed from the original images and reconstructed cameras using an area based stereo algorithm. Figures courtesy of Marc Pollefeys, Reinhard Koch, and Luc Van Gool.*

internal parameters are known in all views, and f are fixed over the views but unknown (where $k + f \leq 5$). A fixed and known calibration parameter provides one constraint per view via the condition $\boldsymbol{\omega}^{*i} = \mathtt{P}^i \mathtt{Q}^*_\infty \mathtt{P}^{i\mathsf{T}}$, for a total of mk constraints. A fixed but unknown calibration parameter provides one fewer constraint, since just the value of the unknown parameter is missing. Thus f fixed parameters provide a total of $f(m-1)$ constraints. The requirement for calibration is then that

$$mk + (m-1)f \geq 8.$$

Table 19.3 gives values for m for several combinations of constraints. It is important to remember that degenerate configurations are of course possible in which some of the constraints are dependent. This will increase the number of required views.

19.3.5 Limitations of the absolute quadric approach to calibration

The following considerations apply to calibration using this method.

Limitations of least-squares algebraic solution. Since the least-squares solution (e.g. of $\mathtt{A}\mathbf{x} = \mathbf{0}$ in the linear solution for the $\boldsymbol{\omega}^*$) minimizes but does not enforce constraints, the solution obtained will not precisely satisfy the required conditions. This observation holds in over-constrained cases. For instance in the case of estimating focal lengths in example 19.5, the entries ${\boldsymbol{\omega}^*}^i_{11}$ and ${\boldsymbol{\omega}^*}^i_{22}$ will not be in the required

Condition	fixed f	known k	views m
Constant internal parameters	5	0	3
Aspect ratio and skew known, focal length and principal point vary	0	2	4*
Aspect ratio and skew constant, focal length and principal point vary	2	0	5*
Skew zero, all other parameters vary	0	1	8*
p.p. known all other parameters vary	0	2	4*, 5(linear)
p.p. known skew zero	0	3	3(linear)
p.p., skew and aspect ratio known	0	4	2, 3(linear)

Table 19.3. *The number of views m required under various conditions in order for there to be enough constraints for auto-calibration. For those cases marked with an asterisk there may be multiple solutions, even for general motion between views.*

ratio, nor will the off-diagonal entries be precisely zero. This means that the $\mathtt{K}^i$ will not be precisely of the desired form. The absolute dual quadric computed by linear means will not have rank 3 in general, since this is not enforced by the linear equations. A rank 3 matrix for $\mathtt{Q}^*_\infty$ can be obtained by setting the smallest eigenvalue to zero in its eigenvalue decomposition (in a similar manner to using the SVD to obtain a rank 2 matrix for $\mathtt{F}$ in the 8-point algorithm in section 11.1.1(*p*280)). This rank 3 matrix may then be directly decomposed to obtain the rectifying homography (19.2) using result 19.4. Alternatively, the rank 3 matrix can provide the starting point for an iterative minimization as described in section 19.3.3.

The positive-definiteness condition. The most troublesome failing of this method is the difficulty in enforcing the condition that $\mathtt{Q}^*_\infty$ is positive semi-definite, (or negative semi-definite if the sign is reversed). This is related to the condition that $\boldsymbol{\omega}^* = \mathtt{P}\mathtt{Q}^*_\infty\mathtt{P}^\mathsf{T}$ should be positive-definite. If $\boldsymbol{\omega}^*$ is not positive-definite, then it can not be decomposed using Cholesky factorization to compute the calibration matrix. This is a recurring problem with auto-calibration methods based on estimation of the IAC or DIAC. If the data is noisy, then this problem may occur indicating that the data are not consistent with metric reconstruction. It is not appropriate if this occurs to seek the closest positive-definite solution, since this will generally be a boundary case leading to a spurious calibration.

19.4 The Kruppa equations

A different method of auto-calibration involves the use of the Kruppa equations, which were originally introduced into computer vision by Faugeras, Luong and Maybank [Faugeras-92a] and historically are seen as the first auto-calibration method. They

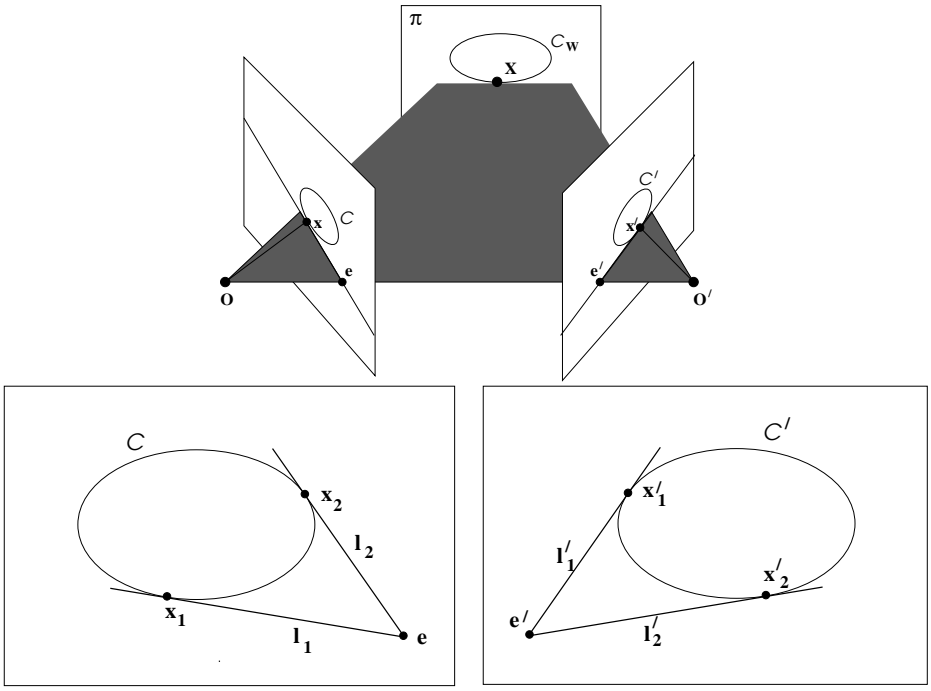

Fig. 19.2. **Epipolar tangency for a conic.** *A conic* $\mathtt{C}_\mathrm{w}$ *on a world plane* $\boldsymbol{\pi}$ *is imaged as corresponding conics* $\mathtt{C} \leftrightarrow \mathtt{C}'$ *in two views. The imaged conics are consistent with the epipolar geometry of the view pair. Upper: An epipolar plane tangent to the world conic* $\mathtt{C}_\mathrm{w}$ *defines corresponding epipolar lines which are tangent to the imaged conics. Lower: epipolar lines,* $\mathbf{l}_1, \mathbf{l}_2$ *tangent to the imaged conic in the first view correspond to the epipolar lines* $\mathbf{l}'_1, \mathbf{l}'_2$*, respectively, tangent to the imaged conic in the second.*

are two-view constraints that require only $\mathtt{F}$ to be known, and consist of two independent quadratic equations in the elements of $\boldsymbol{\omega}^*$.

The Kruppa equations are an algebraic representation of the correspondence of epipolar lines tangent to a conic. The geometry of this correspondence is illustrated in figure 19.2. Suppose the conics $\mathtt{C}$ and $\mathtt{C}'$ are the images of a conic $\mathtt{C}_\mathrm{w}$ on a world plane in the first and second views respectively, and that $\mathtt{C}^*$ and $\mathtt{C}^{*\prime}$ are their duals. In the first view the two epipolar tangent lines $\mathbf{l}_1$ and $\mathbf{l}_2$ may be combined into a single degenerate point conic (see example 2.8(p32)) as $\mathtt{C}_\mathrm{t} = [\mathbf{e}]_\times \mathtt{C}^* [\mathbf{e}]_\times$. (It may be verified that any point $\mathbf{x}$ on the lines $\mathbf{l}_1$ and $\mathbf{l}_2$ satisfies $\mathbf{x}^\mathsf{T}\mathtt{C}_\mathrm{t}\mathbf{x} = 0$). Similarly, in the second view the corresponding epipolar lines $\mathbf{l}'_1$ and $\mathbf{l}'_2$ may be written as $\mathtt{C}'_\mathrm{t} = [\mathbf{e}']_\times \mathtt{C}^{*\prime} [\mathbf{e}']_\times$. The epipolar tangent lines correspond under the homography $\mathtt{H}$ induced by any world plane $\boldsymbol{\pi}$. Since $\mathtt{C}_\mathrm{t}$ is a point conic it transforms according to result 2.13(p37) $\mathtt{C}'_\mathrm{t} = \mathtt{H}^{-\mathsf{T}} \mathtt{C}_\mathrm{t} \mathtt{H}^{-1}$, and the correspondence of the lines requires that

$$\begin{aligned} [\mathbf{e}']_\times \mathtt{C}^{*\prime} [\mathbf{e}']_\times &= \mathtt{H}^{-\mathsf{T}} [\mathbf{e}]_\times \mathtt{C}^* [\mathbf{e}]_\times \mathtt{H}^{-1} \\ &= \mathtt{F}\mathtt{C}^*\mathtt{F}^\mathsf{T} \end{aligned} \tag{19.17}$$

the last equality following from $\mathtt{F} = \mathtt{H}^{-\mathsf{T}}[\mathbf{e}]_\times$ (see page 335). Note, this equation does not enforce the condition that the epipolar tangent lines map individually to their corresponding lines, only that their symmetric product maps to their symmetric product.

The development to this point applies to any conic. However, in the case of interest

here the world conic is the absolute conic on the plane at infinity, so that $\mathtt{C}^* = \boldsymbol{\omega}^*$, $\mathtt{C}^{*\prime} = \boldsymbol{\omega}^{*\prime}$, (and $\mathtt{H} = \mathtt{H}_\infty$), and (19.17) specializes to

$$[\mathbf{e}']_\times \boldsymbol{\omega}^{*\prime} [\mathbf{e}']_\times = \mathtt{F}\boldsymbol{\omega}^*\mathtt{F}^\mathsf{T} \tag{19.18}$$

If the internal parameters are constant over the views then $\boldsymbol{\omega}^{*\prime} = \boldsymbol{\omega}^*$ and so $[\mathbf{e}']_\times \boldsymbol{\omega}^* [\mathbf{e}']_\times = \mathtt{F}\boldsymbol{\omega}^*\mathtt{F}^\mathsf{T}$, which are the Kruppa equations in a form originally given by Viéville [Vieville-95]. On eliminating the homogeneous scale factor, one obtains equations quadratic in the elements of $\boldsymbol{\omega}^*$.

Although (19.18) concisely expresses the Kruppa equations, it is not in a form that can be easily applied. A succinct and easily usable form of the Kruppa equations is now given. We show that the null-space of $[\mathbf{e}']_\times$, which is common to both sides of (19.18), can be eliminated leaving an equation between two 3-vectors.

Result 19.7. *The Kruppa equations (19.18) are equivalent to*

$$\begin{pmatrix} \mathbf{u}_2^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2 \\ -\mathbf{u}_1^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2 \\ \mathbf{u}_1^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_1 \end{pmatrix} \times \begin{pmatrix} \sigma_1^2\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_1 \\ \sigma_1\sigma_2\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_2 \\ \sigma_2^2\mathbf{v}_2^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_2 \end{pmatrix} = \mathbf{0} \tag{19.19}$$

where $\mathbf{u}_i$, $\mathbf{v}_i$ and σ_i are the columns and singular values of the SVD of $\mathtt{F}$. This provides three quadratic equations in the elements $\boldsymbol{\omega}^*_{ij}$ of $\boldsymbol{\omega}^*$, of which two are independent.

Proof. The fundamental matrix has rank 2, and thus has an SVD expansion

$$\mathtt{F} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T} = \mathtt{U} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \mathtt{V}^\mathsf{T}$$

where the null-vectors are $\mathtt{F}^\mathsf{T}\mathbf{u}_3 = \mathbf{0}$ and $\mathtt{F}\mathbf{v}_3 = \mathbf{0}$. This means that the epipoles are $\mathbf{e} = \mathbf{v}_3$ and $\mathbf{e}' = \mathbf{u}_3$. Substituting this expansion into (19.18) we obtain

$$[\mathbf{u}_3]_\times \boldsymbol{\omega}^{*\prime} [\mathbf{u}_3]_\times = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}\boldsymbol{\omega}^*\mathtt{V}\mathtt{D}\mathtt{U}^\mathsf{T}. \tag{19.20}$$

We now use the property that $\mathtt{U}$ is an orthogonal matrix. On pre-multiplying (19.20) by $\mathtt{U}^\mathsf{T}$, and post-multiplying by $\mathtt{U}$, the LHS becomes

$$\begin{aligned} \mathtt{U}^\mathsf{T}[\mathbf{u}_3]_\times \boldsymbol{\omega}^{*\prime} [\mathbf{u}_3]_\times \mathtt{U} &= \begin{bmatrix} \mathbf{u}_2 & -\mathbf{u}_1 & \mathbf{0} \end{bmatrix}^\mathsf{T} \boldsymbol{\omega}^{*\prime} \begin{bmatrix} \mathbf{u}_2 & -\mathbf{u}_1 & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{u}_2^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2 & -\mathbf{u}_2^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_1 & 0 \\ -\mathbf{u}_1^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2 & \mathbf{u}_1^\mathsf{T}\boldsymbol{\omega}^{*\prime}\mathbf{u}_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

and the RHS of (19.20) becomes

$$\begin{aligned} \mathtt{D}\mathtt{V}^\mathsf{T}\boldsymbol{\omega}^*\mathtt{V}\mathtt{D} &= \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \mathtt{V}^\mathsf{T}\boldsymbol{\omega}^*\mathtt{V} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_1 & \sigma_1\sigma_2\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_2 & 0 \\ \sigma_1\sigma_2\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_2 & \sigma_2^2\mathbf{v}_2^\mathsf{T}\boldsymbol{\omega}^*\mathbf{v}_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

It is evident that both sides have reduced to symmetric matrices each with three independent elements. These three elements may be represented by a homogeneous 3-vector on each side:

$$\begin{aligned}\mathbf{x}_{\mathrm{LHS}}^{\mathsf{T}} &= \left(\ \mathbf{u}_2^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2,\quad -\mathbf{u}_1^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2,\quad \mathbf{u}_1^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_1\ \right)\\ \mathbf{x}_{\mathrm{RHS}}^{\mathsf{T}} &= \left(\ \sigma_1^2\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_1,\quad \sigma_1\sigma_2\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_2,\quad \sigma_2^2\mathbf{v}_2^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_2\ \right).\end{aligned}$$

The two sides are only equal up to a scale factor. However, equalities are obtained in the usual way using a vector cross-product, $\mathbf{x}_{\mathrm{LHS}} \times \mathbf{x}_{\mathrm{RHS}} = \mathbf{0}$. An alternative derivation is given in [Hartley-97d]. □

Note, the Kruppa equations involve the DIAC, rather than the IAC, since they arise from tangent line constraints, and line constraints are more simply expressed using a dual (line) conic.

