16

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

This chapter describes numerical methods for estimating the trifocal tensor given a set of point and line correspondences across three views. The development will be very similar to that for the fundamental matrix, using much the same techniques as those of chapter 11. In particular, five methods will be discussed:

(i) A linear method based on direct solution of a set of linear equations (after appropriate data normalization) (section 16.2).
(ii) An iterative method, that minimizes the algebraic error, while satisfying all appropriate constraints on the tensor (section 16.3).
(iii) An iterative method that minimizes geometric error (the "Gold Standard" method) (section 16.4.1).
(iv) An iterative method that minimizes the Sampson approximation to geometric error (section 16.4.3).
(v) Robust estimation based on RANSAC (section 16.6).

16.1 Basic equations

A complete set of the (tri-)linear equations involving the trifocal tensor is given in table 16.1. All of these equations are linear in the entries of the trifocal tensor $\mathcal{T}$.

Correspondence	Relation	Number of equations
three points	$x^i x'^j x''^k \epsilon_{jqs} \epsilon_{krt} \mathcal{T}_i^{qr} = 0_{st}$	4
two points, one line	$x^i x'^j l''_r \epsilon_{jqs} \mathcal{T}_i^{qr} = 0_s$	2
one point, two lines	$x^i l'_q l''_r \mathcal{T}_i^{qr} = 0$	1
three lines	$l_p l'_q l''_r \epsilon^{piw} \mathcal{T}_i^{qr} = 0^w$	2

Table 16.1. **Trilinear relations between point and line coordinates in three views.** *The final column denotes the number of linearly independent equations. The notation 0_{st} means a 2-dimensional tensor with all zero entries. Thus, the first line in this table corresponds to a set of 9 equations, one for each choice of s and t. However, among this set of 9 equations, only 4 are linearly independent.*

Given several point or line correspondences between three images, the complete set of equations generated is of the form $\mathtt{A}\mathbf{t} = \mathbf{0}$, where $\mathbf{t}$ is the 27-vector made up of the entries of the trifocal tensor. From these equations, one may solve for the entries of the tensor. Note that equations involving points may be combined with those involving lines – in general all available equations from table 16.1 may be used simultaneously. Since $\mathcal{T}$ has 27 entries, 26 equations are needed to solve for $\mathbf{t}$ up to scale. With more than 26 equations, a least-squares solution is computed. As with the fundamental matrix, one minimizes $\|\mathtt{A}\mathbf{t}\|$ subject to the constraint $\|\mathbf{t}\| = 1$ using algorithm A5.4(*p*593).

This gives a bare outline of a linear algorithm for computing the trifocal tensor. However, in order to build a practical algorithm out of this several issues, such as normalization, need to be addressed. In particular the tensor that is estimated must obey various constraints, and we consider these next.

16.1.1 The internal constraints

The most notable difference between the fundamental matrix and the trifocal tensor is the greater number of constraints that apply to the trifocal tensor. The fundamental matrix has a single constraint, namely $\det(\mathtt{F}) = 0$, leaving 7 degrees of freedom, discounting the arbitrary scale factor. The trifocal tensor, on the other hand, has 27 entries, but 18 parameters only are required to specify the equivalent camera configuration, up to projectivity. The elements of the tensor therefore satisfy 8 independent algebraic constraints. This condition is conveniently stated as follows.

Definition 16.1. A trifocal tensor $\mathcal{T}_i^{jk}$ is said to be "geometrically valid" or "satisfy all internal constraints" if there exist three camera matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}'$ and $\mathtt{P}''$ such that $\mathcal{T}_i^{jk}$ corresponds to the three camera matrices according to (15.9–*p*376).

Just as with the fundamental matrix it is important to enforce these constraints in some way so as to arrive at a geometrically valid trifocal tensor. If the tensor does not satisfy the constraints, there are consequences similar to a fundamental matrix which is not of rank 2 – where epipolar lines, computed as $\mathtt{F}\mathbf{x}$ for varying $\mathbf{x}$, do not intersect in a single point (see figure 11.1(*p*280)). For example, if the tensor does not satisfy the internal constraints and is used to transfer a point to a third view, given a correspondence over two views as described in section 15.3, then the position of the transferred point will vary depending on which set of equations from table 16.1 is used. In the following the objective is always to estimate a geometrically valid tensor.

The constraints satisfied by the trifocal tensor elements are not so simply expressed (as $\det = 0$), and some have thought this an impediment to accurate computation of the trifocal tensor. However, in reality, in order to work with or compute the trifocal tensor it is not necessary to express these constraints explicitly – rather they are implicitly enforced by an appropriate parametrization of the trifocal tensor, and rarely cause any trouble. We will return to the issue of parametrization in section 16.3 and section 16.4.2.

16.1.2 The minimum case – 6 point correspondences

A geometrically valid trifocal tensor may be computed from images of a 6 point configuration, provided the scene points are in general position. There are one or three real solutions. The tensor is computed from the three camera matrices which are obtained using algorithm 20.1(*p*511), as described in section 20.2.4(*p*510). This minimal six point solution is used in the robust algorithm of section 16.6.

16.2 The normalized linear algorithm

In forming the matrix equation $\mathtt{A}\mathbf{t} = \mathbf{0}$ from the equations on $\mathcal{T}$ in table 16.1 it is not necessary to use the complete set of equations derived from each correspondence, since not all of these equations are linearly independent. For instance in the case of a point–point–point correspondence (first row of table 16.1) all choices of s and t lead to a set of 9 equations, but only 4 of these equations are linearly independent, and these may be obtained by choosing two values for each of s and t, for instance 1 and 2. This point is discussed in more detail in section 17.7(*p*431).

The reader may verify that the three points equation obtained from table 16.1 for a given choice of s and t may be expanded as

$$x^k(x'^i x''^m T_k^{jl} - x'^j x''^m T_k^{il} - x'^i x''^l T_k^{jm} + x'^j x''^l T_k^{im}) = 0^{ijlm} \quad . \tag{16.1}$$

when $i, j \neq s$ and $l, m \neq t$. Equation (16.1) collapses for $i = j$ or $l = m$, and swapping i and j (or l and m) simply changes the sign of the equation. One choice of the four independent equations is obtained by setting $j = m = 3$, and letting i and l range freely. The coordinates x^3, x'^3 and x''^3 may be set to 1 to obtain a relationship between the observed image coordinates. Equation (16.1) then becomes

$$x^k(x'^i x''^l T_k^{33} - x''^l T_k^{i3} - x'^i T_k^{3l} + T_k^{il}) = 0. \tag{16.2}$$

The four different choices of $i, l = 1, 2$ give four different equations in terms of the observed image coordinates.

