PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Minimizing algebraic error

The Royal Society

Phil. Trans. R. Soc. Lond. A 1998 **356**, 1175-1192
doi: 10.1098/rsta.1998.0216

The Royal Society

Minimizing algebraic error

By Richard I. Hartley

GE Corporate Research and Development, 1 Research Circle, Niskayuna, NY 12309, USA

This paper gives a widely applicable technique for solving many of the parameter estimation problems encountered in geometric computer vision. A commonly used approach in such parameter minimization is to minimize an algebraic error function. It has generally been thought that minimizing a more meaningful geometric error function gives preferable results. It is claimed in this paper, however, that minimizing algebraic error will usually give excellent results, and in fact the main problem with most algorithms minimizing algebraic distance is that they do not take account of mathematical constraints that should be imposed on the quantity being estimated. This paper gives an efficient method of minimizing algebraic distance while taking account of the constraints. This provides new algorithms for the problems of resectioning a pinhole camera, computing the fundamental matrix, and computing the trifocal tensor.

Keywords: calibration and pose estimation; stereo and motion; image sequence analysis; fundamental matrix; trifocal tensor

1. Introduction

For many problems related to camera calibration and scene reconstruction, linear algorithms are known for solving for the entity required. In the sort of problem that will be addressed in this paper, a set of data (such as point correspondences) is used to construct a set of linear equations, and the solution of these equations, usually in the least-squares sense, provides an estimate of the entity being computed. As examples of such problems we have:

1. The direct linear transformation (DLT) algorithm for computing a camera matrix given a set of points in space, and corresponding points in the image. Provided at least six correspondences are given (more precisely $5\frac{1}{2}$ correspondences), one can solve for the camera matrix.
2. Computation of the fundamental matrix. From 8-point correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}'_i$ between two images one can construct the fundamental matrix using equations ${\boldsymbol{u}'_i}^{\mathrm{T}} F \boldsymbol{u}_i = 0$.
3. Computation of the trifocal tensor given a set of feature correspondences across three views.
4. Computation of the quadrifocal tensor given a set of feature correspondences across four views. This example will not be considered in this paper, however.

These linear algorithms have been found to give poor quality results on occasions and much research has been expended in seeking more reliable, but complex methods.

A previous paper (Hartley 1995*a*) showed that normalization of the data in a systematic manner will improve the results immeasurably, and such normalization must be routinely carried out. Nevertheless, a common criticism of such linear algorithms is that they 'do not minimize the right thing', and we really should be minimizing 'geometric error', which is related to the actual means of error occurrence. Although this criticism is correct, it is the thesis of this paper that it really does not matter very much whether we minimize algebraic or geometric error. What does make a significant difference is whether one does or does not enforce the constraints imposed by the geometry of the situation. Usually, one may be content with minimizing algebraic error, as long as one enforces the constraints. This leads to simpler and more efficient algorithms than are possible when minimizing geometric error.

In these four examples, and many others, the linear algorithm will lead to a solution that does not satisfy certain constraints that the estimated quantity must satisfy. In the cases considered here, the constraints are the following.

1. The *skew* parameter of a camera matrix estimated using the DLT method will not generally be zero. This constraint, meaning the pixels are rectangular, should be enforced in cases where it is known to hold.

2. The fundamental matrix must satisfy a constraint $\det F = 0$.

3. The trifocal tensor must satisfy eight nonlinear constraints. The form of these constraints is not easily determined, but it is essential to constrain the tensor to correspond to a valid set of camera matrices.

4. The quadrifocal tensor must satisfy 51 constraints, the nature of which is not well understood.

These constraints are not in general linear constraints, and in general, it will be necessary to resort to iterative techniques to enforce them. Since iterative techniques are slow and potentially unstable, it is important to use them sparingly. Further, the smaller the dimension of the minimization problem, the faster and generally more stable the solution will be. In this paper an iterative algorithm is used to solve the problems posed above. In each case the algorithms are based on a common technique of data reduction, whereby the input data are condensed into a *reduced measurement matrix*. The size of the iteration problem is then independent of the size of the input set. In the case of estimation of the fundamental matrix, only three homogeneous parameters are used to parametrize the minimization problem, whereas for the trifocal tensor, just six parameters are used.

The problem of camera calibration solved using the DLT algorithm will be treated first. It will be used to illustrate the techniques that apply to the other problems.