We now discuss the solution of the Kruppa equations, beginning with a simple example where all the internal parameters are known apart from the focal length. An alternative method for solving this problem using the absolute dual quadric was given in example 19.5.

Example 19.8. Focal lengths for a view pair

Suppose two cameras have zero skew and known principal point and aspect ratio, but unknown and different focal lengths (as in example 19.5). Then from (19.11) by a suitable change of coordinates their DIACs may be written as

$$\boldsymbol{\omega}^* = \mathrm{diag}(\alpha^2, \alpha^2, 1),\quad \boldsymbol{\omega}^{*\prime} = \mathrm{diag}(\alpha'^2, \alpha'^2, 1)$$

where α, α' are the unknown focal lengths of the first and second view, respectively. Writing the Kruppa equations (19.19) as

$$\frac{\mathbf{u}_2^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2}{\sigma_1^2\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_1} = -\frac{\mathbf{u}_1^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_2}{\sigma_1\sigma_2\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_2} = \frac{\mathbf{u}_1^{\mathsf{T}}\boldsymbol{\omega}^{*\prime}\mathbf{u}_1}{\sigma_2^2\mathbf{v}_2^{\mathsf{T}}\boldsymbol{\omega}^{*}\mathbf{v}_2}$$

it is evident that each numerator is linear in terms of α'^2 and each denominator linear in α^2. Cross-multiplying provides two simple quadratic equations in α^2 and α'^2 which are easily solved. The values of α and α' are found by taking the square roots. Note, if the internal parameters are the same for the two views (that is $\alpha = \alpha'$) then each equation of result 19.7 provides a quadratic in the single unknown α^2. △

Extending the Kruppa equations to multiple views. In the absence of knowledge of the internal parameters, other than that the parameters are constant across views, the Kruppa equations provide two independent constraints on the five unknown parameters. Thus given three views, with F known between each pair, there are in principle six quadratic constraints, which is sufficient to determine $\boldsymbol{\omega}^*$. Using any five of these equations results in five quadratics in five unknowns, a total of 2^5 possible solutions. Solving this set of equations is not a particularly promising approach, although solutions have been obtained by homotopy continuation [Luong-92] and by minimizing algebraic residuals using every view pair for a sequence of images [Zeller-96].

Ambiguities. If there is no rotation between views then the Kruppa equations provide no constraint on $\boldsymbol{\omega}^*$. This may be seen from (19.18) which, in the case of pure translation, reduces to $[\mathbf{e}']_\times \boldsymbol{\omega}^* [\mathbf{e}']_\times = [\mathbf{e}']_\times \boldsymbol{\omega}^* [\mathbf{e}']_\times$, since $\mathtt{F} = [\mathbf{e}']_\times$.

The Kruppa equations are closely related to the calibration constraint provided by the transfer of the IAC under the infinite homography, as discussed later in section 19.5.2. It will follow from that discussion that the constraint placed on $\boldsymbol{\omega}^*$ by the Kruppa equations for a pair of views is weaker than that placed by the infinite homography constraint (19.25). Consequently ambiguities of $\boldsymbol{\omega}^*$ imposed by the Kruppa equations are a superset of those imposed by (19.25).

The application of the Kruppa equations to three or more views provides weaker constraints than those obtained by other methods such as the modulus constraint (section 19.5.1) or the absolute dual quadric (section 19.3.1). This is because the Kruppa constraints are a view-pair constraint for *conics* obtained as a projection of a 3D (dual) *quadric*. They do not enforce that the (dual) quadric is degenerate, or equivalently do not enforce a common support plane for Ω_∞ over the multiple views. Consequently, there are additional ambiguous solutions as described in [Sturm-97b].

Although the application of the Kruppa equations to auto-calibration is chronologically the first example in the literature, the difficulty of their solution and the problem with ambiguities has seen them losing favour in the face of more tractable methods such as the dual quadric formulation. However, if only two views are given then the Kruppa equations are *the* constraint available on $\boldsymbol{\omega}^*$.

19.5 A stratified solution

Determining a metric reconstruction involves simultaneously obtaining both the camera calibration $\mathtt{K}$ and the plane at infinity, $\boldsymbol{\pi}_\infty$. An alternative approach is first to obtain by some means $\boldsymbol{\pi}_\infty$, or equivalently an affine reconstruction. The subsequent determination of $\mathtt{K}$ is then relatively simple because there exists a linear solution. This approach will now be described starting with methods of determining $\boldsymbol{\pi}_\infty$, and hence $\mathtt{H}_\infty$, and followed by methods of computing $\mathtt{K}$ given $\mathtt{H}_\infty$.

19.5.1 Affine reconstruction – determining $\boldsymbol{\pi}_\infty$

For general motion and constant internal parameters, (19.3–*p*461) can be rearranged into providing a constraint only on $\boldsymbol{\pi}_\infty$, known as the *modulus constraint*. This allows the coordinates $\mathbf{p}$ of $\boldsymbol{\pi}_\infty$ to be solved for directly, and is described below.

The modulus constraint

The modulus constraint is a polynomial equation in the coordinates of $\boldsymbol{\pi}_\infty$. Assume the internal parameters are constant; then from (19.3–*p*461) with $\mathtt{K}^i = \mathtt{K}$

$$\mathtt{A} - \mathbf{a}\mathbf{p}^\mathsf{T} = \mu \mathtt{K}\mathtt{R}\mathtt{K}^{-1} \tag{19.21}$$

where the scale factor μ is explicitly included, and for clarity the superscripts are omitted. Since $\mathtt{K}\mathtt{R}\mathtt{K}^{-1}$ is conjugate to a rotation, it has eigenvalues $\{1, e^{i\theta}, e^{-i\theta}\}$. Consequently, the eigenvalues of $(\mathtt{A} - \mathbf{a}\mathbf{p}^\mathsf{T})$ are $\{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\}$, and thus have equal moduli. This is the modulus constraint on the coordinates of the plane at infinity, $\mathbf{p}$.

To develop this constraint further consider the characteristic polynomial of $\mathtt{A} - \mathbf{a}\mathbf{p}^\mathsf{T}$ which is

$$\begin{aligned}\det(\lambda\mathtt{I} - \mathtt{A} + \mathbf{a}\mathbf{p}^\mathsf{T}) &= (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \\ &= \lambda^3 - f_1\lambda^2 + f_2\lambda - f_3\end{aligned}$$

where λ_i are the three eigenvalues, and

$$\begin{aligned}f_1 &= \lambda_1 + \lambda_2 + \lambda_3 = \mu(1 + 2\cos\theta) \\ f_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 = \mu^2(1 + 2\cos\theta) \\ f_3 &= \lambda_1\lambda_2\lambda_3 = \mu^3.\end{aligned}$$

Eliminating the scalar μ and angle θ we obtain

$$f_3 f_1^3 = f_2^3.$$

Looking more closely at the characteristic polynomial we observe that $\mathbf{p}$ appears only as part of the rank-1 term $\mathbf{a}\mathbf{p}^\mathsf{T}$. This means that the elements of $\mathbf{p}$ appear only linearly in the determinant $\det(\lambda\mathtt{I} - \mathtt{A} + \mathbf{a}\mathbf{p}^\mathsf{T})$, and hence linearly in each of f_1, f_2, f_3. Hence the modulus constraint may be written as a *quartic* polynomial in the three elements p_i of $\mathbf{p}$. This polynomial equation is only a necessary condition for the eigenvalues to have equal moduli, not sufficient.

Each view pair generates a quartic equation in the coordinates of $\boldsymbol{\pi}_\infty$. Thus, in principle, three views determine $\boldsymbol{\pi}_\infty$, but only as the intersection of three quartics in three variables – a possible 4^3 solutions. However, for three views an additional cubic equation is available from the modulus constraint, and this equation can be used to eliminate many of the spurious solutions. This cubic equation is developed in [Schaffalitzky-00a]. The modulus constraint may also be combined with scene information. For example, if a corresponding vanishing line is available in two views, then $\boldsymbol{\pi}_\infty$ is determined up to a one-parameter ambiguity. Applying the modulus constraint resolves this ambiguity, and results in a quartic equation in one variable.

The modulus constraint may be considered the cousin of the Kruppa equations: the Kruppa equations are equations on $\boldsymbol{\omega}^*$ which do not involve $\boldsymbol{\pi}_\infty$; conversely, the modulus constraint is an equation on $\boldsymbol{\pi}_\infty$ which does not involve $\boldsymbol{\omega}^*$. Once one of $\boldsymbol{\omega}^*$ or $\boldsymbol{\pi}_\infty$ is known the other can subsequently be determined.

Other methods of finding $\boldsymbol{\pi}_\infty$

Because of the problem with solving sets of simultaneous quartic equations, the modulus constraint is not very satisfactory as a practical means of finding the plane at infinity. In fact, finding the plane at infinity is the hardest part of auto-calibration and the place where one is most likely to run into difficulties.

The plane at infinity may be identified by various other methods. Several of these are described in chapter 10. One straightforward method (which is outside the province of pure auto-calibration) is to use properties of the scene geometry. For example, the correspondence of a vanishing point between two views determines a point on $\boldsymbol{\pi}_\infty$, and three such correspondences determine $\boldsymbol{\pi}_\infty$ in a projective reconstruction. Indeed,

a good approximation for $\boldsymbol{\pi}_\infty$ may well be obtained from the correspondence of three points that are distant in the scene. A second method is to employ a pure translation between two views, i.e. the camera translates but does not rotate or change internal parameters; $\boldsymbol{\pi}_\infty$ is determined uniquely by such a motion.

As seen in result 19.3(*p*463) the plane at infinity may also be computed from the absolute dual quadric and this method is quite attractive if the principal point is known. Methods of bounding the position of $\boldsymbol{\pi}_\infty$ using cheiral inequalities will be described in chapter 21. These use information about which points are in front of the cameras to get a *quasi-affine* reconstruction, which is close to an affine reconstruction. A method for finding the plane at infinity by iterative search from this initial quasi-affine reconstruction is described in [Hartley-94b]. More recently, the bounds imposed by cheirality have been used [Hartley-99] to define a rectangular region of 3D parameter space inside which the vector $\mathbf{p}$ representing the plane at infinity must lie. Then an exhaustive search is undertaken to find the elusive plane at infinity inside this region.

19.5.2 Affine to metric conversion – determining $\mathtt{K}$ given $\boldsymbol{\pi}_\infty$

Once the plane at infinity has been determined, an affine reconstruction is effectively known. The remaining step is to transform from affine to metric. It turns out that this is a far easier step than the step from projective to affine. In fact, a linear algorithm is available based on the transformation of the IAC or its dual.

The infinite homography. The infinite homography $\mathtt{H}_\infty$ is the plane projective transformation between two images induced by the plane at infinity $\boldsymbol{\pi}_\infty$ (see chapter 13). If the plane at infinity $\boldsymbol{\pi}_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$ and camera matrices $[\mathtt{A}^i \mid \mathbf{a}^i]$ are known in any projective coordinate frame, an explicit formula for the infinite homography can be derived from result 13.1(*p*326)

$$\mathtt{H}^i_\infty = \mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T} \tag{19.22}$$

where $\mathtt{H}^i_\infty$ represents the homography from a camera $[\mathtt{I} \mid \mathbf{0}]$ to the camera $[\mathtt{A}^i \mid \mathbf{a}^i]$. So $\mathtt{H}^i_\infty$ may be computed from a projective reconstruction once the plane at infinity is known.

If the first camera is not in the canonical form $[\mathtt{I} \mid \mathbf{0}]$, then one can still compute the homography $\mathtt{H}^i_\infty$ from the first image to the i-th by writing

$$\mathtt{H}^i_\infty = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right)\left(\mathtt{A}^1 - \mathbf{a}^1\mathbf{p}^\mathsf{T}\right)^{-1}. \tag{19.23}$$

This is not strictly necessary, however, since one can *invent* a new view that *does* have camera matrix in the canonical form $[\mathtt{I} \mid \mathbf{0}]$, and express the infinite homographies with respect to this view. In the following discussion, we write $\mathtt{K}$ and $\boldsymbol{\omega}$ (without superscripts) to refer either to this reference view, or to the first view, if it is in canonical form.

The absolute conic lies on $\boldsymbol{\pi}_\infty$ so its image is mapped between views by $\mathtt{H}_\infty$. Under the point transformation $\mathbf{x}^i = \mathtt{H}^i_\infty\mathbf{x}$, where $\mathtt{H}^i_\infty$ is the infinite homography between the reference view and view i, the transformation rules for a dual conic (result 2.14(*p*37))

and a point conic (result 2.13(*p*37)) lead to the relations

$$\boldsymbol{\omega}^{*i} = \mathtt{H}^i_\infty \boldsymbol{\omega}^* \mathtt{H}^{i\ \mathsf{T}}_\infty \quad \text{and} \quad \boldsymbol{\omega}^i = (\mathtt{H}^i_\infty)^{-\mathsf{T}} \boldsymbol{\omega}\, (\mathtt{H}^i_\infty)^{-1}. \tag{19.24}$$

where $\boldsymbol{\omega}^i$ is the IAC in the i-th view. It may be verified that these equations are precisely the auto-calibration equations (19.4–*p*461), and this is another geometric interpretation of those equations.

These are among the most important relationships for auto-calibration. They are the basis for obtaining metric reconstruction from affine reconstruction, and also for calibrating a non-translating camera, as will be seen later in section 19.6. The significance of this relation for auto-calibration is that if $\mathtt{H}^i_\infty$ is known, then these are *linear* relations between $\boldsymbol{\omega}^i$ and $\boldsymbol{\omega}$ (and similarly for $\boldsymbol{\omega}^*$). This means that constraints placed on $\boldsymbol{\omega}^i$ in one view can easily be transferred to another and in this manner sufficient constraints may be assembled to determine $\boldsymbol{\omega}$ by linear means. Once $\boldsymbol{\omega}$ is determined, then $\mathtt{K}$ follows by the Cholesky decomposition. We will illustrate this approach for the example of fixed internal parameters.