How to represent lines

The three lines correspondence equation of table 16.1 may be written in the form

$$l_i = l'_j l''_k \mathcal{T}_i^{jk},$$

where, as usual with homogeneous entities, the equality is up to scale. In the presence of noise, this relationship will only be approximately satisfied by the *measured* lines $\mathbf{l}$, $\mathbf{l}'$ and $\mathbf{l}''$, but will be satisfied exactly for three lines $\hat{\mathbf{l}}$, $\hat{\mathbf{l}}'$ and $\hat{\mathbf{l}}''$ that are close to the measured lines.

The question is whether two sets of homogeneous coordinates that differ by a small amount represent lines that are close to each other in some geometric sense. Consider the two vectors $\mathbf{l}_1 = (0.01, 0, 1)^\mathsf{T}$ and $\mathbf{l}_2 = (0, 0.01, 1)^\mathsf{T}$. Clearly as vectors they are not very different, and in fact $\|\mathbf{l}_1 - \mathbf{l}_2\|$ is small. On the other hand, $\mathbf{l}_1$ represents the line $x = 100$, and $\mathbf{l}_2$ represents the line $y = 100$. Thus in a geometric sense, these lines are totally different. Note that this problem is alleviated by scaling. If coordinates are

Objective

Given $n \geq 7$ image point correspondences across 3 images, or at least 13 line correspondences, or a mixture of point and line correspondences, compute the trifocal tensor.

Algorithm

(i) Find transformation matrices $\mathtt{H}$, $\mathtt{H}'$ and $\mathtt{H}''$ to apply to the three images.

(ii) Transform points according to $x^i \mapsto \hat{x}^i = \mathtt{H}^i_j x^j$, and lines according to $l_i \mapsto \hat{l}_i = (\mathtt{H}^{-1})^j_i l_j$. Points and lines in the second and third image transform in the same way.

(iii) Compute the trifocal tensor $\hat{\mathcal{T}}$ linearly in terms of the transformed points and lines using the equations in table 16.1 by solving a set of equation of the form $\mathtt{At} = \mathbf{0}$, using algorithm A5.4(*p*593).

(iv) Compute the trifocal tensor corresponding to the original data according to $\mathcal{T}_i^{jk} = \mathtt{H}^r_i (\mathtt{H}'^{-1})^j_s (\mathtt{H}''^{-1})^k_t \hat{\mathcal{T}}_r^{st}$.

Algorithm 16.1. *The normalized linear algorithm for computation of* $\mathcal{T}$.

scaled by a factor of 0.01, then the coordinates for the lines become $\mathbf{l}_1 = (1, 0, 1)^\mathsf{T}$ and $\mathbf{l}_2 = (0, 1, 1)^\mathsf{T}$, which are quite different.

Nevertheless, this observation indicates that care is needed when representing lines. Suppose one is given a correspondence between three lines $\mathbf{l}$, $\mathbf{l}'$ and $\mathbf{l}''$. Two points $\mathbf{x}_1$ and $\mathbf{x}_2$ lying on $\mathbf{l}$ are selected. Each of these points provides a correspondence $\mathbf{x}_s \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$, for $s = 1, 2$, between the three views, in the sense that there exists a 3D line that maps to $\mathbf{l}'$ and $\mathbf{l}''$ in the second and third images and to a line (namely $\mathbf{l}$) passing through $\mathbf{x}_s$ in the first image. Two equations of the form $x^i_s l'_j l''_k \mathcal{T}_i^{jk} = 0_s$ for $s = 1, 2$ result from these correspondences. In this way one avoids the use of lines in the first image, though not the other images. Often lines in images are defined naturally by a pair of points, possibly the two endpoints of the lines. Even lines that are defined as the best fit to a set of edge points in an image may be treated as if they were defined by just two points, as will be described in section 16.7.2.

Normalization

As in all algorithms of this type, it is necessary to carry out prenormalization of the input data before forming and solving the linear equation system. Subsequently, it is necessary to correct for this normalization to find the trifocal tensor for the original data. The recommended normalization is much the same as that given for the computation of the fundamental matrix. A translation is applied to each image such that the centroid of the points is at the origin, and then a scaling is applied so that the average (RMS) distance of the points from the origin is $\sqrt{2}$. In the case of lines, the transformation should be defined by considering each line's two endpoints (or some representative line points visible in the image). The transformation rule for the trifocal tensor under these normalizing transformations is given in section A1.2(*p*563). The normalized linear algorithm for computing $\mathcal{T}$ is summarized in algorithm 16.1.

This algorithm does not consider the constraints discussed in section 16.1.1 that should be applied to $\mathcal{T}$. These constraints ought to be enforced before the denormal-

ization step (final step) in the above algorithm. Methods of enforcing these constraints will be considered next.

16.3 The algebraic minimization algorithm

The linear algorithm 16.1 will give a tensor not necessarily corresponding to any geometric configuration, as discussed in section 16.1.1. The next task is to correct the tensor to satisfy all required constraints.

Our task will be to compute a geometrically valid trifocal tensor $\mathcal{T}_i^{jk}$ from a set of image correspondences. The tensor computed will minimize the algebraic error associated with the input data. That is, we minimize $\|\mathtt{A}\hat{\mathbf{t}}\|$ subject to $\|\hat{\mathbf{t}}\| = 1$, where $\hat{\mathbf{t}}$ is the vector of entries of a geometrically valid trifocal tensor. The algorithm is quite similar to the algebraic algorithm (section 11.3(*p*282)) for computation of the fundamental matrix. Just as with the fundamental matrix, the first step is the computation of the epipoles.

Retrieving the epipoles

Let $\mathbf{e}'$ and $\mathbf{e}''$ be the epipoles in the second and third images corresponding to (that is being images of) the first camera centre. Recall from result 15.4(*p*373) that the two epipoles $\mathbf{e}'$ and $\mathbf{e}''$ are the common perpendicular to the left (respectively right) null-vectors of the three $\mathtt{T}_i$. In principle then, the epipoles may be computed from the trifocal tensor using the algorithm outlined in algorithm 15.1(*p*375). However, in the presence of noise, this translates easily into an algorithm for computing the epipoles based on four applications of algorithm A5.4(*p*593).

(i) For each $i = 1, \ldots, 3$ find the unit vector $\mathbf{v}_i$ that minimizes $\|\mathtt{T}_i\mathbf{v}_i\|$, where $\mathtt{T}_i = \mathcal{T}_i^{\cdot\cdot}$. Form the matrix $\mathtt{V}$, the i-th row of which is $\mathbf{v}_i^\mathsf{T}$.
(ii) Compute the epipole $\mathbf{e}''$ as the unit vector that minimizes $\|\mathtt{V}\mathbf{e}''\|$.