2. Computing the camera matrix

(*a*) *The DLT algorithm*

We consider a set of point correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{u}_i$ between three-dimensional (3D) points $\boldsymbol{x}_i$ and image points $\boldsymbol{u}_i$, expressed in homogeneous coordinates. Our problem is to compute a 3×4 matrix P such that $P\boldsymbol{x}_i = \boldsymbol{u}_i$ for each i. We begin with a simple linear algorithm for determining P. Note that the equation $P\boldsymbol{x}_i = \boldsymbol{u}_i$ involves

homogeneous vectors; thus $\boldsymbol{u}_i$ and $P\boldsymbol{x}_i$ may differ by a non-zero scale factor. One may, however, write the equation in terms of the vector cross product as $\boldsymbol{u}_i \times P\boldsymbol{x}_i = 0$.

If the jth row of the matrix P is denoted by $\boldsymbol{p}^{j\mathrm{T}}$, then we may write $P\boldsymbol{x}_i = (\boldsymbol{p}^{1\mathrm{T}}\boldsymbol{x}_i, \boldsymbol{p}^{2\mathrm{T}}\boldsymbol{x}_i, \boldsymbol{p}^{3\mathrm{T}}\boldsymbol{x}_i)^{\mathrm{T}}$. Writing $\boldsymbol{u}_i = (u_i, v_i, w_i)^{\mathrm{T}}$, the cross product may then be given explicitly as

$$\boldsymbol{u}_i \times P\boldsymbol{x}_i = \begin{pmatrix} v_i\boldsymbol{p}^{3\mathrm{T}}\boldsymbol{x}_i - w_i\boldsymbol{p}^{2\mathrm{T}}\boldsymbol{x}_i \\ w_i\boldsymbol{p}^{1\mathrm{T}}\boldsymbol{x}_i - u_i\boldsymbol{p}^{3\mathrm{T}}\boldsymbol{x}_i \\ u_i\boldsymbol{p}^{2\mathrm{T}}\boldsymbol{x}_i - v_i\boldsymbol{p}^{1\mathrm{T}}\boldsymbol{x}_i \end{pmatrix}.$$

Since $\boldsymbol{p}^{j\mathrm{T}}\boldsymbol{x}_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{p}^j$ for $j = 1, \ldots, 3$, this gives a set of three equations, in the entries of P, which may be written in the form

$$\begin{bmatrix} 0 & -w_i\boldsymbol{x}_i^{\mathrm{T}} & v_i\boldsymbol{x}_i^{\mathrm{T}} \\ w_i\boldsymbol{x}_i^{\mathrm{T}} & 0 & -u_i\boldsymbol{x}_i^{\mathrm{T}} \\ -v_i\boldsymbol{x}_i^{\mathrm{T}} & u_i\boldsymbol{x}_i^{\mathrm{T}} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^1 \\ \boldsymbol{p}^2 \\ \boldsymbol{p}^3 \end{pmatrix} = \mathbf{0}. \tag{2.1}$$

Note that $(\boldsymbol{p}^1, \boldsymbol{p}^2, \boldsymbol{p}^3)^{\mathrm{T}}$ which appears in (2.1) is a 12-vector made up of the entries of the matrix P. Although there are three equations, only two of them are linearly independent. Thus each point correspondence gives two equations in the entries of P. One may choose to omit the third equation, or else include all three equations, which may sometimes give a better conditioned set of equations. In future, we will assume that only the first two equations are used, namely

$$\begin{bmatrix} 0 & -w_i\boldsymbol{x}_i^{\mathrm{T}} & v_i\boldsymbol{x}_i^{\mathrm{T}} \\ w_i\boldsymbol{x}_i^{\mathrm{T}} & 0 & -u_i\boldsymbol{x}_i^{\mathrm{T}} \end{bmatrix} \begin{pmatrix} \boldsymbol{p}^1 \\ \boldsymbol{p}^2 \\ \boldsymbol{p}^3 \end{pmatrix} = \mathbf{0}. \tag{2.2}$$

The equations (2.2) may be denoted by $M_i\boldsymbol{p} = \mathbf{0}$, where the vector $\boldsymbol{p}$ is a 12-vector, corresponding to the 12 entries of P. The set of all equations derived from several point correspondences may be written $M\boldsymbol{p} = \mathbf{0}$, where M is the matrix of equation coefficients. This matrix M will be called the *measurement matrix*. The obvious solution $\boldsymbol{p} = \mathbf{0}$ is of no interest to us, so we seek a non-zero solution $\boldsymbol{p}$.