Sketch solution for identical internal parameters. If the internal parameters are constant over m views then $\mathtt{K}^i = \mathtt{K}$ and $\boldsymbol{\omega}^{*i} = \boldsymbol{\omega}^*$ for $i = 1, \ldots, m$, and the $\boldsymbol{\omega}^*$ equation of (19.24) becomes

$$\boldsymbol{\omega}^* = \mathtt{H}^i_\infty \boldsymbol{\omega}^* \mathtt{H}^{i\ \mathsf{T}}_\infty. \tag{19.25}$$

A very important point here relates to the scale factors in the equation (19.25).

- *Although (19.25) is a relationship between homogeneous quantities, the scale factor in the homogeneous equation can be chosen as unity provided* $\mathtt{H}^i_\infty$ *is normalized as* $\det \mathtt{H}^i_\infty = 1$.

This results in six equations for the independent elements of the symmetric matrix $\boldsymbol{\omega}^*$. Then (19.25) can be written in a homogeneous *linear* form

$$\mathtt{A}\mathbf{c} = \mathbf{0}, \tag{19.26}$$

where $\mathtt{A}$ is a 6×6 matrix composed from the elements of $\mathtt{H}^i_\infty$, and $\mathbf{c}$ is the conic $\boldsymbol{\omega}^*$ written as a 6-vector. As discussed below, $\mathbf{c}$ is not uniquely determined by one such equation since $\mathtt{A}$ has at most rank 4. However, if linear equations (19.26) from $m \geq 2$ view pairs are combined, so that $\mathtt{A}$ is now a $6m \times 6$ matrix, and provided the rotations between the views are about different axes, then in general $\mathbf{c}$ is determined uniquely.

Related to the issue of uniqueness is the issue of numerical stability. Under a single motion the linear computation of $\mathtt{K}$ from $\mathtt{H}_\infty$ is extremely sensitive to the accuracy of $\mathtt{H}_\infty$. If $\mathtt{H}_\infty$ is inaccurate it is not always possible to obtain a positive-definite matrix $\boldsymbol{\omega}$ (or $\boldsymbol{\omega}^*$) and thus to apply the Cholesky decomposition to obtain $\mathtt{K}$. This sensitivity is reduced if further motions are made, and $\boldsymbol{\omega}$ obtained from the combined constraints of a number of $\mathtt{H}_\infty$'s.

Advantage of using the IAC. In an analogous manner to the linear solution for $\boldsymbol{\omega}^*$, a linear solution may be obtained for $\boldsymbol{\omega}$ starting from (19.24). In fact using the equations involving the IAC is attractive for the following reason. In the zero-skew case the form of (19.12–*p*464) for the IAC is simpler and more clearly reflects the role of each calibration parameter than does the corresponding formula (19.11–*p*464) for the DIAC. In order to obtain linear equations using the DIAC equations (19.11) it is necessary to assume that the principal point is known. This is not necessary for equations deriving from the IAC. An assumption of zero skew is quite natural and is a safe assumption for most imaging conditions. However, an assumption of known principal point is much less tenable. For this reason it is usually preferable to use the IAC constraints of (19.24) for auto-calibration rather than using the DIAC constraints.

Other calibration constraints. The algorithm just described was for constant but arbitrary internal parameters. If more is known about $\mathtt{K}$, such as the value of the aspect ratio or that the skew is zero, the corresponding constraints may be simply added to the set of equations on $\boldsymbol{\omega}$ (or $\boldsymbol{\omega}^*$), and imposed as soft constraints. The possible constraints are shown in table 19.2(*p*465) for the DIAC (the same constraints that are used to compute $\mathtt{Q}_\infty^*$ in section 19.3) and table 19.4 for the IAC.

As mentioned above, the constraints derived from the IAC are generally linear, whereas the constraints derived from the DIAC, are linear only under the assumption that the principal point is known (and at the origin).

Just as with the absolute dual quadric method, it is possible to allow varying internal parameters for the cameras, as long as sufficiently many constraints are imposed. The constraint of constant internal parameters for the cameras imposed a total of $5(m-1)$ constraints on the calibration parameters of the m views. We can make do with fewer constraints, letting certain parameters vary. The method of calibration with varying internal parameters is quite analogous to that used in the case of the absolute dual quadric. Each entry of $\boldsymbol{\omega}^i$ is expressible as a linear expression in the entries of $\boldsymbol{\omega}$ according to (19.24). A linear constraint on some entry of $\boldsymbol{\omega}^i$ therefore maps back to a linear constraint on the entries of $\boldsymbol{\omega}$.

Note: To avoid treating the first image differently, any constraints imposed on the first camera, such as $\boldsymbol{\omega}^1_{12} = 0$ (camera 1 has zero skew) should be treated by adding this equation to the complete equation set, rather than by decreasing the number of parameters used to describe $\boldsymbol{\omega}^1$. The latter method would cause the zero-skew constraint to be enforced exactly in the first image (a hard constraint), but it would be a soft constraint in the other images.

Algorithm 19.2 summarizes the stratified method for both constant and varying parameters. One could imagine implementing this algorithm directly for a camera mounted on a robot: the camera is first moved by a pure translation in order to determine $\boldsymbol{\pi}_\infty$; and in subsequent motions the camera may both translate and rotate until sufficient rotations have accumulated to determine $\mathtt{K}$ uniquely.

Condition	constraint	type	# constraints
zero skew	$\boldsymbol{\omega}_{12} = 0$	linear	m
principal point at origin	$\boldsymbol{\omega}_{13} = \boldsymbol{\omega}_{23} = 0$	linear	$2m$
known aspect ratio $r = \alpha_y/\alpha_x$ (assuming zero skew)	$\boldsymbol{\omega}_{11} = r^2\boldsymbol{\omega}_{22}$	linear	m
fixed (unknown) aspect ratio (assuming zero skew)	$\boldsymbol{\omega}^i_{11}/\boldsymbol{\omega}^i_{22} = \boldsymbol{\omega}^j_{11}/\boldsymbol{\omega}^j_{22}$	quadratic	$m-1$

Table 19.4. **Auto-calibration constraints derived from the IAC**. *These constraints are derived directly from the form of (19.10–*p*464) and (19.12–*p*464). The number of constraints column gives the total number of constraints over m views, assuming the constraint is true for each view.*

Using hard constraints. Algorithm 19.2 comes down to solving a homogeneous set of equations of the form $\mathtt{A}\mathbf{c} = \mathbf{0}$, where $\mathbf{c}$ represents $\boldsymbol{\omega}$ arranged as a 6-vector. Generally the supplied information on $\mathtt{K}$, such as that the skew is zero, will not be satisfied exactly. As discussed in section 8.8(*p*223) known information can be imposed as a hard constraint by parametrizing $\boldsymbol{\omega}^i$ in each view to satisfy this constraint. For instance if the camera is known to have square pixels then the remaining parameters for the IAC of each view can be represented by a homogeneous 4-vector. Linear equations for the unknown parameters in each view may again be obtained from (19.24). A homogeneous set of equations of the form $\mathtt{A}\mathbf{c} = \mathbf{0}$ may then be assembled, where $\mathbf{c}$ now represents the unknown parameters of $\boldsymbol{\omega}^i$ over all views, and a solution which minimizes $\|\mathtt{A}\mathbf{c}\|$ obtained in the usual manner via the SVD. An alternative is to include all the parameters in each view and use algorithm A5.5(*p*594) to minimize $\|\mathtt{A}\mathbf{c}\|$, while satisfying constraints $\mathtt{C}\mathbf{c} = \mathbf{0}$ exactly.

19.5.3 The ambiguities in using the infinite homography relation

In this section we describe the ambiguities in determining the internal parameters from (19.25) that occur if only a single rotation axis is used. It will be assumed that the internal parameters are unknown but fixed.

A rotation matrix $\mathtt{R}$ has an eigenvector $\mathbf{d}_\mathrm{r}$ with unit eigenvalue, $\mathtt{R}\mathbf{d}_\mathrm{r} = 1\mathbf{d}_\mathrm{r}$, where $\mathbf{d}_\mathrm{r}$ is the direction of the axis of the rotation. Consequently, the matrix $\mathtt{H}^i_\infty = \mathtt{K}\mathtt{R}^i\mathtt{K}^{-1}$ also has an eigenvector with unit eigenvalue (provided $\mathtt{H}^i_\infty$ is normalized as $\det \mathtt{H}^i_\infty = 1$). This eigenvector is $\mathbf{v}_\mathrm{r} = \mathtt{K}\mathbf{d}_\mathrm{r}$, and the image point $\mathbf{v}_\mathrm{r}$ corresponds to the vanishing point of the rotation axis direction. Suppose $\boldsymbol{\omega}^*_\mathrm{true}$ is the true $\boldsymbol{\omega}^*$; then, it may be verified that if $\boldsymbol{\omega}^*_\mathrm{true}$ satisfies (19.25) with $\mathtt{H}^i_\infty = \mathtt{K}\mathtt{R}^i\mathtt{K}^{-1}$, then so does the one-parameter family of (dual) conics

$$\boldsymbol{\omega}^*(\mu) = \boldsymbol{\omega}^*_\mathrm{true} + \mu\mathbf{v}_\mathrm{r}\mathbf{v}_\mathrm{r}^\mathsf{T} \tag{19.27}$$

where μ parametrizes the family. In a similar manner there is a one parameter family (pencil) of solutions for the IAC equation of (19.24). This argument indicates that

Objective

Given a projective reconstruction $\{\mathtt{P}^i, \mathbf{X}_j\}$, where $\mathtt{P}^i = [\mathtt{A}^i \mid \mathbf{a}^i]$, determine a metric reconstruction via an intermediate affine reconstruction.

Algorithm

(i) **Affine rectification:** Determine the vector $\mathbf{p}$ that defines $\boldsymbol{\pi}_\infty$, using one of the methods described in section 19.5.1. At this point an affine reconstruction may be obtained as $\{\mathtt{P}^i\mathtt{H}_\mathrm{P}, \mathtt{H}_\mathrm{P}^{-1}\mathbf{X}_j\}$ with

$$\mathtt{H}_\mathrm{P} = \begin{bmatrix} \mathtt{I} & \mathbf{0} \\ -\mathbf{p}^\mathsf{T} & 1 \end{bmatrix}.$$

(ii) **Infinite homography:** Compute the infinite homography between the reference view and the others as

$$\mathtt{H}^i_\infty = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^\mathsf{T}\right).$$

Normalize the matrix so that $\det \mathtt{H}^i_\infty = 1$.

(iii) **Compute $\boldsymbol{\omega}$:**

- In the case of constant calibration: rewrite the equations $\boldsymbol{\omega} = (\mathtt{H}^i_\infty)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i_\infty)^{-1}$, $i = 1, \ldots, m$ as $\mathtt{A}\mathbf{c} = \mathbf{0}$ with $\mathtt{A}$ a $6m \times 6$ matrix, and $\mathbf{c}$ the elements of the conic $\boldsymbol{\omega}$ arranged as a 6-vector, **or**
- For variable calibration parameters, use the equation $\boldsymbol{\omega}^i = (\mathtt{H}^i_\infty)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i_\infty)^{-1}$ to express linear constraints on entries of $\boldsymbol{\omega}^i$ (e.g. zero skew) as linear equations in the entries of $\boldsymbol{\omega}$.

(iv) Obtain a least-squares solution to $\mathtt{A}\mathbf{c} = \mathbf{0}$ via SVD.

(v) **Metric rectification:** Determine the camera matrix $\mathtt{K}$ from the Cholesky decomposition $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$. Then a metric reconstruction is obtained as $\{\mathtt{P}^i\mathtt{H}_\mathrm{P}\mathtt{H}_\mathrm{A}, (\mathtt{H}_\mathrm{P}\mathtt{H}_\mathrm{A})^{-1}\mathbf{X}_j\}$ with

$$\mathtt{H}_\mathrm{A} = \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}.$$

(vi) Use iterative least-squares minimization to improve the solution (see section 19.3.3).

Algorithm 19.2. *Stratified auto-calibration algorithm using IAC constraints.*

although the infinite homography constraint seemingly provides six constraints on the five degrees of freedom of $\boldsymbol{\omega}^*$, only four of these constraints are linearly independent.

Removing the ambiguity. The one-parameter ambiguity may be resolved in several ways. First, if there is another view available related by a rotation around an axis with a direction different to $\mathbf{d}_\mathrm{r}$, then the combination of both sets of constraints will not have this ambiguity. A linear solution is easily obtained in the manner of (19.26). Thus with a minimum of three views (i.e. more than one rotation) a unique solution can generally be obtained. A second method of resolving the ambiguity is to make assumptions on the internal parameters of the cameras: for instance an assumption of zero skew (see table 19.4). The equations enforcing zero skew may be added as hard constraints to the set of equations being solved.

An alternative (but equivalent) method enforces the constraints *a posteriori* in the following manner. An ambiguity in solving for $\mathbf{c}$, from the linear equation system

$\mathtt{A}\mathbf{c} = \mathbf{0}$, occurs when $\mathtt{A}$ has a 2-dimensional (or greater) right null-space. In this case in solving for $\boldsymbol{\omega}$ there would be a family of solutions of the form

$$\boldsymbol{\omega}(\alpha) = \boldsymbol{\omega}_1 + \alpha\boldsymbol{\omega}_2.$$

Here $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ are known from the null-space generators, and α must be determined. It remains simply to find the value of α that leads to a solution satisfying the chosen constraint condition in table 19.4. This is solved linearly. One could do the same thing solving for the DIAC, but then the constraint condition would be quadratic (see table 19.2(*p*465)), and one of the solutions would be spurious.

In certain cases, these additional constraints do not resolve the ambiguity. For example, skew-zero does not resolve the ambiguity if the rotation is about the image x- or y-axes. Such exceptions are described in more detail in [Zisserman-98], and we give a few commonly occurring examples now.

Typical ambiguities. The one-parameter family of solutions given in (19.27) for $\boldsymbol{\omega}^*(\mu)$ corresponds to a one-parameter family of calibration matrices obtained from $\boldsymbol{\omega}^*(\mu)$ as $\boldsymbol{\omega}^*(\mu) = \mathtt{K}(\mu)\mathtt{K}(\mu)^\mathsf{T}$. For simplicity we will assume that the true camera $\mathtt{K}$ (which is a member of this family) has skew zero, so $\mathtt{K}$ has four unknown parameters.