The epipole $\mathbf{e}'$ is computed similarly, using $\mathtt{T}_i^\mathsf{T}$ instead of $\mathtt{T}_i$.

Algebraic minimization

Having computed the epipoles the next step is to determine the remaining elements of the camera matrices $\mathtt{P}', \mathtt{P}''$ from which the trifocal tensor can be calculated. This step is linear.

From the form (15.9–*p*376) of the trifocal tensor, it may be seen that once the epipoles $\mathbf{e}'^j = a_4^j$ and $\mathbf{e}''^k = b_4^k$ are known, the trifocal tensor may be expressed linearly in terms of the remaining entries of the matrices a_i^j and b_i^k. This relationship may be written linearly as $\mathbf{t} = \mathtt{E}\mathbf{a}$ where $\mathbf{a}$ is the vector of the remaining entries a_j^i and b_j^i, $\mathbf{t}$ is the vector of entries of the trifocal tensor, and $\mathtt{E}$ is the linear relationship expressed by (15.9–*p*376). We wish to minimize the algebraic error $\|\mathtt{A}\mathbf{t}\| = \|\mathtt{AE}\mathbf{a}\|$ over all choices of $\mathbf{a}$ constrained such that $\|\mathbf{t}\| = 1$, that is $\|\mathtt{E}\mathbf{a}\| = 1$. This minimization problem is solved by algorithm A5.6(*p*595). The solution $\mathbf{t} = \mathtt{E}\mathbf{a}$ represents a trifocal tensor satisfying all constraints, and minimizing the algebraic error, subject to the given choice of epipoles.

Objective

Given a set of point and line correspondences in three views, compute the trifocal tensor.

Algorithm

(i) From the set of point and line correspondences compute the set of equations of the form $\mathtt{A}\mathbf{t} = \mathbf{0}$, from the relations given in table 16.1.

(ii) Solve these equations using algorithm A5.4(p593) to find an initial estimate of the trifocal tensor $\mathcal{T}_i^{jk}$.

(iii) Find the two epipoles $\mathbf{e}'$ and $\mathbf{e}''$ from $\mathcal{T}_i^{jk}$ as the common perpendicular to the left (respectively right) null-vectors of the three $\mathtt{T}_i$.

(iv) Construct the 27×18 matrix $\mathtt{E}$ such that $\mathbf{t} = \mathtt{E}\mathbf{a}$ where $\mathbf{t}$ is the vector of entries of $\mathcal{T}_i^{jk}$, $\mathbf{a}$ is the vector representing entries of a_i^j and b_i^k, and where $\mathtt{E}$ expresses the linear relationship $\mathcal{T}_i^{jk} = a_i^j e''^k - e'^j b_i^k$.

(v) Solve the minimization problem: minimize $\|\mathtt{A}\mathtt{E}\mathbf{a}\|$ subject to $\|\mathtt{E}\mathbf{a}\| = 1$, using algorithm A5.6($p$595). Compute the error vector $\boldsymbol{\epsilon} = \mathtt{A}\mathtt{E}\mathbf{a}$.

(vi) **Iteration:** The mapping $(\mathbf{e}', \mathbf{e}'') \mapsto \boldsymbol{\epsilon}$ is a mapping from $\mathbb{R}^6$ to $\mathbb{R}^{27}$. Iterate on the last two steps with varying $\mathbf{e}'$ and $\mathbf{e}''$ using the Levenberg–Marquardt algorithm to find the optimal $\mathbf{e}', \mathbf{e}''$. Hence find the optimal $\mathbf{t} = \mathtt{E}\mathbf{a}$ containing the entries of $\mathcal{T}_i^{jk}$.

Algorithm 16.2. **Computing the trifocal tensor minimizing algebraic error.** *The computation should be carried out on data normalized in the manner of algorithm 16.1. Normalization and denormalization steps are omitted here for simplicity. This algorithm finds the geometrically valid trifocal tensor that minimizes algebraic error. At the cost of a slightly inferior solution, the last iteration step may be omitted, providing a fast non-iterative algorithm.*

Iterative method

The two epipoles used to compute a geometrically valid tensor $\mathcal{T}_i^{jk}$ are determined using the estimate of $\mathcal{T}_i^{jk}$ obtained from the linear algorithm. Analogous to the case of the fundamental matrix, the mapping $(\mathbf{e}', \mathbf{e}'') \mapsto \mathtt{A}\mathtt{E}\mathbf{a}$ is a mapping $\mathbb{R}^6 \rightarrow \mathbb{R}^{27}$. An application of the Levenberg–Marquardt algorithm to optimize the choice of the epipoles will result in an optimal (in terms of algebraic error) estimate of the trifocal tensor. Note that the iteration problem is of modest size, since only 6 parameters, the homogeneous coordinates of the epipoles, are involved in the iteration problem.

This contrasts with an iterative estimation of the optimal trifocal tensor in terms of geometric error, considered later. This latter problem requires estimating the parameters of the three cameras, plus the coordinates of all the points, a large estimation problem.

The complete algebraic method for estimating the trifocal tensor is summarized in algorithm 16.2.

16.4 Geometric distance

16.4.1 The Gold Standard method for the trifocal tensor

As with the computation of the fundamental matrix, best results may be expected from the maximum likelihood (or "Gold Standard") solution. Since this has been adequately described for the case of the fundamental matrix computation, little needs to be added for the three-view case.

Objective

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

The MLE involves also solving for a set of subsidiary point correspondences $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i \leftrightarrow \hat{\mathbf{x}}''_i\}$, which exactly satisfy the trilinear relations of the estimated tensor and which minimize

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

Algorithm

(i) Compute an initial geometrically valid estimate of $\mathcal{T}$ using a linear algorithm such as algorithm 16.2.

(ii) Compute an initial estimate of the subsidiary variables $\{\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i, \hat{\mathbf{x}}''_i\}$ as follows:

  (a) Retrieve the camera matrices $\mathtt{P}'$ and $\mathtt{P}''$ from $\mathcal{T}$.

  (b) From the correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ and $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}', \mathtt{P}''$ determine an estimate of $\widehat{\mathbf{X}}_i$ using the triangulation method of chapter 12.