(*b*) *Scaling*

One of the most important things to do in implementing an algorithm of this sort is to prenormalize the data. This type of data normalization was discussed in Hartley (1995*a*). Without this normalization, all these algorithms are guaranteed to perform extremely poorly.

Data normalization is designed to improve the conditioning of the measurement matrix M. The appropriate scaling is to translate all data points so that their centroid is at the origin. Then the data should be scaled so that the average distance of any data point from the origin is equal to $\sqrt{2}$ for image points and $\sqrt{3}$ for 3D points. The algorithms are then carried out with the normalized data, and final transformations are applied to the result to compensate for the normalizing transforms.

(*c*) *Algebraic error*

In the presence of noise, one cannot expect to obtain an exact solution to an overconstrained set of equations of the form $M\boldsymbol{p} = \mathbf{0}$ such as those that arise in the DLT method.

The DLT algorithm instead finds the unit-norm vector $\boldsymbol{p}$ that minimizes $\|M\boldsymbol{p}\|$. The vector $\boldsymbol{e} = M\boldsymbol{p}$ is the error vector and it is this error vector that is minimized. The solution is the unit singular vector corresponding to the smallest singular value of M.

Define a vector $(\hat{u}_i, \hat{v}_i, \hat{w}_i)^{\mathrm{T}} = \hat{\boldsymbol{u}}_i = P\boldsymbol{x}$. Using this notation, we may write

$$M_i\boldsymbol{p} = \boldsymbol{e}_i = \begin{pmatrix} v_i\hat{w}_i - w_i\hat{v}_i \\ w_i\hat{u}_i - u_i\hat{w}_i \end{pmatrix} = \mathbf{0}. \tag{2.3}$$

This vector is the *algebraic error vector* associated with the point correspondence $\boldsymbol{u}_i \leftrightarrow \boldsymbol{x}_i$ and the camera mapping P. Thus

$$d_{\mathrm{alg}}(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 = (v_i\hat{w}_i - w_i\hat{v}_i)^2 + (w_i\hat{u}_i - u_i\hat{w}_i)^2. \tag{2.4}$$

Given several point correspondences, the quantity $\boldsymbol{e} = M\boldsymbol{p}$ is the algebraic error vector for the complete set, and one sees that

$$\sum_i d_{\mathrm{alg}}(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 = \|M\boldsymbol{p}\|^2 = \|\boldsymbol{e}\|^2. \tag{2.5}$$

The main lesson that we want to keep from this discussion is:

Proposition 2.1. *Given any set of 3D to image correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{x}_i$, let M be the measurement matrix as in (2.2). For any camera matrix P the vector $M\boldsymbol{p}$ is the algebraic error vector, where $\boldsymbol{p}$ is the vector of entries of P.*

(*d*) *Geometric distance*

Under the assumption that measurement error is confined to image measurements, and an assumption of a Gaussian error model for the measurement of two-dimensional (2D) image coordinates, the optimal estimate for the camera matrix P is the one that minimizes the error function

$$\sum_i d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2, \tag{2.6}$$

where $d(\cdot,\cdot)$ represents Euclidean distance in the image. The quantity $d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)$ is known as the *geometric distance* between $\boldsymbol{u}_i$ and $\hat{\boldsymbol{u}}_i$. Thus the error to be minimized is the sum of squares of geometric distances between measured and projected points.

For points $\boldsymbol{u}_i = (u_i, v_i, w_i)^{\mathrm{T}}$ and $\hat{\boldsymbol{u}}_i = (\hat{u}_i, \hat{v}_i, \hat{w}_i)^{\mathrm{T}}$, the geometric distance is

$$\begin{aligned} d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i) &= ((u_i/w_i - \hat{u}_i/\hat{w}_i)^2 + (v_i/w_i - \hat{u}_i/\hat{w}_i)^2)^{1/2} \\ &= d_{\mathrm{alg}}(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)/w_i\hat{w}_i. \end{aligned} \tag{2.7}$$

Thus geometric distance is related to, but not quite the same as algebraic distance. Nevertheless, it will turn out that minimizing algebraic distance gives very good results in general.