If the rotation axis is parallel to the camera X-axis, then $\mathbf{d}_\mathrm{r} = (1,0,0)^\mathsf{T}$ and $\mathbf{v}_\mathrm{r} = \mathtt{K}\mathbf{d}_\mathrm{r} = \alpha_x(1,0,0)^\mathsf{T}$. From the form (19.11–*p*464) of $\boldsymbol{\omega}^*$ with no skew, the family (19.27) is

$$\boldsymbol{\omega}^*(\mu) = \boldsymbol{\omega}^*_\mathrm{true} + \mu\mathbf{v}_\mathrm{r}\mathbf{v}_\mathrm{r}^\mathsf{T} = \begin{bmatrix} \alpha_x^2(1+\mu) + x_0^2 & x_0y_0 & x_0 \\ x_0y_0 & \alpha_y^2 + y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix}. \qquad (19.28)$$

Note that the entire family has skew-zero, and in this case only the element $\boldsymbol{\omega}^*_{11}$ is varying. This means that the principal point and α_y are unambiguously determined – since they may be read-off from elements which are unaffected by the ambiguity. However, it is apparent that α_x cannot be determined because it only appears in the varying element $\boldsymbol{\omega}^*_{11}(\mu)$. To summarize this, and two other canonical cases:

- *If* $\mathtt{K}$ *is computed from the infinite homography relation (19.25) assuming a zero-skew camera, then for some motions, there remains one undetermined calibration parameter. For rotation about various axes this ambiguity is as follows.*

  (i) X**-axis:** α_x *is undetermined;*
  (ii) Y**-axis:** α_y *is undetermined;*
  (iii) Z**-axis (principal axis):** α_x *and* α_y *are undetermined, but their ratio* α_y/α_x *is determined.*

Geometric note. These ambiguities are not limited to calibration from a pair of views, but apply to complete sequences. For instance if the set of rotations in a camera motion are all about the X-axis of the camera, then there will be a reconstruction ambiguity, and

the same is true for Y-axis rotations. One can see this geometrically as follows. Consider a metric reconstruction of a scene from a sequence of images with only Y-axis rotations of the camera. One can define a coordinate system in which the world Y-axis is aligned with the direction of the camera's y-axis. Now, consider "squashing" the whole reconstruction (points and camera positions) so that their Y coordinate is multiplied by some factor k. From the imaging geometry, it is easy to see that this will have the effect of multiplying the y image coordinate of any imaged point by the same factor k, but not affecting the x-coordinate. However, this effect can be undone by multiplying the scale factor α_y of coordinates in the image by the inverse factor k^{-1}, thereby leaving image coordinates unchanged. This shows that α_y is not unambiguously determined, in fact it is unconstrained. In summary there is a one-parameter ambiguity parallel to the rotation axis in the metric reconstruction and a corresponding one-parameter ambiguity in the internal parameters. This argument shows that the problem of ambiguity is intrinsic to the motion, and not to any particular auto-calibration algorithm.

Relationship to the Kruppa equations. Writing (19.24) for two views as $\boldsymbol{\omega}^{*\prime} = \mathtt{H}_\infty \boldsymbol{\omega}^* \mathtt{H}_\infty^\mathsf{T}$ and multiplying before and after by the matrix $[\mathbf{e}']_\times$ leads to

$$[\mathbf{e}']_\times \boldsymbol{\omega}^{*\prime} [\mathbf{e}']_\times = [\mathbf{e}']_\times \mathtt{H}_\infty \boldsymbol{\omega}^* \mathtt{H}_\infty^\mathsf{T} [\mathbf{e}']_\times = \mathtt{F} \boldsymbol{\omega}^* \mathtt{F}^\mathsf{T}$$

since $\mathtt{F} = [\mathbf{e}']_\times \mathtt{H}_\infty$. This is simply the Kruppa equations (19.18–*p*471), which shows that they follow immediately from the infinite homography constraint. Since $[\mathbf{e}']_\times$ is not invertible, one can not go the other direction and derive the infinite homography constraint from the Kruppa equations. Thus, the Kruppa equations are a weaker constraint.

However the difference is that to apply (19.24) one needs to know the plane at infinity (and hence affine structure), since it is true only for the infinite homography, and not for an arbitrary H. The Kruppa equations, on the other hand, do not involve any knowledge of affine structure of the scene. Nevertheless, this relationship shows that for a sequence of images, any calibration ambiguity under the infinite homography relation is also an ambiguity of the Kruppa equations.

19.6 Calibration from rotating cameras

In this section, we begin consideration of calibration under special imaging conditions. The situation considered here is the one in which the camera rotates about its centre but does not translate. We will consider both the case of fixed internal parameters, and the case of some parameters known and fixed whilst others are unknown and varying.

This situation is one that occurs frequently. Examples include: pan–tilt and zoom surveillance cameras; cameras used for broadcasts of sporting events which are almost invariably fixed in location but free to rotate and zoom; and hand-held camcorders which are very often panned from a single viewpoint. Even though the rotation is not exactly about the centre, in practice the translation is generally negligible compared to the distance of scene points, and a fixed centre is an excellent approximation.

The calibration problem from rotating cameras is mathematically identical with the affine-to-metric calibration step in stratified reconstruction, as given in section 19.5.2.

Objective

Given $m \geq 2$ views acquired by a camera rotating about its centre with fixed or varying internal parameters, compute the parameters of each camera. It is assumed that the rotations are not all about the same axis.

Algorithm

(i) **Inter-image homographies:** Compute the homography $\mathtt{H}^i$ between each view i and a reference view such that $\mathbf{x}^i = \mathtt{H}^i\mathbf{x}$ using, for example, algorithm 4.6(p123). Normalize the matrices such that $\det \mathtt{H}^i = 1$.

(ii) **Compute $\boldsymbol{\omega}$:**
- In the case of constant calibration: rewrite the equations $\boldsymbol{\omega} = (\mathtt{H}^i)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i)^{-1}$, $i = 1, \ldots, m$ as $\mathtt{A}\mathbf{c} = \mathbf{0}$ where $\mathtt{A}$ is a $6m \times 6$ matrix, and $\mathbf{c}$ the elements of the conic $\boldsymbol{\omega}$ arranged as a 6-vector, **or**
- For variable calibration parameters, use the equation $\boldsymbol{\omega}^i = (\mathtt{H}^i)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i)^{-1}$ to express linear constraints on entries of $\boldsymbol{\omega}^i$ in table 19.4 (e.g. unit aspect ratio) as linear equations in the entries of $\boldsymbol{\omega}$.

(iii) **Compute $\mathtt{K}$:** Determine the Cholesky decomposition of $\boldsymbol{\omega}$ as $\boldsymbol{\omega} = \mathtt{U}\mathtt{U}^{\mathsf{T}}$, and thence $\mathtt{K} = \mathtt{U}^{-\mathsf{T}}$.

(iv) **Iterative improvement:** (Optional) Refine the linear estimate of $\mathtt{K}$ by minimizing

$$\sum_{i=2,m;\ j=1,n} d(\mathbf{x}^i_j, \mathtt{K}\mathtt{R}^i\mathtt{K}^{-1}\mathbf{x}_j)^2$$

over $\mathtt{K}$ and $\mathtt{R}^i$, where $\mathbf{x}_j, \mathbf{x}^i_j$ are the position of the j-th point measured in the first and i-th images respectively. Initial estimates for the minimization are obtained from $\mathtt{K}$ and $\mathtt{R}^i = \mathtt{K}^{-1}\mathtt{H}^i\mathtt{K}$.

Algorithm 19.3. *Calibration for a camera rotating about its centre.*

From a non-translating camera it is impossible to achieve an affine (or any) reconstruction, because there is no way to resolve depth. Nevertheless, we may compute the infinite homography between the images, which is all that is needed to determine the camera calibration.

As has been shown earlier (section 8.4(p202)) the images of two cameras with a common centre are related by a plane projective transformation. Indeed, if $\mathbf{x}^i$ and $\mathbf{x}$ are corresponding image points then they are related by $\mathbf{x}^i = \mathtt{H}^i\mathbf{x}$, where $\mathtt{H}^i = \mathtt{K}^i\mathtt{R}^i(\mathtt{K})^{-1}$, and $\mathtt{R}^i$ is the rotation between view i and the reference view. Furthermore, since this map is independent of the depth of the points imaged at $\mathbf{x}$, it applies also to points at infinity, so as shown in section 13.4(p338),

$$\mathtt{H}^i = \mathtt{H}^i_\infty = \mathtt{K}^i\mathtt{R}^i(\mathtt{K})^{-1}.$$

Thus we have a convenient means of measuring $\mathtt{H}_\infty$ directly from images.

Given $\mathtt{H}_\infty$ a solution for the calibration matrices $\mathtt{K}^i$ of all the images in the set acquired by the rotating camera may be obtained as described in section 19.5.2. The method may be applied to either fixed or variable internal parameters and is summarized in algorithm 19.3. We will illustrate this by a number of examples.

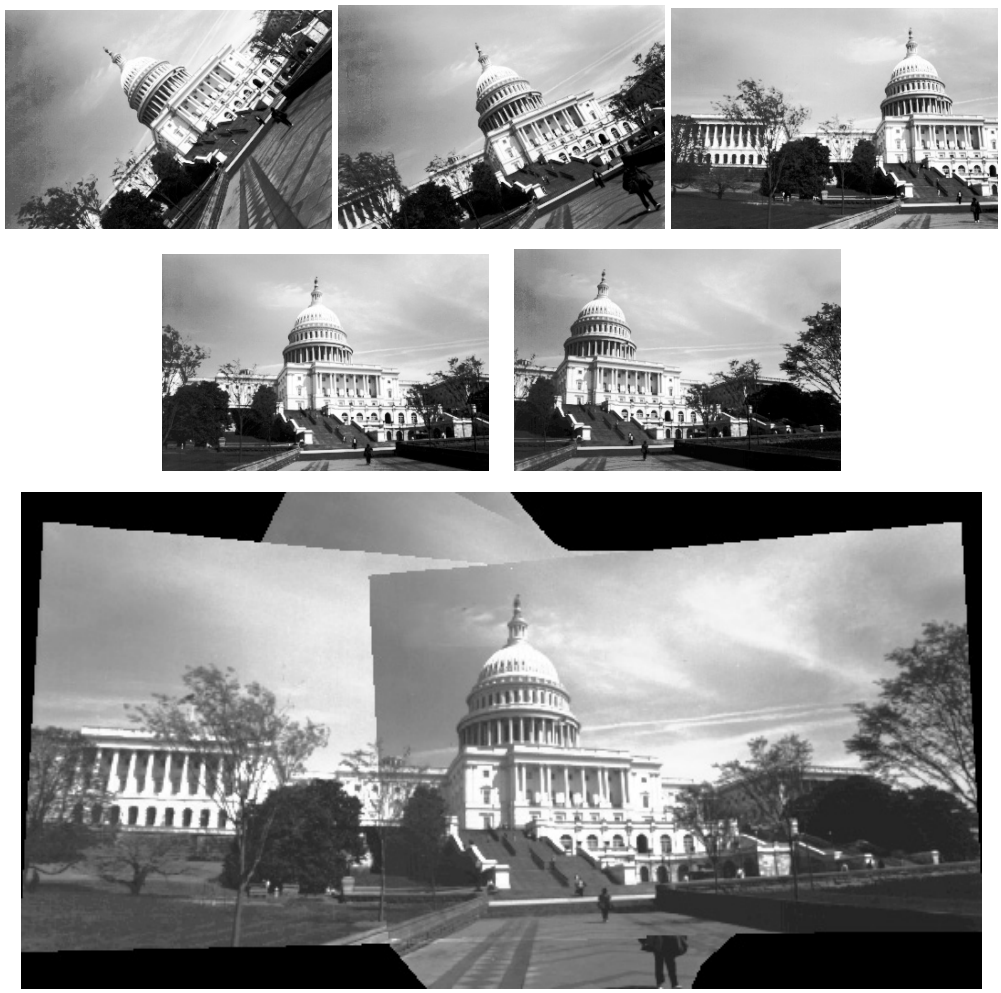

Fig. 19.3. **Calibrating a camera rotating about its centre.** *(***upper***) Five images of the US Capitol acquired by approximately rotating the camera about its centre. (***lower***) A mosaic image constructed from the five images (see example 8.14(*p207*)). The mosaic image shows very clearly the distortion effect of the infinite homography between the images. Analysis of this distortion provides the basis for the auto-calibration algorithm. The calibration is computed as described in algorithm 19.3.*

Example 19.9. Rotation about centre with fixed internal parameters

The images in figure 19.3 were obtained using a 35mm camera with ordinary black and white film to produce negatives. The camera was hand-held, and no particular care was taken to ensure that the camera centre remained stationary.

Prints enlarged from these negatives were digitized using a flat-bed scanner. The enlargement process can lead to a non-zero value of s and unequal values of α_x and α_y if the negative and print paper are not precisely parallel. The resulting image size was 776×536 pixels.