  (c) The correspondence consistent with $\mathcal{T}$ is obtained as $\hat{\mathbf{x}}_i = \mathtt{P}\widehat{\mathbf{X}}_i,\ \hat{\mathbf{x}}'_i = \mathtt{P}'\widehat{\mathbf{X}}_i,\ \hat{\mathbf{x}}''_i = \mathtt{P}''\widehat{\mathbf{X}}_i$.

(iii) Minimize the cost

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

over $\mathcal{T}$ and $\widehat{\mathbf{X}}_i, i = 1, \ldots, n$. The cost is minimized using the Levenberg–Marquardt algorithm over $3n + 24$ variables: $3n$ for the n 3D points $\widehat{\mathbf{X}}_i$, and 24 for the elements of the camera matrices $\mathtt{P}', \mathtt{P}''$.

Algorithm 16.3. *The Gold Standard algorithm for estimating $\mathcal{T}$ from image correspondences.*

Given a set of point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$ in three views, the cost function to be minimized is

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 + d(\mathbf{x}''_i, \hat{\mathbf{x}}''_i)^2 \tag{16.3}$$

where the points $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i, \hat{\mathbf{x}}''_i$ satisfy a trifocal constraint (as in table 16.1) exactly for the estimated trifocal tensor. As in the case of the fundamental matrix one needs to introduce further variables corresponding to 3D points $\mathbf{X}_i$ and parametrize the trifocal tensor by the entries of the matrices $\mathtt{P}'$ and $\mathtt{P}''$ (see below). The cost function is then minimized over the position of the 3D points $\mathbf{X}_i$ and the two camera matrices $\mathtt{P}'$ and $\mathtt{P}''$ with $\hat{\mathbf{x}}_i = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_i,\ \hat{\mathbf{x}}'_i = \mathtt{P}'\mathbf{X}_i$, and $\hat{\mathbf{x}}''_i = \mathtt{P}''\mathbf{X}_i$. Essentially one is carrying out bundle adjustment over three views. The sparse matrix techniques of section A6.3-(*p*602) should be used.

A good way to find an initial estimate is the algebraic algorithm 16.2, though the final iterative step can be omitted. This algorithm gives a direct estimate of the entries of $\mathtt{P}'$ and $\mathtt{P}''$. The initial estimate of the 3D points $\mathbf{X}_i$ may be obtained using the linear triangulation method of section 12.2(*p*312). The steps of the algorithm are summarized in algorithm 16.3.

The technique can be extended to include line correspondences. To do this, one needs to find a representation of a 3D line convenient for computation. Given a 3-view line correspondence $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$, the lines being perhaps defined by their endpoints in each image, a very convenient way to represent the 3D line during the LM parameter minimization is by its projections $\hat{\mathbf{l}}'$ and $\hat{\mathbf{l}}''$ in the second and third views. Given a candidate trifocal tensor, one can easily compute the projection of the 3D line into the first view using the line transfer equation $\hat{l}_i = \hat{l}'_j \hat{l}''_k \mathcal{T}_i^{jk}$. Then one minimizes the sum-of-squares line distance

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

for some appropriate interpretation of the distance $d(\mathbf{l}'_i, \hat{\mathbf{l}}'_i)^2$ between the measured and estimated line. If the measured line is specified by its endpoints, then the obvious distance metric to use is the distance of the estimated line from the measured endpoints. In general a Mahalanobis distance may be used.

16.4.2 Parametrization of the trifocal tensor

If the tensor is parametrized simply by its 27 entries, then the estimated tensor will not satisfy the internal constraints. A parametrization which ensures that the tensor does satisfy its constraints, and so is geometrically valid, is termed *consistent*.

Since, from definition 16.1, a tensor is geometrically valid if it is generated from three camera matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}', \mathtt{P}''$ by (15.9–*p*376), it follows that the three camera matrices give a consistent parametrization. Note that this is an *over*parametrization since it requires 24 parameters to be specified, namely the 12 entries each of the matrices $\mathtt{P}' = [\mathtt{A}|\mathbf{a}_4]$ and $\mathtt{P}'' = [\mathtt{B}|\mathbf{b}_4]$. There is no need to attempt to define a minimal set of (18) parameters, which is a difficult task. Any choice of cameras is a consistent parametrization, the particular projective reconstruction has no effect on the tensor.

Another consistent parametrization is obtained by computing the tensor from six point correspondences across three views as in section 20.2(*p*508). Then the position of the points in each image is the parametrization – a total of 6 (points) $\times 2$ (for x, y) $\times$ 3 (images) $= 36$ parameters. However, only a subset of the points need be varied during the minimization, or the movement of the points can be restricted to be perpendicular to the variety of trifocal tensors.

16.4.3 First-order geometric error (Sampson distance)

The trifocal tensor may be computed using a geometric cost function based on the Sampson approximation in a manner entirely analogous to the Sampson method used to compute the fundamental matrix (section 11.4.3(*p*287)). Again the advantage is that it is not necessary to introduce a set of subsidiary variables, as this first-order geometric error requires a minimization only over the parametrization of the tensor (e.g. only 24 parameters if $\mathtt{P}', \mathtt{P}''$ is used as above). The minimization can be carried out with a simple iterative Levenberg–Marquardt algorithm, and the method initialized by the iterative algebraic algorithm 16.2.

The Sampson cost function is a little more complex computationally than the corresponding cost function for the fundamental matrix (11.9–*p*287), because each point correspondence gives four equations, instead of just one for the fundamental matrix. The more general case was discussed in section 4.2.6(*p*98). The error function (4.13–*p*100) in the present case is

$$\sum_i \boldsymbol{\epsilon}_i^{\mathsf{T}} (\mathtt{J}_i \mathtt{J}_i^{\mathsf{T}})^{-1} \boldsymbol{\epsilon}_i \tag{16.4}$$

where $\boldsymbol{\epsilon}_i$ is the algebraic error vector $\mathtt{A}_i\mathbf{t}$ corresponding to a single 3-view correspondence (a 4-vector in the case of 4 equations per point), and $\mathtt{J}$ is the 4×6 matrix of partial derivatives of $\boldsymbol{\epsilon}$ with respect to the coordinates of each of the corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$. As in the programming hint given in exercise (vii) on page 129, the computation of the partial derivative matrix $\mathtt{J}$ may be simplified by observing that the cost function is multilinear in the coordinates of the points $\mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}''_i$.

The Sampson error method has various advantages:

- It gives a good approximation to actual geometric error (the optimum), using a relatively simple iterative algorithm.
- As in the case of actual geometric error, non-isotropic and unequal error distributions may be specified for each of the points without significantly complicating the algorithm. See exercises in chapter 4.

16.5 Experimental evaluation of the algorithms

A brief comparison is now given of the results of the (iterative) algebraic algorithm 16.2 along with the Gold Standard algorithm 16.3 for computing the trifocal tensor. The algorithms are run on synthetic data with controlled levels of noise. This allows a comparison with the theoretically optimal ML results, and a determination of how well these algorithms are able to approximate the theoretical lower bound on residual error, achieved by an optimal ML algorithm.

Computer-generated data sets of 10, 15 and 20 points were used to test the algorithm, and the cameras were placed at random angles around the cloud of points. The camera parameters were chosen to approximate a standard 35mm camera, and the scale was chosen so that the size of the image was 600×600 pixels.

For a given level of added Gaussian noise in the image measurement, one may compute the expected residual achieved by an ML algorithm, according to result 5.2(*p*136). In this case, if n is the number of points, then the number of measurements is $N = 6n$, and the number of degrees of freedom in the fitting is $d = 18 + 3n$, where 18 represents the number of degrees of freedom of the three cameras (3×11 less 15 to account for projective ambiguity) and $3n$ represents the number of degrees of freedom of n points in space. Hence the ML residual is

$$\epsilon_{\text{res}} = \sigma(1 - d/N)^{1/2} = \sigma \left(\frac{n-6}{2n} \right)^{1/2}.$$

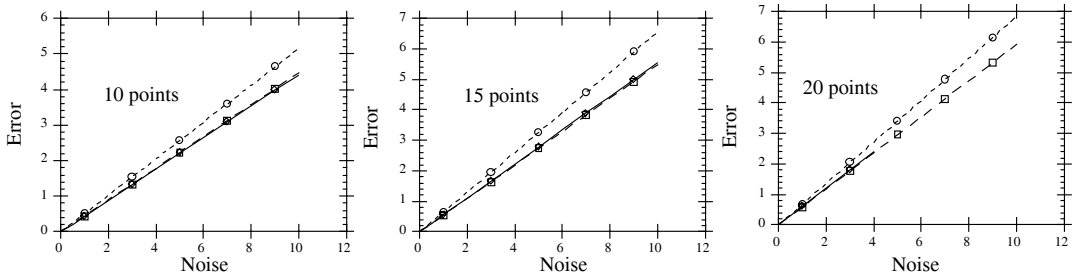

Fig. 16.1. **Comparison of trifocal tensor estimation algorithms.** *The residual error RMS-averaged over 100 runs is plotted against the noise level, for computation of the trifocal tensor using 10, 15 and 20 points. Each graph contains three curves. The top curve is the result of the algebraic error minimization, whereas the lower two curves, actually indistinguishable in the graphs, represent the theoretical minimum error, and the error obtained by the Gold Standard algorithm using the algebraic minimization as a starting point. Note that the residual errors are almost exactly proportional to added noise, as they should be.*

16.5.1 Results and recommendations

The results are shown in figure 16.1. We learn two things from these results. Minimization of the algebraic error achieves residual errors within about 15% of the optimal and using this estimate as a starting point for minimizing geometric error achieves a virtually optimal estimate.

All the algorithms developed above, except the linear method of section 16.1, enforce the internal constraints on the tensor. The linear method is not recommended for use on its own, but is necessary for initialization in most of the other methods. As in the case of estimating the fundamental matrix our recommendations are to use the iterative algebraic algorithm 16.2 or the Sampson geometric approximation of section 16.4.3. Both give excellent results. Again to be certain of getting the best results, if Gaussian noise is a viable assumption, implement the Gold Standard algorithm 16.3.

16.6 Automatic computation of $\mathcal{T}$

This section describes an algorithm to compute the trifocal geometry between three images automatically. The input to the algorithm is simply the triplet of images, with no other *a priori* information required; and the output is the estimated trifocal tensor together with a set of interest points in correspondence across the three images.

The fact that the trifocal tensor may be used to determine the exact image position of a point in a third view, given its image position in the other two views, means that there are fewer mismatches over three views than there are over two. In the two view case there is only the weaker geometric constraint of an epipolar line against which to verify a possible match.

The three-view algorithm uses RANSAC as a search engine in a similar manner to its use in the automatic computation of a homography described in section 4.8-(p123). The ideas and details of the algorithm are given there, and are not repeated here. The method is summarized in algorithm 16.4, with an example of its use shown

Objective Compute the trifocal tensor between three images.

Algorithm

(i) **Interest points:** Compute interest points in each image.
(ii) **Two-view correspondences:** Compute interest point correspondences (and F) between views 1 & 2, and 2 & 3 using algorithm 11.4(*p*291).
(iii) **Putative three-view correspondences:** Compute a set of interest point correspondences over three views by joining the two-view match sets.
(iv) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5(*p*121):
  (a) Select a random sample of 6 correspondences and compute the trifocal tensor using algorithm 20.1(*p*511). There will be one or three real solutions.
  (b) Calculate the distance $d_\perp$ in $\mathbb{R}^6$ from each putative correspondence to the variety described by $\mathcal{T}$, as in section 16.6.
  (c) Compute the number of inliers consistent with $\mathcal{T}$ by the number of correspondences for which $d_\perp < t$.
  (d) If there are three real solutions for $\mathcal{T}$ the number of inliers is computed for each solution, and the solution with most inliers retained.

  Choose the $\mathcal{T}$ with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.
(v) **Optimal estimation:** Re-estimate $\mathcal{T}$ from all correspondences classified as inliers using the Gold Standard algorithm 16.3 or the Sampson approximation to this.
(vi) **Guided matching:** Further interest point correspondences are now determined using the estimated $\mathcal{T}$ as described in the text.

The last two steps can be iterated until the number of correspondences is stable.

Algorithm 16.4. *Algorithm to automatically estimate the trifocal tensor over three images using RANSAC.*

in figure 16.2, and additional explanation of the steps given below. Figure 16.3 shows a second example which includes automatically computed line matches.

The distance measure – reprojection error. Given the match $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ and the current estimate of $\mathcal{T}$ we need to determine the minimum of the reprojection error – $d_\perp^2 = d^2(\mathbf{x}, \hat{\mathbf{x}}) + d^2(\mathbf{x}', \hat{\mathbf{x}}') + d^2(\mathbf{x}'', \hat{\mathbf{x}}'')$, where the image points $\hat{\mathbf{x}}, \hat{\mathbf{x}}', \hat{\mathbf{x}}''$ are consistent with $\mathcal{T}$. As usual the consistent images points may be obtained from the projection of a 3-space point $\widehat{\mathbf{X}}$

$$\hat{\mathbf{x}} = [\mathtt{I} \mid \mathbf{0}]\widehat{\mathbf{X}}, \quad \hat{\mathbf{x}}' = \mathtt{P}'\widehat{\mathbf{X}}, \quad \hat{\mathbf{x}}'' = \mathtt{P}''\widehat{\mathbf{X}}$$

where the camera matrices $\mathtt{P}', \mathtt{P}''$ are extracted from $\mathcal{T}$. The distance $d_\perp^2$ is then obtained by determining the point $\widehat{\mathbf{X}}$ which minimizes the image distance between the measured points $\mathbf{x}, \mathbf{x}', \mathbf{x}''$ and the projected points.