The constraint applied here is that the internal parameters are constant. The camera

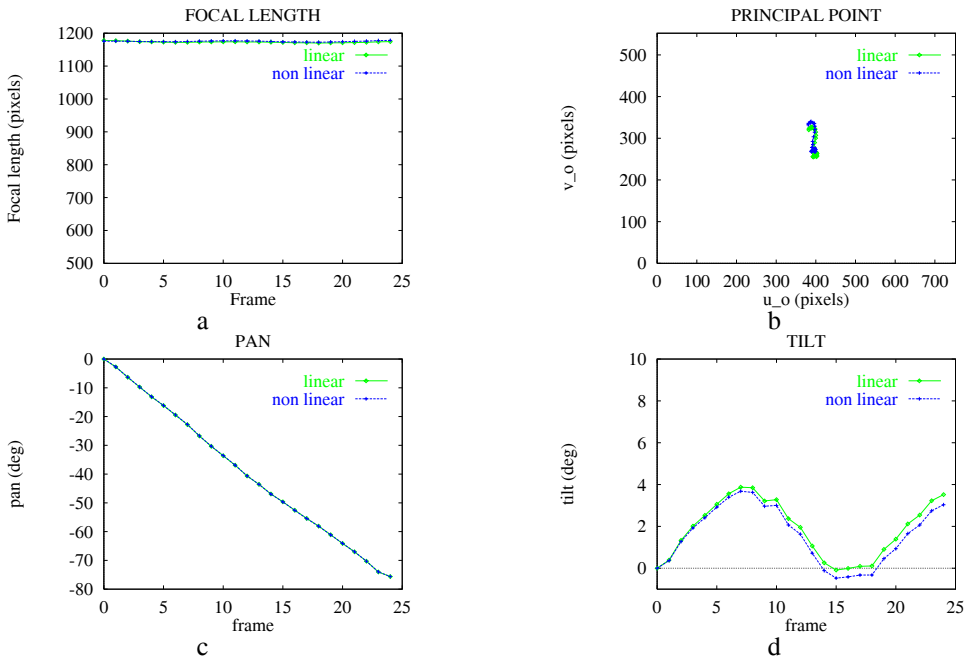

Fig. 19.4. **Rotation for varying internal parameters assuming square pixels.** *These are for the panned sequence of figure 8.9*(p*206). (a) Focal length. (b) Principal point. (c) Pan angle. (d) Tilt angle. Figures courtesy of Lourdes de Agapito Vicente.*

matrix computed as described in algorithm 19.3 is

$$\mathtt{K}_{\text{linear}} = \begin{bmatrix} 964.4 & -4.9 & 392.8 \\ & 966.4 & 282.0 \\ & & 1 \end{bmatrix} \quad \mathtt{K}_{\text{iterative}} = \begin{bmatrix} 956.8 & -6.4 & 392.0 \\ & 959.3 & 281.4 \\ & & 1 \end{bmatrix}.$$

There is little difference between the linear and iterative estimates, and the computed aspect ratio (virtually unity) and principal point are very reasonable. △

Example 19.10. Rotation about centre with varying internal parameters

The images in figure 8.9(*p*206) were acquired by panning a camcorder approximately about its centre. The camera was not zoomed, but due to auto-focus there might be slight variations in the focal length and principal point.

In this example the constraint used is that the pixels are square: i.e. that the skew is zero and the aspect ratio unity; but the focal length and principal point are unknown and not fixed. Then from table 19.4 we have two linear constraints on $\boldsymbol{\omega}$ from each view. From zero skew $(\mathtt{H}_\infty^{i\ -\mathsf{T}}\,\boldsymbol{\omega}\,\mathtt{H}_\infty^{i\ -1})_{12} = 0$, and from unit aspect ratio $(\mathtt{H}_\infty^{i\ -\mathsf{T}}\,\boldsymbol{\omega}\,\mathtt{H}_\infty^{i\ -1})_{11} = (\mathtt{H}_\infty^{i\ -\mathsf{T}}\,\boldsymbol{\omega}\,\mathtt{H}_\infty^{i\ -1})_{22}$. These constraints are assembled to give a linear system of equations on $\boldsymbol{\omega}$ as described in algorithm 19.3.

The internal parameters of the computed camera matrix for each view are shown in figure 19.4. It is evident that the recovered focal length and principal point are quite constant (even though this was not imposed), and the pan and tilt angles are very reasonable for this hand-held sequence. △

19.7 Auto-calibration from planes

For a set of images of a planar scene, the two-step approach of estimating a projective reconstruction followed by computation of a rectifying transformation to take it to metric does not work. This is because it is not possible to determine the cameras without depth relief. As seen in section 17.5.2(*p*425), a minimum of two points not lying on the plane are required. Nevertheless, auto-calibration from scene planes *is* possible. This was shown by [Triggs-98] who gave a solution in the case of constant internal parameters. The method is especially interesting from the point of view of potential applications. Scenes consisting of planes are extremely common in man-made environments, e.g. the ground plane. Furthermore, in aerial images acquired by a high-flying aircraft or a satellite, the depth-relief of the scene is small compared with the extent of the image, and the scene may be accurately approximated as a plane and the auto-calibration method will apply.

The starting point of the algorithm is a set of image–to–image homographies induced by the world plane. These can all be related back to the first image, providing a set of homographies $\mathtt{H}^i$. Geometrically auto-calibration from planes is then a marriage of two ideas. First, the circular points on the plane, which are the intersection of the absolute conic with the plane, are mapped from image to image via the homographies. Second, as we have seen in example 8.18(*p*211), the calibration matrix $\mathtt{K}$ may be determined from the imaged circular points of a plane (with two constraints provided by each image).

Thus suppose the images of the circular points (4 dof) are determined in the first image (by some means), then they may be transferred to the other views by the known $\mathtt{H}^i$. In each view then there are two constraints on $\boldsymbol{\omega}$, since the two imaged circular points lie on $\boldsymbol{\omega}$. In detail, if we denote the (at this stage unknown) imaged circular points in the first view by $\mathbf{c}_j, j = 1, 2$. Then the auto-calibration equations are

$$(\mathtt{H}^i\mathbf{c}_j)^\mathsf{T}\boldsymbol{\omega}^i(\mathtt{H}^i\mathbf{c}_j) = 0, \quad i = 1, \ldots, m \quad j = 1, 2 \tag{19.29}$$

where $\mathtt{H}^1 = \mathtt{I}$. In solving these equations the unknowns are the coordinates of the circular points in the first image, and some number of unknown calibration parameters. Although the circular points are complex points, they are complex conjugates of each other, so 4 parameters suffice to describe them. If in addition there are a total of v unknowns in the internal parameters $\mathtt{K}^i$ of all m views, then a solution is possible provided $2m \geq v + 4$, since each view provides two equations.

Restrictions on the internal parameters of the cameras lead to further algebraic constraints according to table 19.4 and these are used to supplement the constraints imposed by (19.29). Various cases are considered in table 19.5. In most cases, calibration from a plane is a non-linear problem and nothing more will be said here about computational aspects of how to find the solution. Iterative methods are necessary, minimizing some cost function. See [Triggs-98] for more details of minimization methods.

Implementation. A considerable implementational advantage of this method is that it only requires the homography between planes, and not the point correspondences

Condition	dof(v)	views
Unknown but constant internal parameters	5	5
Constant known skew and aspect ratio. Constant unknown principal point and focal length	3	4
All internal parameters known except varying focal length	m	4
Varying focal length, all other internal parameters fixed but unknown	$m+4$	8

Table 19.5. **The number of views required for calibration from a plane under various conditions.** *Calibration is (in principle) possible if* $2m \geq v + 4$.

arising from 3-space points that are generally required to estimate multiple view tensors, such as the fundamental matrix. The matching transformation between planes is a much simpler, stabler and accurate computation because of the constrained nature of the inter-image transformation which is point-to-point. The method of algorithm 4.6-(p123) may be used to estimate this transformation between two images. Alternatively, correlation-based methods that estimate the parametrized homography directly from image intensity may be used.

Including additional information. If additional information is available on the plane or the motion then the complexity of auto-calibration using scene planes may be reduced. For example, if the vanishing line of the imaged plane can be determined then only two parameters are required to specify the circular points since the imaged circular points lie on the vanishing line. Indeed if the plane provides sufficient information to estimate its imaged circular points directly (such as a square grid) then the problem reduces to that of calibration from scene constraints discussed in section 8.8(p223).

Similarly constraints on the motion may be used to simplify the problem. One particular example is where the rotation axis describing the camera motion is parallel to the scene plane normal. In this case the imaged circular points may be computed directly from the fixed points of the homography and two linear constraints placed on $\boldsymbol{\omega}$. This situation is discussed further in Note (vii) at the end of the chapter. An example of such a motion is planar motion which is discussed in detail in the following section.

19.8 Planar motion

A case of some practical importance is that of a camera moving in a plane and rotating about an axis perpendicular to that plane. This is the case for a roaming vehicle moving on a ground plane, with a camera fixed with respect to the vehicle body. In this case, the camera must move in a plane parallel to the (horizontal) ground, and as the vehicle turns, the camera will rotate about a vertical axis. It is *not* assumed that the camera is pointing horizontally, or is in any other particular orientation with respect to the vehicle. However, we assume constant internal calibration for the camera. The constrained nature of the motion makes the calibration task significantly simpler.

It will be shown that given three or more images from a sequence of planar motions an affine reconstruction may be computed. To do this, we need to determine the plane at infinity. This will be done by identifying three points on the plane at infinity, thereby defining the plane. These points will be identified as being fixed points in the sequence of images.

Fixed image points under planar motion. According to section 3.4.1(*p*77), any rigid motion (for instance the camera motion) can be interpreted as a rotation about screw axis along with a translation along the direction of the axis. For a planar motion, the rotation axis is perpendicular to the plane of motion, and the translational part of the screw motion is zero. Think of the vehicle as being swung horizontally around the screw axis. The position of the screw axis with respect to the camera remains fixed, and so it will constitute a line of fixed points in the image. If a second motion takes place about a different axis, then the image of the second axis will be fixed in the second pair of images. The images of the two axes will in general be different, but will intersect in the image of the point where the two screw axes meet. Since the two axes are vertical, they are perforce parallel, and so will meet at their common direction on the plane at infinity. This direction projects into the images at the intersection of the images of the screw axes, which is the vanishing point of the screw axes direction. This image point is called the apex. We now have one fixed point over the views, and this will be used to determine one point on the plane at infinity in a projective reconstruction.

As we have seen the image of the screw axis is a line of fixed points for an image pair. There is also a fixed line which is identifiable from a pair of images as follows. Because the plane of motion of the camera (which will be called the ground plane) is fixed, the set of points in the plane is mapped to the same line in all the images. This line is called the horizon line and is the vanishing line of the ground plane. Since each of the cameras lie on the ground plane their epipoles must all lie on the horizon line. Unlike the image of the screw axis, the horizon line is a fixed line, but not a line of fixed points.

Although the horizon line is not fixed pointwise, it contains two points that are fixed in the image pair, namely the image of the two circular points on the ground plane. These circular points are the intersection of the absolute conic with the ground plane. Since the image of the absolute conic is fixed under rigid motion, and the image of the ground plane is fixed, the images of two circular points must be fixed. In fact, they will be fixed in all images from the planar motion sequence. This is illustrated in figure 19.5.

So far we have described the fixed points of the motion sequence. Computing these fixed points is equivalent to affine reconstruction, since we can back-project from the fixed image points to find the corresponding 3D points on the plane at infinity. Although the apex may be computed from two views; it requires three views to compute the imaged circular points.

Computing the fixed points. The set of points in space that map to the same image point in two images is called the horopter. In general, the horopter is a twisted cubic,

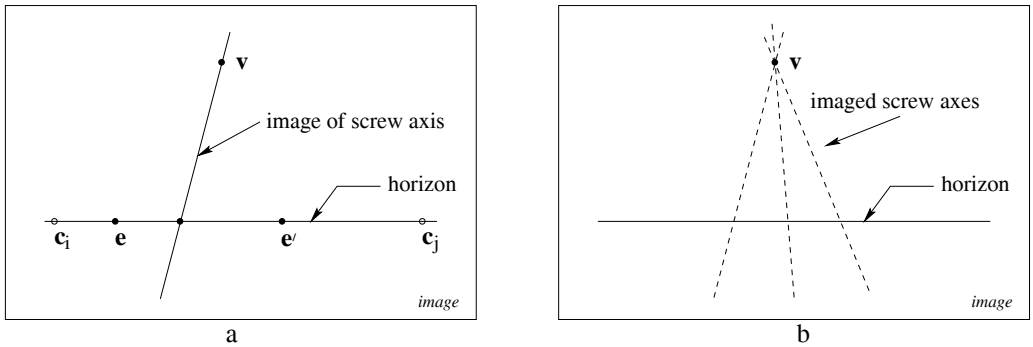

Fig. 19.5. **Fixed image entities for planar motion.** *(a) For two views the imaged screw axis is a line of fixed points in the image under the motion. The horizon is a fixed line under the motion. The epipoles* $\mathbf{e}, \mathbf{e}'$ *and imaged circular points of the ground plane* $\mathbf{c}_i, \mathbf{c}_j$ *lie on the horizon. (b) The relation between the fixed lines obtained pairwise for three images under planar motion. The image horizon lines for each pair are coincident, and the imaged screw axes for each pair intersect in the apex* $\mathbf{v}$*. All the epipoles lie on the horizon line.*

but in the case of planar motion it degenerates to a line (the screw axis) and a conic on the ground plane. The image of the horopter is the conic defined by $\mathtt{F} + \mathtt{F}^\mathsf{T}$, where $\mathtt{F}$ is the fundamental matrix (see section 9.4(*p*250)). In the case of planar motion this will be a degenerate conic consisting of two lines, the image of the screw axis and the horizon line (see figure 9.11(*p*253)). By decomposing the conic, these two lines are determined. From three images we can compute the image of the horopter for each of three pairs, and thereby obtain three sets of horizon lines and imaged axes. The horizon line will be a common component of these sets, and the other components (the images of the screw axes) will intersect in the image at the apex.

Now we turn to computing the circular points. It is useful to understand the geometry of a pair of horopters corresponding to two pairs of images. Let $\mathtt{C}^{12}$ be a conic representing the portion of the horopter for images 1 and 2, lying in the ground plane. This conic will pass through the two circular points, the two camera centres and the intersection of the screw axis with the ground plane. Since the conic contains the circular points it is a circle. Let $\mathtt{C}^{23}$ be the corresponding circle defined from images 2 and 3. The two circular points and the centre of the second camera will lie on both circles. Since two conics intersect in general in four points, there must be a further (real) intersection point. However, this can be discarded, since the points of interest are the two complex intersection points, namely the circular points on the ground plane.

In implementations, authors have chosen different ways of finding the two circular points. In [Armstrong-96b] the method is based on finding fixed lines in three views through the apex using the trifocal tensor. In [Faugeras-98, Sturm-97b] the method involves computing the trifocal tensor of a 1D camera, applied to imaging the 2D ground plane to a 1D image. In both cases the positions of the circular points on the horizon are obtained as the solution of a cubic in one variable.

The main steps of this affine calibration method are summarized in algorithm 19.4.

Objective

Given three (or more) images acquired by a camera with constant internal parameters undergoing planar motion compute an affine reconstruction.

Algorithm

(i) **Compute a projective reconstruction.** from the trifocal tensor $\mathcal{T}$ for the three views. The trifocal tensor may be computed by, e.g. algorithm 16.4(*p*401).
(ii) **Compute pairwise fundamental matrices from** $\mathcal{T}$. See algorithm 15.1(*p*375). Decompose the symmetric part of each fundamental matrix into two lines, a horizon and the image of the screw axes. See section 9.4.1(*p*252).
(iii) **Compute the apex.** Intersect the three imaged screw axes to determine the apex $\mathbf{v}$.
(iv) **Compute the horizon for the triplet.** Obtain the six epipoles from the three fundamental matrices, and determine the horizon by an orthogonal regression fit to these epipoles.
(v) **Compute the imaged circular points.** Compute the position of the imaged circular points on the horizon (see text) $\mathbf{c}_i, \mathbf{c}_j$.
(vi) **Compute the plane at infinity.** Triangulate points on the plane at infinity from the corresponding image points $\mathbf{x} \leftrightarrow \mathbf{x}'$, with $\mathbf{x} = \mathbf{x}'$ for each of $\mathbf{v}, \mathbf{c}_i, \mathbf{c}_j$. This determines three points on $\boldsymbol{\pi}_\infty$, and hence determines the plane.
(vii) **Compute an affine reconstruction.** Rectify the projective reconstruction using the computed $\boldsymbol{\pi}_\infty$ as in algorithm 19.2.

Algorithm 19.4. *Affine calibration for planar motion.*

Metric reconstruction. Once the apex and two circular points are found, the plane at infinity is computed, and the infinite homographies between the images can be computed. Calibration and metric reconstruction now proceeds in the usual way. However, it must be noted that the constrained nature of the motion means that there is a one-parameter family of solutions for the calibration because all the camera rotations are about the same axis. We have the sort of calibration ambiguity considered in section 19.5.3. It is necessary to make an assumption about internal calibration in order to find a unique result. If the y-axis of the camera is parallel with the rotation axis (which may be true in a practical situation), then we have seen that the zero-skew constraint is not sufficient. The best plan is to enforce a zero-skew and known aspect ratio constraint (e.g. if the pixels are square).

Example 19.11. Metric reconstruction for planar motion.
Figure 19.6(a) shows four of seven images of a planar motion sequence. For this sequence the elevation angle of the camera is approximately $20°$. The computed imaged screw axes and horizon lines using all possible pairs are shown in figure 19.6(b), with the resulting estimated apex and horizon line. The positions of the imaged circular points were estimated as $x = 104 \pm 362i,\ y = -86 \mp 2i$. Assuming an aspect ratio of 1.1, the internal parameters of the calibration matrix `K` were computed as $\alpha_x = 330, \alpha_y = 363, x_0 = 123, y_0 = 50$. The accuracy of the metric reconstruction was assessed by measuring metric invariants. Typical results are shown in figure 19.6(c). △

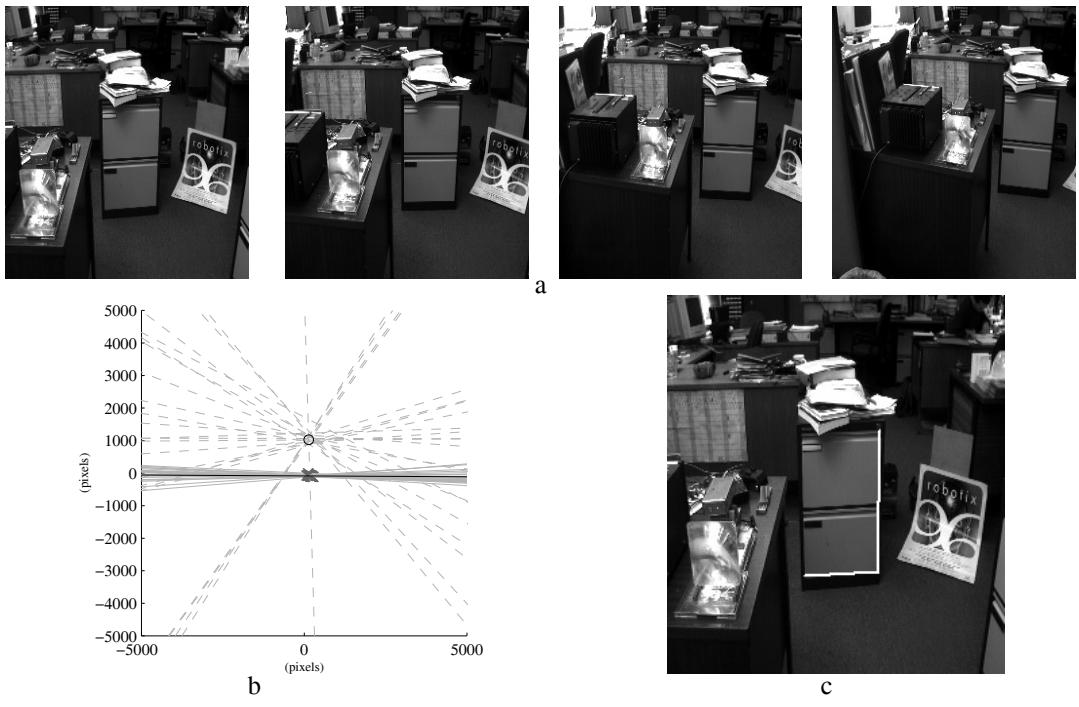

Fig. 19.6. **Planar motion sequence.** *(a) Four images (of seven used) acquired from a camera mounted on a vehicle moving on a plane. (b) The computed epipoles* $(\times)$*, horizon lines (grey solid), and imaged screw axes (grey dashed) for all image pairs. (c) Euclidean invariants measured in the metric reconstruction the right angle is measured as* $89°$*, the ratio of the non-parallel lengths is measured as 0.61 (compared to the approximate veridical value of 0.65).*

19.9 Single axis rotation – turntable motion

In this section we discuss auto-calibration of single axis motions where the relative motion between scene and cameras is equivalent to a rotation about a single fixed axis. This is a specialization of the planar motion case of section 19.8 where here the screw axes for each motion are coincident. This situation occurs, for example, in the case of a static camera viewing an object rotating on a turntable. A second example is that of a camera rotating about a fixed axis (offset from its centre). A third example is that of a camera viewing a rotating mirror.

We will consider here turntable motion and for ease of discussion assume that the axis is vertical so that the motion occurs in a horizontal plane. Again it is *not* assumed that the camera is pointing horizontally, or is in any other particular orientation with respect to the axis. It is assumed that the internal calibration of the camera is constant.

The fixed image points under a sequence of single axis rotation are those of planar motion described in section 19.8, and shown in figure 19.5(a), with the addition that the imaged screw axis is a line of fixed points. The constraint of figure 19.5(b) is not available here as all the imaged screw axes are coincident. The consequence is that it is not possible to determine the apex $\mathbf{v}$ directly, and only the two circular points on $\boldsymbol{\pi}_\infty$ may be recovered from imaged fixed points. This means that the reconstruction, in the absence of constraints on the internal parameters, is not affine, but a particular

parametrized projective transformation. Metric structure is known in the horizontal planes (since the circular points of these planes is known) but there is a 1D projective transformation in the vertical (Z) direction. The resulting ambiguity is the transformation $\mathbf{X}_{\mathrm{P}} = \mathtt{H}\mathbf{X}_{\mathrm{E}}$ where

$$\mathtt{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & \delta & 1 \end{bmatrix} \tag{19.30}$$

and γ and δ are scalars which determine the intersection of the Z axis with $\boldsymbol{\pi}_\infty$ and the relative scaling between the horizontal and vertical directions. An example of the projective transformations represented by family of mappings is shown in figure 1.4-(*p*16).

Computing the fixed points. One way to proceed is to determine the imaged circular points directly. As is evident from figure 19.7(b) the point tracks are ellipses which are images of circles. In 3-space these circles lie in parallel horizontal planes and intersect in the circular points on $\boldsymbol{\pi}_\infty$. In the image, ellipses may be fitted to these tracks, and the common intersection points of the image conics are the (complex conjugate) imaged circular points. The 3D circular points may then be determined by triangulation from two or more views. This is the approach taken by [Jiang-02].

An alternative more algebraic way to proceed is to model the camera matrices as $\mathtt{P}^i = \mathtt{H}_{3\times3}[\mathtt{R}_{\mathtt{Z}}(\theta^i) \mid \mathbf{t}]$ where

$$\mathtt{P}^i = \begin{bmatrix} & & \\ \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \\ & & \end{bmatrix} \begin{bmatrix} \cos\theta^i & -\sin\theta^i & 0 & t \\ \sin\theta^i & \cos\theta^i & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{19.31}$$

with $\mathbf{h}_k$ the columns of $\mathtt{H}_{3\times3}$. This division of the internal and external parameters means that $\mathtt{H}_{3\times3}$ and $\mathbf{t}$ are fixed over the sequence, and only the angle of rotation, θ^i, about the Z axis varies for each camera $\mathtt{P}^i$.

Given this parametrization, the estimation problem can then be precisely stated as determining the common matrix $\mathtt{H}_{3\times3}$ and the angles θ_i in order to estimate the set of cameras $\mathtt{P}^i$ for the sequence. Thus a total of $8+m$ parameters must be estimated for m views, where 8 is the number of degrees of freedom of the homography $\mathtt{H}_{3\times3}$. This is a considerable saving over the $11m$ that would be required for a projective reconstruction of a general motion sequence.

For single axis motion the fundamental matrix has the special form described in section 9.4.1(*p*252), and writing the matrix in terms of the camera matrices (19.31) gives

$$\mathtt{F} = \alpha[\mathbf{h}_2]_\times + \beta\left((\mathbf{h}_1 \times \mathbf{h}_3)(\mathbf{h}_1 \times \mathbf{h}_2)^\mathsf{T} + (\mathbf{h}_1 \times \mathbf{h}_2)(\mathbf{h}_1 \times \mathbf{h}_3)^\mathsf{T}\right)$$

which means that the columns of $\mathtt{H}_{3\times3}$ are partially determined once the fundamental matrix is computed. This is the approach taken by [Fitzgibbon-98b] where a fundamental matrix of the special form is fitted to point correspondences (see section 11.7.2-(*p*293)). It may be shown that from 3 or more views the first two columns $\mathbf{h}_1, \mathbf{h}_2$ of

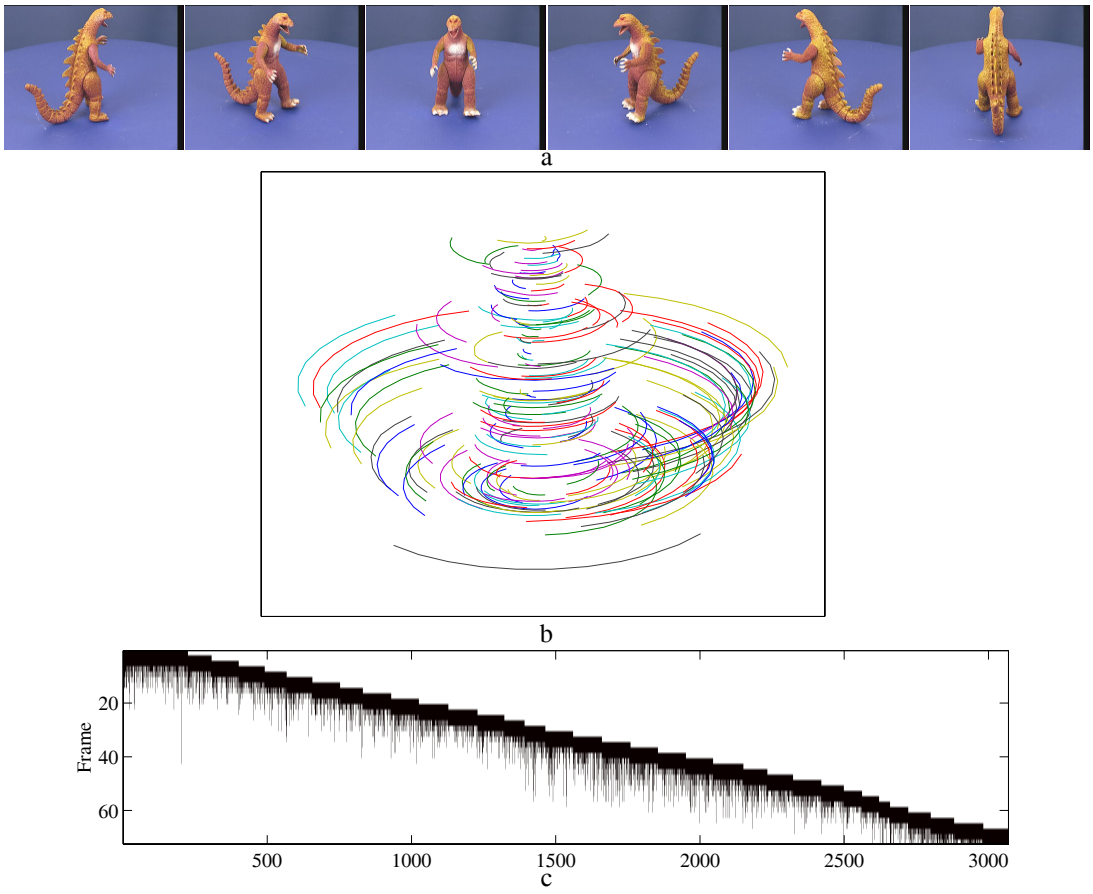

Fig. 19.7. **Dinosaur sequence and tracking:** *(a) six frames from a sequence of 36 of a dinosaur rotating on a turntable (Image sequence courtesy of the University of Hannover [Niem-94]). (b) A subset of the point tracks – only the 200 tracks which survived for longer than 7 successive views are shown. (c)* **Track lifetimes***: Each vertical bar corresponds to a single point track, extending from the first to last frame in which that point was seen. The horizontal ordering is according to where the point first appeared in the sequence. The measurement matrix is relatively sparse, and few points survive longer than 15 frames.*

$\mathtt{H}_{3\times3}$ and the angle θ^i can be fully determined, but $\mathbf{h}_3$ is only determined up to the two-parameter family corresponding to the ambiguity of (19.30).

Metric reconstruction. The two parameter ambiguity in the reconstruction given by (19.30) can be resolved by providing additional information on the internal parameters, for example that the pixels are square. However, if the camera is horizontal with image y axis parallel to the rotation axis, then square pixels only provides one additional constraint (from the aspect ratio, as the skew zero constraint does not provide an additional constraint). In this case further information on the camera is required (e.g. the y coordinate of the principal point) or the aspect ratio of the scene may be used.