Another way of obtaining this distance is to use the Sampson error (16.4), which is a first-order approximation to the geometric error. However, in practice it is quicker to estimate the error directly by non-linear least-squares iteration (a small Levenberg–Marquardt problem). Starting from an initial estimate of $\widehat{\mathbf{X}}$, one iterates varying the coordinates of $\widehat{\mathbf{X}}$ to minimize the reprojection error.

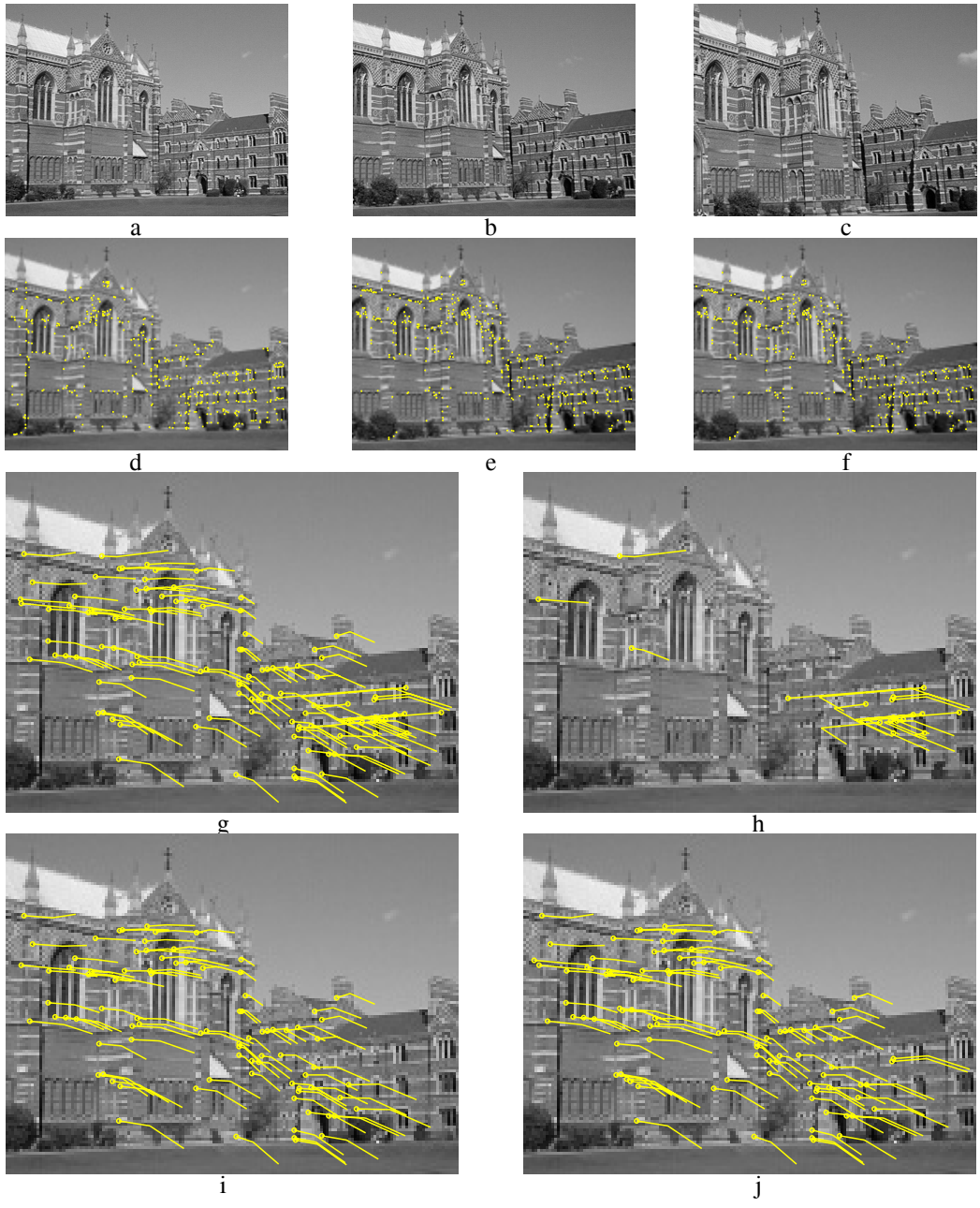

Fig. 16.2. **Automatic computation of the trifocal tensor between three images using RANSAC**. *(a - c) raw images of Keble College, Oxford. The motion between views consists of a translation and rotation. The images are* 640×480 *pixels. (d - f) detected corners superimposed on the images. There are approximately 500 corners on each image. The following results are superimposed on the (a) image: (g) 106 putative matches shown by the line linking corners, note the clear mismatches; (h) outliers – 18 of the putative matches. (i) inliers – 88 correspondences consistent with the estimated* $\mathcal{T}$*; (j) final set of 95 correspondences after guided matching and MLE. There are no mismatches.*

Guided matching. We have an initial estimate of $\mathcal{T}$ and wish to use this to generate and assess additional point correspondences across the three-views. The first step is to extract the fundamental matrix $\mathtt{F}_{12}$ between views 1 & 2 from $\mathcal{T}$. Then two-view

Fig. 16.3. **Image triplet matching.** *The trifocal tensor is computed automatically from interest points using algorithm 16.4, and subsequently used to match line features across views. (a) Three images of a corridor sequence. (b) Automatically matched line segments. The matching algorithm is described in [Schmid-97].*

guided matches are computed using loose thresholds on matching. Each two-view match is corrected using $\mathtt{F}_{12}$ to give points $\hat{\mathbf{x}}, \hat{\mathbf{x}}'$ which are consistent with $\mathtt{F}_{12}$. These corrected two-view matches (together with $\mathcal{T}$) define a small search window in the third view in which the corresponding point is sought. Any three-view point correspondence is assessed by computing $d_\perp$, as described above. The match accepted if $d_\perp$ is less than the threshold t. Note, the same threshold is used for inlier detection within the RANSAC stage and guided matching.