Example 19.12. Reconstruction from a turntable sequence

Figure 19.7 shows frames from a sequence of a model dinosaur rotating on a turntable, and the resulting image point tracks. Feature extraction is performed on the luminance component of the colour signal. The projective geometry of the turntable motion is de-

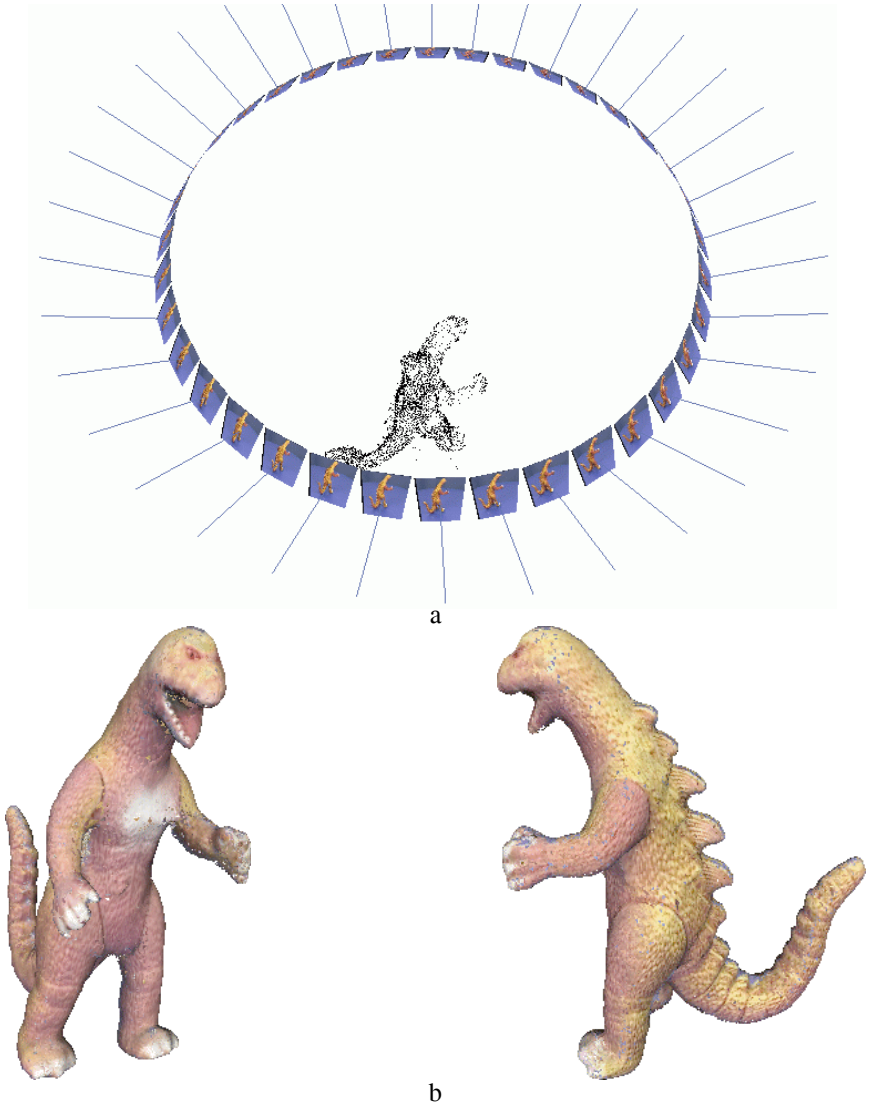

Fig. 19.8. **Dinosaur:** *(a) 3D scene points (about 5000) and camera positions for the Dinosaur sequence. (b) The automatic computation of a 3D graphical model for this sequence is described in [Fitzgibbon-98b].*

termined from these tracks (and no other information) and the resulting reconstruction of cameras and 3D points is shown in figure 19.8. Effectively the camera circumnavigates the object. A 3D texture mapped graphical model may then be computed, in principle, by back-projecting cones defined by the dinosaur silhouette in each frame, and intersecting the set of cones to determine the visual hull of the 3D object. △

19.10 Auto-calibration of a stereo rig

In this section we describe a stratified method for calibrating a "fixed" two-camera stereo rig. Fixed here means that the relative orientation of the cameras on the rig is unchanged during the motion, and the internal parameters of each camera are also

unchanged. It will be shown that from a single motion of the rig the plane at infinity is determined uniquely.

Suppose a fixed stereo rig undergoes a general motion. The projective structure of the scene can be obtained before ($\mathbf{X}$) and after ($\mathbf{X}'$) the motion. Since $\mathbf{X}$ and $\mathbf{X}'$ are two projective reconstructions of the same scene they are related by a 4×4 projective transformation $\mathtt{H}_{\mathrm{P}}$, as

$$\mathbf{X}' = \mathtt{H}_{\mathrm{P}}\mathbf{X}.$$

However, the actual motion of the rig is Euclidean, and it follows (see below) that the homography $\mathtt{H}_{\mathrm{P}}$ is conjugate to the Euclidean transformation representing the motion. Conjugacy is the key result because under a conjugacy relation fixed entities are mapped to fixed entities. Consequently the fixed entities of the Euclidean motion (in particular the plane at infinity) can be *accessed* from the fixed entities of the projective motion represented by $\mathtt{H}_{\mathrm{P}}$.

Conjugacy relation. Suppose $\mathbf{X}_{\mathrm{E}}$ represents a point in 3-space in a Euclidean coordinate frame attached to the rig, and $\mathbf{X}'_{\mathrm{E}}$ represents the same point after the motion of the rig. Then the points are related as

$$\mathbf{X}'_{\mathrm{E}} = \mathtt{H}_{\mathrm{E}}\mathbf{X}_{\mathrm{E}} \tag{19.32}$$

where $\mathtt{H}_{\mathrm{E}}$ is a non-singular 4×4 Euclidean transformation matrix that represents the rotation and translation of the rig. Suppose also that the point is represented in a projective coordinate frame attached to the rig (and which is obtained by a projective reconstruction); then

$$\mathbf{X}_{\mathrm{E}} = \mathtt{H}_{\mathrm{EP}}\mathbf{X} \quad \mathbf{X}'_{\mathrm{E}} = \mathtt{H}_{\mathrm{EP}}\mathbf{X}' \tag{19.33}$$

where $\mathtt{H}_{\mathrm{EP}}$ is a non-singular 4×4 matrix that relates projective to metric structure. An essential point to note here is that the two projective reconstructions, before and after the camera must be in the same projective coordinate frame, in other words, the same pair of cameras matrices must be used before and after.

From (19.32) and (19.33) if follows that

$$\mathtt{H}_{\mathrm{P}} = \mathtt{H}_{\mathrm{EP}}^{-1}\,\mathtt{H}_{\mathrm{E}}\,\mathtt{H}_{\mathrm{EP}} \tag{19.34}$$

so that $\mathtt{H}_{\mathrm{P}}$ is conjugate to a Euclidean transformation. There are two important properties of this conjugacy relation:

(i) $\mathtt{H}_{\mathrm{P}}$ and $\mathtt{H}_{\mathrm{E}}$ have the same eigenvalues.

(ii) If $\mathbf{E}$ is an eigenvector of $\mathtt{H}_{\mathrm{E}}$ then the corresponding eigenvector of $\mathtt{H}_{\mathrm{P}}$, with the same eigenvalue, is $(\mathtt{H}_{\mathrm{EP}}^{-1}\mathbf{E})$, i.e. the eigenvectors of $\mathtt{H}_{\mathrm{E}}$ are mapped to the eigenvectors of $\mathtt{H}_{\mathrm{P}}$ by the point transformation (19.33). This follows from (19.34), for if $\mathtt{H}_{\mathrm{E}}\mathbf{E} = \lambda\mathbf{E}$ then $\mathtt{H}_{\mathrm{EP}}\mathtt{H}_{\mathrm{P}}\mathtt{H}_{\mathrm{EP}}^{-1}\mathbf{E} = \lambda\mathbf{E}$, and pre-multiplying by $\mathtt{H}_{\mathrm{EP}}^{-1}$ gives the desired result.

Fixed points of a Euclidean transformation. Consider the Euclidean transformation represented by the matrix

$$\mathtt{H}_{\mathrm{E}} = \begin{bmatrix} \mathtt{R} & \mathtt{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This is a rotation by θ about the Z-axis together with a unit translation along the Z-axis (it is a general screw motion). The eigenvectors of $\mathtt{H}_{\mathrm{E}}$ are the fixed points under the transformation (refer to section 2.9(*p*61)). In this case the eigenvalues are $\{e^{i\theta}, e^{-i\theta}, 1, 1\}$ and the corresponding eigenvectors of $\mathtt{H}_{\mathrm{E}}$ are

$$\mathbf{E}_1 = \begin{pmatrix} 1 \\ i \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{E}_2 = \begin{pmatrix} 1 \\ -i \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{E}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{E}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

All the eigenvectors lie on $\boldsymbol{\pi}_\infty$. This means that $\boldsymbol{\pi}_\infty$ is fixed as a set, but is *not* a plane of fixed points. The eigenvectors $\mathbf{E}_1$ and $\mathbf{E}_2$ are the circular points for planes perpendicular to the Z (rotation) axis. The other two (identical) eigenvectors $\mathbf{E}_3$ and $\mathbf{E}_4$ are the direction of the rotation axis.

Computing $\boldsymbol{\pi}_\infty$. If the point transformation matrix is $\mathtt{H}_{\mathrm{E}}$ then from (3.6–*p*68) the plane transformation matrix is $\mathtt{H}_{\mathrm{E}}^{-\mathsf{T}}$. The eigenvectors of $\mathtt{H}_{\mathrm{E}}^{-\mathsf{T}}$ are the fixed planes under the motion. The matrix $\mathtt{H}_{\mathrm{E}}^{-\mathsf{T}}$ also has two equal, unit eigenvalues and a single eigenvector corresponding to these which is the plane $\boldsymbol{\pi}_\infty$ as may easily be verified. The eigenvectors of $\mathtt{H}_{\mathrm{E}}^{-\mathsf{T}}$ are mapped to those of $\mathtt{H}_{\mathrm{P}}^{-\mathsf{T}}$, in the same manner as the mapping of eigenvectors of $\mathtt{H}_{\mathrm{E}}$ to those of $\mathtt{H}_{\mathrm{P}}$ described above. Consequently, $\boldsymbol{\pi}_\infty$ in the projective reconstruction is the eigenvector corresponding to the (double) real eigenvalue of $\mathtt{H}_{\mathrm{P}}^{-\mathsf{T}}$. Thus,

- $\boldsymbol{\pi}_\infty$ *may be computed uniquely as the real eigenvector of* $\mathtt{H}_{\mathrm{P}}^{-\mathsf{T}}$*, or equivalently, and more simply, as the real eigenvector of* $\mathtt{H}_{\mathrm{P}}^{\mathsf{T}}$.

We observe here that although the real eigenvalue has algebraic multiplicity of two, its geometric multiplicity (in the case of non-planar motions) is one. This is what enables us to find the plane at infinity.

The procedure for affine calibration is summarized in algorithm 19.5.

Metric calibration and ambiguities. Once $\boldsymbol{\pi}_\infty$ is identified, the metric calibration may proceed as described in the stratified algorithm 19.2(*p*479). Since the rig is fixed, the parameters of the left camera are unchanged during the motion (and similarly for the right camera). From a single motion the internal parameters of each camera are determined up to the one-parameter family resulting from a single rotation as described in section 19.5.3.

As usual the ambiguity from a single motion is removed by additional motions or

Objective

Given two (or more) stereo pairs of images acquired by a fixed stereo rig undergoing general motions (i.e. both $\mathtt{R}$ and $\mathbf{t}$ are non-zero, and $\mathbf{t}$ not perpendicular to the axis of $\mathtt{R}$), compute an affine reconstruction.

Algorithm

(i) **Compute an initial projective reconstruction $\mathbf{X}$:** Using the first stereo pair compute a projective reconstruction $(\mathtt{P}^{\mathrm{L}}, \mathtt{P}^{\mathrm{R}}, \{\mathbf{X}_j\})$ as described in chapter 10. This involves computing the fundamental matrix $\mathtt{F}$ and point correspondences between the images of the first pair $x_j^{\mathrm{L}} \leftrightarrow x_j^{\mathrm{R}}$, e.g. use algorithm 11.4(*p*291).

(ii) **Compute a projective reconstruction $\mathbf{X}'$ after the motion:** Compute correspondences between the images of the second stereo pair $x_j'^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{R}}$. Since both the internal and relative external parameters of the cameras are fixed, the second stereo pair has the same fundamental matrix $\mathtt{F}$ as the first. The same camera matrices $\mathtt{P}^{\mathrm{L}}, \mathtt{P}^{\mathrm{R}}$ are used for triangulating points $\mathbf{X}'_j$ in 3-space from the computed correspondences $x_j'^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{R}}$ in the second stereo pair.

(iii) **Compute the 4×4 matrix $\mathtt{H}_{\mathrm{P}}$ which relates $\mathbf{X}$ to $\mathbf{X}'$:** Compute correspondences between the left images of the two stereo pairs $x_j^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{L}}$ (e.g. again use algorithm 11.4(*p*291)). This establishes correspondences between the space points $\mathbf{X}_j \leftrightarrow \mathbf{X}'_j$. The homography $\mathtt{H}_{\mathrm{P}}$ may be estimated linearly from five or more of these 3-space point correspondences, and then the estimate refined by minimizing a suitable cost function over $\mathtt{H}_{\mathrm{P}}$. For example, minimizing $\sum_j (d(x_j^{\mathrm{L}}, \mathtt{P}^{\mathrm{L}}\mathtt{H}\mathbf{X}'_j)^2 + d(x_j^{\mathrm{r}}, \mathtt{P}^{\mathrm{R}}\mathtt{H}\mathbf{X}'_j)^2)$ minimizes the distance between the measured and reprojected image points.

(iv) **Affine reconstruction:** Compute $\boldsymbol{\pi}_\infty$ from the real eigenvector of $\mathtt{H}_{\mathrm{P}}^{\mathsf{T}}$ and thence an affine reconstruction.

Algorithm 19.5. *Affine calibration of a fixed stereo rig.*

by supplying additional constraints, such as that the pixels are square. If there are additional motions then an improved estimate of $\boldsymbol{\pi}_\infty$ may also be computed. The outcome of metric calibration is the complete calibration of the rig (i.e. the relative external orientation of the cameras and their internal parameters).

Planar motion. In the special case of orthogonal (planar) motion, where the translation is orthogonal to the rotation axis direction, the space of eigenvectors corresponding to the repeated (real) eigenvalue is two-dimensional. Consequently, $\boldsymbol{\pi}_\infty$ is determined only up to a one-parameter family. We are therefore unable to find the plane at infinity uniquely (this is examined in detail in example 3.8(*p*81)). The ambiguity may be resolved by a second orthogonal motion about an axis with a different direction from the first.