In practice it is found that the stage of guided matching is more significant here, in that it generates additional correspondences, than in the case of homography estimation.

Implementation and run details. For the example of figure 16.2, the search window was ± 300 pixels. The inlier threshold was $t = 1.25$ pixels. A total of 26 samples were required. The RMS pixel error after RANSAC was 0.43 (for 88 correspondences), after MLE it was 0.23 (for 88 correspondences), and after MLE and guided matching it was 0.19 (for 95 correspondences). The MLE required 10 iterations of the Levenberg–Marquardt algorithm.

Note, RANSAC has to do far less work than in algorithm 11.4(*p*291) to estimate $\mathtt{F}$ and correspondences, because the two-view algorithm has already removed many outliers before the putative correspondences over three views are generated.

16.7 Special cases of $\mathcal{T}$-computation

16.7.1 Computing $\mathcal{T}_i^{jk}$ from a plane plus parallax

We describe here the computation of $\mathcal{T}_i^{jk}$ from the image of a special configuration consisting of a world plane (from which a homography between views can be computed) and two points off the plane. Of course, it is not necessary for the plane to actually be present. It may be virtual, or the homography may simply be specified by the images of four coplanar points or four coplanar lines. The method is the analogue of algorithm 13.2(*p*336) for the fundamental matrix.

The solution is obtained by constructing the three camera matrices (up to a common projective transformation of 3-space) and then computing the trifocal tensor from these matrices according to (15.9–*p*376). The homography induced by the world (reference) plane between the first and second view is $\mathtt{H}_{12}$, and between the first and third views is $\mathtt{H}_{13}$. As shown in section 13.3(*p*334) the epipole $\mathbf{e}'$ may be computed directly from the two point correspondences off the plane for the first and second views, and the camera matrices chosen as $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}' = [\mathtt{H}_{12} \mid \mu\mathbf{e}']$, where μ is a scalar. Note the scale of both $\mathtt{H}_{12}$ and $\mathbf{e}'$ is considered fixed here, so they are no longer homogeneous quantities. Similarly, $\mathbf{e}''$ may be determined from the two point correspondences for views one and three and the camera matrices chosen as $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}'' = [\mathtt{H}_{13} \mid \lambda\mathbf{e}'']$, where λ is a scalar.

It is then easily verified that a consistent set of cameras for the three views (see the discussion on consistent camera triplets on page 375) is given by

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \quad \mathtt{P}' = [\mathtt{H}_{12} \mid \mathbf{e}'], \quad \mathtt{P}'' = [\mathtt{H}_{13} \mid \lambda\mathbf{e}''] \tag{16.5}$$

where μ has been set to unity. The value of λ is determined from one of the point correspondences over three views, and this is left as an exercise. For more on plane-plus-parallax reconstruction, see section 18.5.2(*p*450).

Note that the estimation of the trifocal tensor for this configuration is over-determined. In the case of the fundamental matrix over two views the homographies determine all but 2 degrees of freedom (the epipole), and each of the point correspondence provides one constraint, so that the number of constraints equals the number of degrees of freedom of the matrix. In the case of the trifocal tensor the homography determines all but 5 degrees of freedom (the two epipoles and their relative scaling). However, each point correspondence provides three constraints (there are six coordinate measurements less three for the point's position in 3-space), so that there are six constraints on 5 degrees of freedom. Since there are more measurements than degrees of freedom in this case, the tensor should be estimated by minimizing a cost function based on geometric error.

16.7.2 Lines specified by several points

In describing the reconstruction algorithm from lines, we have considered the case where lines are specified by their two endpoints. Another common way that lines may

be specified in an image is as the best line fit to several points. It will be shown now how that case may easily be reduced to the case of a line defined by two endpoints. Consider a set of points $\mathbf{x}_i$ in an image, normalized to have third component equal to 1. Let $\mathbf{l} = (l_1, l_2, l_3)^\mathsf{T}$ be a line, which we suppose is normalized such that $l_1^2 + l_2^2 = 1$. In this case, the distance from a point $\mathbf{x}_i$ to the line $\mathbf{l}$ is equal to $\mathbf{x}_i^\mathsf{T}\mathbf{l}$. The squared distance may be written as $d^2 = \mathbf{l}^\mathsf{T}\mathbf{x}_i\mathbf{x}_i^\mathsf{T}\mathbf{l}$, and the sum-of-squares of all distances is

$$\sum_i \mathbf{l}^\mathsf{T}\mathbf{x}_i\mathbf{x}_i^\mathsf{T}\mathbf{l} = \mathbf{l}^\mathsf{T}(\sum_i \mathbf{x}_i\mathbf{x}_i^\mathsf{T})\mathbf{l} \ .$$

The matrix $\mathtt{E} = (\sum_i \mathbf{x}_i\mathbf{x}_i^\mathsf{T})$ is positive-definite and symmetric.

Lemma 16.2. *Matrix* $(\mathtt{E} - \epsilon_0\mathtt{J})$ *is positive-semidefinite, where* $\mathtt{J}$ *is the matrix* $\mathrm{diag}(1, 1, 0)$ *and* ϵ_0 *is the smallest solution to the equation* $\det(\mathtt{E} - \epsilon\mathtt{J}) = 0$.

Proof. We start by computing the vector $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$ that minimizes $\mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x}$ subject to the condition $x_1^2 + x_2^2 = 1$. Using the method of Lagrange multipliers, this comes down to finding the extrema of $\mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x} - \xi(x_1^2 + x_2^2)$, where ξ denotes the Lagrange coefficient. Taking the derivative with respect to $\mathbf{x}$ and setting it to zero, we find that $2\mathtt{E}\mathbf{x} - \xi(2x_1, 2x_2, 0)^\mathsf{T} = \mathbf{0}$. This may be written as $(\mathtt{E} - \xi\mathtt{J})\mathbf{x} = \mathbf{0}$. It follows that ξ is a root of the equation $\det(\mathtt{E} - \xi\mathtt{J}) = 0$ and $\mathbf{x}$ is the generator of the null-space of $\mathtt{E} - \xi\mathtt{J}$. Since $\mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x} = \xi\mathbf{x}^\mathsf{T}\mathtt{J}\mathbf{x} = \xi(x_1^2 + x_2^2) = \xi$, it follows that to minimize $\mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x}$ one must choose ξ to be the minimum root ξ_0 of the equation $\det(\mathtt{E} - \xi\mathtt{J}) = 0$. In this case one has $\mathbf{x}_0^\mathsf{T}\mathtt{E}\mathbf{x}_0 - \xi_0 = 0$ for the minimizing vector $\mathbf{x}_0$. For any other vector $\mathbf{x}$, not necessarily the minimizing vector, one has $\mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x} - \xi_0 \geq 0$. Then, $\mathbf{x}^\mathsf{T}(\mathtt{E} - \xi_0\mathtt{J})\mathbf{x} = \mathbf{x}^\mathsf{T}\mathtt{E}\mathbf{x} - \xi_0 \geq 0$, and so $\mathtt{E} - \xi_0\mathtt{J}$ is positive-semidefinite. □