Example 19.13. Auto-calibration from two stereo pairs

Figure 19.9(a)(b) shows the two stereo pairs used for the affine calibration of the stereo rig following the procedure of algorithm 19.5. The accuracy of the calibration is assessed by computing a vanishing point in the right image in two ways: first, as the intersection of imaged parallel lines; second, by determining the corresponding vanishing point in the left image (from images of the same parallel lines), and mapping this vanishing point to the right image using the infinite homography computed from

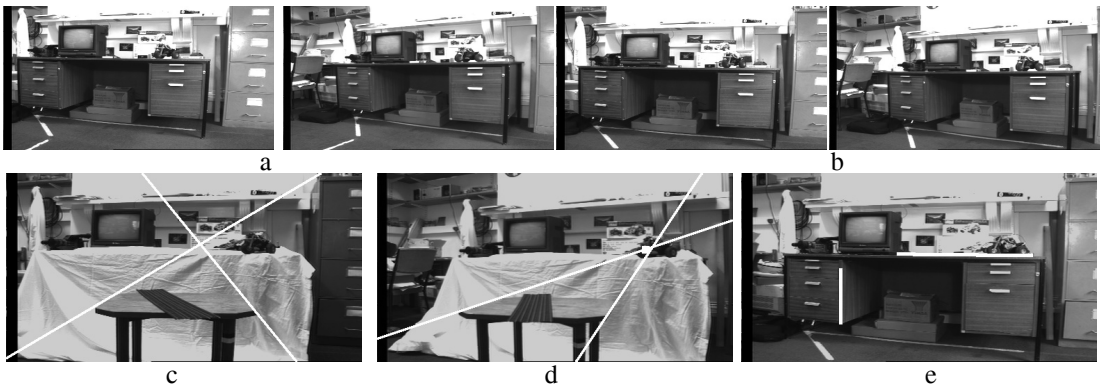

Fig. 19.9. **Auto-calibration of a stereo rig.** *The input stereo pairs before (a) and after (b) the motion of the rig. The stereo rig moves left by about 20 cm, pans by about* 10° *and changes elevation by about* 10°*. The accuracy of the computed* $\mathtt{H}_\infty$ *is assessed on another stereo pair acquired by the same rig as follows: In (c), the left image (of a stereo pair), a vanishing point is computed by intersecting the imaged sides of the table (which are parallel in the scene). In (d), the right image (of a stereo pair), the corresponding vanishing point is computed. The white square (near the line intersection) is the vanishing point from the left image mapped to the right using the computed* $\mathtt{H}_\infty$*. In the absence of error the points should be identical. (e) Following metric calibration the computed angle between the desk sides (shown in white) from the 3D reconstruction is* 90.7°*, in very good agreement with the veridical value.*

the eigenvector of $\mathtt{H}_{\mathrm{P}}^{-\mathsf{T}}$. The discrepancy between the vanishing points is a measure of the accuracy of the computed $\mathtt{H}_\infty$. The metric calibration uses the zero skew constraint to resolve the one-parameter ambiguity. Angles in the resulting metric reconstruction are accurate to within 1°. △

19.11 Closure

Research in the area of auto-calibration is still quite active, and better methods than those described in this chapter may yet be developed. There is still a lack of closed form solutions from multiview tensors, and of algorithms to automatically detect critical motion sequences (see below).

Critical motion sequences. It has been seen in this chapter that for certain classes of motion it is not possible to completely determine the rectifying homography $\mathtt{H}$. The resulting reconstruction is then at some level between metric and projective. For example, for constant internal parameters in the case of planar motion there is a one-parameter scaling ambiguity parallel to the rotation axis; and for pure translation under constant internal parameters the reconstruction is affine. Sequences of camera motions for which such ambiguities arise are termed "Critical motion sequences" and have been systematically classified by Sturm [Sturm-97a, Sturm-97b] in the case of constant internal parameters. This classification has been extended to more general calibration constraints, such as varying focal lengths [Pollefeys-99b, Sturm-99b]. For recent work see [Kahl-99, Kahl-01b, Ma-99, Sturm-01].

Recommendations. It may seem that auto-calibration offers a complete solution to metric reconstruction. Calibrated cameras are not necessary and we can do with constraints as weak as the zero-skew constraint on the camera. Unfortunately one must be wary of putting complete trust in auto-calibration. Auto-calibration can work well in the right circumstances, but used recklessly it will fail. Several specific recommendations can be made.

(i) Take care to avoid ambiguous motion sequences. It has been seen that calibration degeneracies occur if the motion is too restricted, such as being about a single axis. The motion should not be too small, or cover too small a field of view. Auto-calibration often comes down to estimating the infinite homography, the effects of which are not apparent on small fields of view.

(ii) Use as much information as you have. Although it is possible to calibrate from minimal information such as zero skew, this should be avoided if other information is available. For instance the known aspect ratio constraint should be used if it is valid, as should the knowledge of the principal point. Even if the values are known only imprecisely, this information can be incorporated into a linear auto-calibration method by including an equation, but with low weight.

(iii) This recommendation relates to bundle adjustment as well. Generally it is best to finish off with a bundle adjustment. In doing this, it is recommended that the internal parameters of the camera not be left to float unbounded. For instance, even if the principal point is not known exactly, it is usually known within some reasonable bounds (it is not close to infinity for instance). Similarly, aspect ratio normally lies between 0.5 and 3. This knowledge should be incorporated in a bundle adjustment by adding further constraints to the cost function, with small weights (standard deviations) if necessary. This can give an enormous improvement in results where auto-calibration is poorly conditioned (and hence unstable) by preventing the solution from wandering off into remote regions of parameter space in quest of a minor and insignificant improvement in the cost function.

(iv) Methods that use restricted motions usually are more reliable than those that allow general motion. For instance the methods that involve a rotating but non-translating camera are generally much more reliable than general motion methods. The same is true of affine reconstruction from a translational motion.

19.11.1 The literature

The idea of auto calibrating a camera originated in Faugeras *et al.* [Faugeras-92a] where the Kruppa equations were used. The early papers considered the case of constant internal parameters. [Hartley-94b] and Mohr *et al.* [Mohr-93] investigated bundle-adjustment like methods for more than two views.

The affine reconstruction solution for the case of pure translation was given by Moons *et al.* [Moons-94], and was extended to a combination of pure translation followed by rotation for a full metric reconstruction by Armstrong *et al.* [Armstrong-94]. The case of auto-calibration for a camera rotating about its centre

was given by [Hartley-94a]. The modulus constraint was first published by Pollefeys *et al.* [Pollefeys-96].

The original method for auto-calibration of a stereo rig was given by Zisserman *et al.* [Zisserman-95b], with alternative parametrizations given in Devernay and Faugeras [Devernay-96], and Horaud and Csurka [Horaud-98]. The special case of planar motion of a stereo rig is covered in [Beardsley-95b, Csurka-98]. For planar motion of a monocular camera the original method was published by Armstrong *et al.* [Armstrong-96b], and an alternative numerical solution was given in Faugeras *et al.* [Faugeras-98].

In more recent papers less restrictive constraints than constant internal parameters have been investigated. A number of "existence proofs" have been given: Heyden and Åström [Heyden-97b] showed that metric reconstruction is possible knowing only skew and aspect ratio, and [Pollefeys-98, Heyden-98] showed that skew-zero alone was sufficient.

Triggs [Triggs-97] introduced the absolute (dual) quadric as a numerical device for formulating auto-calibration problems, and applied both linear methods and sequential-quadratic programming to solve for $\mathtt{Q}^*_\infty$. Pollefeys *et al.* [Pollefeys-98] showed that computations based on $\mathtt{Q}^*_\infty$ could be used to compute metric reconstructions for varying focal length under general motion for real image sequences.

For the case of a rotating camera de Agapito *et al.* [DeAgapito-98] gave a non-linear solution for varying internal parameters based on the use of the DIAC. This was modified in [DeAgapito-99] to an IAC-based linear method.

19.11.2 Notes and exercises

(i) [Hartley-92a] first gave a solution for the extraction of focal lengths from the fundamental matrix, but the algorithm given there is unwieldy. A simple elegant formula is given in [Bougnoux-98]:

$$\alpha^2 = -\frac{\mathbf{p}'^{\mathsf{T}}[\mathbf{e}']_\times \tilde{\mathtt{I}}\mathtt{F}\mathbf{p}\mathbf{p}^{\mathsf{T}}\mathtt{F}^{\mathsf{T}}\mathbf{p}'}{\mathbf{p}'^{\mathsf{T}}[\mathbf{e}']_\times \tilde{\mathtt{I}}\mathtt{F}\tilde{\mathtt{I}}\mathtt{F}^{\mathsf{T}}\mathbf{p}'} \tag{19.35}$$

where $\tilde{\mathtt{I}}$ is the matrix $\mathrm{diag}(1,1,0)$ and $\mathbf{p}$ and $\mathbf{p}'$ are the principal points in the two images. Unit aspect ratio and zero skew are assumed. The formula for α'^2 is given by reversing the roles of the two images (and transposing $\mathtt{F}$).

Note that the final step of the algorithm is to take a square root. It is assured that α^2 and α'^2 as computed by (19.35) are positive, *given good data and a good guess of the principal point*. However in practice this does not always pertain, and negative values can result. This is the same problem as mentioned previously in section 19.3.5(*p*468). In addition, as flagged in [Newsam-96], the method has an intrinsic degeneracy when the principal rays of the two cameras meet in space, in which case it is impossible to compute the focal lengths independently. This occurs when both cameras are trained on the same point, quite a common occurrence.

A further degeneracy occurs when the plane defined by the baseline and the

principal ray of one camera is perpendicular to the plane defined by the baseline and principal ray of the other camera. Generally speaking, our opinion is that this method is of doubtful value as a means of computing focal lengths.

(ii) Show that if the internal parameters are constant then the constraints on $\mathtt{Q}^*_\infty$ obtained from two views (19.6–*p*462) are equivalent to the Kruppa equations (19.18–*p*471). Hint, from (9.10–*p*256) the cameras may be chosen as $\mathtt{P}^1 = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}^2 = [[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{e}']$.

(iii) Show from (19.21–*p*473) that in the case of a camera translating with constant internal parameters and without rotating, then the plane at infinity may be computed directly from a projective reconstruction.

(iv) The infinite homography relation (19.24–*p*476) may be derived in two lines simply from the definition $\mathtt{H}^{ij}_\infty = \mathtt{K}^i\mathtt{R}^{ij}(\mathtt{K}^j)^{-1}$, in section 13.4(*p*338). This may be rearranged as $\mathtt{H}^{ij}_\infty\mathtt{K}^j = \mathtt{K}^i\mathtt{R}^{ij}$. Eliminating the rotation using orthogonality as $\mathtt{R}^{ij}\mathtt{R}^{ij\mathsf{T}} = \mathtt{I}$ gives $\mathtt{H}^{ij}_\infty(\mathtt{K}^j\mathtt{K}^{j\mathsf{T}})\mathtt{H}^{ij\mathsf{T}}_\infty = (\mathtt{K}^i\mathtt{K}^{i\mathsf{T}})$.

(v) Under $\mathtt{H}_\mathrm{E}$, points on $\boldsymbol{\pi}_\infty$ (i.e. with $\mathrm{X}_4 = 0$) are mapped to points on $\boldsymbol{\pi}_\infty$ by the 3×3 homography $\mathbf{x}_\infty \mapsto \mathtt{R}\mathbf{x}_\infty$. Under this point transformation a conic on $\boldsymbol{\pi}_\infty$ maps according to result 2.13(*p*37) $\mathtt{C} \mapsto \mathtt{R}^{-\mathsf{T}}\mathtt{C}\mathtt{R}^{-1} = \mathtt{R}\mathtt{C}\mathtt{R}^{\mathsf{T}}$. The absolute conic Ω_∞ is fixed since $\mathtt{R}\mathtt{I}\mathtt{R}^{\mathsf{T}} = \mathtt{I}$. Now, denote by $\mathbf{a}$ the direction of the rotation axis, so that $\mathtt{R}\mathbf{a} = 1\mathbf{a}$. The (degenerate) point conic $\mathbf{a}\mathbf{a}^{\mathsf{T}}$ is also fixed. It follows that there is a pencil of fixed conics $\mathtt{C}_\infty(\mu) = \mathtt{I} + \mu\mathbf{a}\mathbf{a}^{\mathsf{T}}$ under the mapping since

$$\begin{aligned}\mathtt{R}(\Omega_\infty + \mu\mathbf{a}\mathbf{a}^{\mathsf{T}})\mathtt{R}^{\mathsf{T}} &= \mathtt{R}\mathtt{I}\mathtt{R}^{\mathsf{T}} + \mu\mathtt{R}\mathbf{a}\mathbf{a}^{\mathsf{T}}\mathtt{R}^{\mathsf{T}} \\ &= \Omega_\infty + \mu\mathbf{a}\mathbf{a}^{\mathsf{T}}.\end{aligned}$$

The scalar μ parametrizes the pencil. This shows that under a particular similarity there is a one-parameter family of fixed conics on $\boldsymbol{\pi}_\infty$. However, it is the case that Ω_∞ is the only conic fixed under any similarity.

(vi) A further calibration ambiguity exists for a common type of robot head, namely a pan-tilt (or alt-azimuth) head. In [DeAgapito-99] it was shown that since the camera may rotate about its X or Y axes its set of orientations form only a 2-parameter family, rather than a 3-parameter family of general rotations. This limitation causes an ambiguity in the aspect ratio α_x of the camera, and consequently of x_0 as well.

(vii) The method of auto-calibration from planes is generally non-linear. However, for special motions linear constraints on $\boldsymbol{\omega}$ can be obtained. Suppose we have two images of a plane that induces a planar homography H between views, and imagine that the motion of the camera relative to the plane is a general screw motion but with the screw axis parallel to the plane normal.

Consider the action of this screw motion on the plane. Since this action (a rotation about the plane's normal and a translation) does not change the plane's orientation, its line of intersection with the plane at infinity is unchanged (as a set). The absolute conic is fixed (as a set) under Euclidean motion. Consequently, the plane's intersection with the absolute conic, which defines the circular points for the plane, is also unchanged under the motion.

Consider now applying the action to the camera viewing the plane. Since the two circular points are fixed (as 3-space points) they have the same image before and after the motion. As the homography H maps points on the plane between images, the imaged circular points must correspond to two of its fixed points (see section 2.9(*p*61)) and can thus be determined directly from the homography. Each circular points places a linear constraint on ω. Further details of this method are given in [Knight-03].