Since the matrix $\mathtt{E} - \xi_0\mathtt{J}$ is symmetric it may be written in the form $\mathtt{E} - \xi_0\mathtt{J} = \mathtt{V}\mathrm{diag}(r, s, 0)\mathtt{V}^\mathsf{T}$ where $\mathtt{V}$ is an orthogonal matrix and r and s are positive. It follows that

$$\begin{aligned} \mathtt{E} - \xi_0\mathtt{J} &= \mathtt{V}\mathrm{diag}(r, 0, 0)\mathtt{V}^\mathsf{T} + \mathtt{V}\mathrm{diag}(0, s, 0)\mathtt{V}^\mathsf{T} \\ &= r\mathbf{v}_1\mathbf{v}_1^\mathsf{T} + s\mathbf{v}_2\mathbf{v}_2^\mathsf{T} \end{aligned}$$

where $\mathbf{v}_i$ is the i-th column of $\mathtt{V}$. Therefore $\mathtt{E} = \xi_0\mathtt{J} + r\mathbf{v}_1\mathbf{v}_1^\mathsf{T} + s\mathbf{v}_2\mathbf{v}_2^\mathsf{T}$. Then for any line $\mathbf{l}$ satisfying $l_1^2 + l_2^2 = 1$ we have

$$\begin{aligned} \sum_i (\mathbf{x}_i^\mathsf{T}\mathbf{l})^2 &= \mathbf{l}^\mathsf{T}\mathtt{E}\mathbf{l} \\ &= \xi_0 + r(\mathbf{v}_1^\mathsf{T}\mathbf{l})^2 + s(\mathbf{v}_2^\mathsf{T}\mathbf{l})^2. \end{aligned}$$

Thus, we have replaced the sum-of-squares of several points by a constant value ξ_0, which is not capable of being minimized, plus the weighted sum-of-squares of the distances to two points $\mathbf{v}_1$ and $\mathbf{v}_2$. To summarize: when forming the trifocal tensor equations involving a line defined by points $\mathbf{x}_i$, formulate two point equations expressed in terms of the points $\mathbf{v}_1$ and $\mathbf{v}_2$ with weights $\sqrt{r}$ and $\sqrt{s}$ respectively.

Orthogonal regression. In the proof of lemma 16.2 above, it was shown that the line $\mathbf{l}$ that minimizes the sum of squared distances to the set of all points $\mathbf{x}_i = (x_i, y_i, 1)^\mathsf{T}$ is obtained as follows.

(i) Define matrices $\mathtt{E} = \sum_i \mathbf{x}_i \mathbf{x}_i^\mathsf{T}$ and $\mathtt{J} = \mathrm{diag}(1, 1, 0)$.
(ii) Let $\boldsymbol{\xi}_0$ be the minimum root of the equation $\det(\mathtt{E} - \xi \mathtt{J}) = 0$.
(iii) The required line $\mathbf{l}$ is the right null-vector of the matrix $\mathtt{E} - \xi_0 \mathtt{J}$.

This gives a least-squares best fit of a line to a set of points. This process is known as *orthogonal regression* and it extends in an obvious way to higher-dimensional fitting of a hyperplane to a set of points in a way that minimizes the sum of squared distances to the points.

16.8 Closure

16.8.1 The literature

A linear method for computing the trifocal tensor was first given in [Hartley-97a], where further experimental results of estimation using both point and line correspondences on real data are reported. An iterative algebraic method for estimating a consistent tensor was given in [Hartley-98d].

Torr and Zisserman [Torr-97] developed an automatic algorithm for estimating a consistent tensor $\mathcal{T}$ from three images. This paper also compared several parametrizations of the iterative minimization. Several methods of representing and imposing the constraints on the tensor are given by Faugeras and Papadopoulo [Faugeras-97].

[Oskarsson-02] gives minimal solutions for reconstruction for the two cases of "four points and three lines in three views", and "two points and six lines in three views".

16.8.2 Notes and exercises

(i) Consider the problem of estimating the 3-space point $\mathbf{X}$ which minimizes reprojection error from measured image points $\mathbf{x}, \mathbf{x}', \mathbf{x}''$, given the trifocal tensor. This is the analogue of the triangulation problem of chapter 12. Show that for general motion the one parameter family parametrization of epipolar lines developed in chapter 12 does not extend from two views to three. However, in the case that the three camera centres are collinear the two-view parametrization can be extended to three and a minimum determined by solving a polynomial in one variable. What is the degree of this polynomial?

(ii) An affine trifocal tensor may be computed from a minimal configuration of 4 points in general position. The computation is similar to that of algorithm 14.2-(*p*352), and the resulting tensor satisfies the internal constraints for an affine trifocal tensor. How many constraints are there in the affine case?

If more than 4 point correspondences are used in the estimation then a geometrically valid tensor is estimated using the factorization algorithm of section 18.2(*p*436).

(iii) The transformation rule for tensors is $\mathcal{T}_i^{jk} = \mathtt{A}_i^r (\mathtt{B}^{-1})_s^j (\mathtt{C}^{-1})_t^k \hat{\mathcal{T}}_r^{st}$. This may be computed easily as

```
Binv = B.inverse();
Cinv = C.inverse();

for (i=1; i<=3; i++) for (j=1; j<=3; j++) for (k=1; k<=3; k++)
{
  T[i][j][k] = 0.0;

  for (r=1; r<=3; r++) for (s=1; s<=3; s++) for (t=1; t<=3; t++)
    T[i][j][k] +=
              A[r][i] * Binv[j][s] * Cinv[k][t] * T_hat[r][s][t];
}
```

How many multiplications and loop iterations does this involve? Find a better way of computing this transformation.

(iv) In the computation of the trifocal tensor using plane plus parallax (section 16.7.1), show that if ρ is the projective depth of one of the points off the plane (i.e. $\mathbf{x}' = \mathtt{H}_{12}\mathbf{x} + \rho\mathbf{e}'$ see (13.9–*p*337)), then the scalar λ in (16.5) may be computed from the equation $\mathbf{x}'' = \mathtt{H}_{13}\mathbf{x} + \rho\lambda\mathbf{e}''$.