It is well known (see, for instance, Kanatani 1996) that in a more general context, the expected lower bound on the root mean squared residual error is equal to

$$E_{\mathrm{opt}} = \sigma(1 - d/N)^{1/2}, \tag{2.8}$$

where σ is the standard deviation of the input noise, N is the number of measurements (in this case 2 × number of points), and d is the number of degrees of freedom of the object being estimated, in this case the camera matrix P. Please note that (2.8) represents the error in *each* of the two image coordinates. This represents the performance of an optimal estimation technique, and we cannot do better.

(*e*) *The reduced measurement matrix*

Let $\boldsymbol{u}_i \leftrightarrow \boldsymbol{x}_i$ be a set of correspondences, and let M be the corresponding measurement matrix. Let P be any camera matrix, and let $\boldsymbol{p}$ be the vector containing its entries. The algebraic error vector corresponding to P is $M\boldsymbol{p}$, and its norm satisfies $\|M\boldsymbol{p}\|^2 = \boldsymbol{p}^{\mathrm{T}} M^{\mathrm{T}} M\boldsymbol{p}$.

In general, the matrix M may have a very large number of rows. It is possible to replace M by a square matrix $\hat{M}$ such that $\|M\boldsymbol{p}\| = \|\hat{M}\boldsymbol{p}\|$ for any vector $\boldsymbol{p}$. Such a matrix $\hat{M}$ is called a *reduced measurement matrix*. One way to do this is using the singular value decomposition (SVD). Let $M = UDV^{\mathrm{T}}$ be the SVD of M, and define $\hat{M} = DV^{\mathrm{T}}$. Then

$$M^{\mathrm{T}}M = (VDU^{\mathrm{T}})(UDV^{\mathrm{T}}) = (VD)(DV^{\mathrm{T}}) = \hat{M}^{\mathrm{T}}\hat{M},$$

as required. Another way of obtaining $\hat{M}$ is to use the QR decomposition $M = Q\hat{M}$, where Q has orthogonal columns and $\hat{M}$ is upper-triangular and square. This shows the following result.

Theorem 2.2. *Let $\boldsymbol{u}_i \leftrightarrow \boldsymbol{x}_i$ be a set of n world-to-image correspondences. Let M be the measurement matrix derived from the point correspondences. Let $\hat{M}$ be a reduced measurement matrix. Then, for any 3D to 2D projective transform P and corresponding 3-vector $\boldsymbol{p}$, one has*

$$\sum_i d_{\mathrm{alg}}(\boldsymbol{u}_i, P\boldsymbol{x}_i)^2 = \|\hat{M}\boldsymbol{p}\|^2.$$

In this way, all the information we need to keep about the set of matched points $\boldsymbol{u}_i \leftrightarrow \boldsymbol{x}_i$ is contained in the single 12×12 matrix $\hat{M}$. If we wish to minimize algebraic error as P varies over some restricted set of transforms, then this is equivalent to minimizing the norm of the 12-vector $\|\hat{M}\boldsymbol{p}\|$.

(*f*) *Restricted camera mappings*

The camera mapping expressed by a general 3D projective transformation is in some respects too general. A non-singular 3×4 matrix P with centre at a finite point may be decomposed as $P = K[R \mid -Rt]$ where R is a 3×3 rotation matrix and

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ & \alpha_v & v_0 \\ & & 1 \end{bmatrix}. \tag{2.9}$$

The non-zero entries of K are geometrically meaningful quantities, the internal calibration parameters of P. A common assumption is that $s = 0$, while for a true pinhole camera, $\alpha_u = \alpha_v$.

Given a set of world-to-image correspondences, one may wish to find a matrix P that minimizes algebraic error, subject to a set of constraints on P. Usually, this will require an iterative solution. For instance, suppose we wish to enforce the constraints $s = 0$ and $\alpha_u = \alpha_v$. One can parametrize the camera matrix using the remaining nine parameters (u_0, v_0, α, plus six parameters representing the orientation R and location $\boldsymbol{t}$ of the camera). Let this set of parameters be denoted collectively by $\boldsymbol{q}$. Then, one has a map $\boldsymbol{p} = g(\boldsymbol{q})$, where $\boldsymbol{p}$ is as before the vector of entries of the matrix P. According to theorem 2.2, minimizing algebraic error over all point matches is

equivalent to minimizing $\|Mg(\boldsymbol{q})\|$. Note that the mapping $\boldsymbol{q} \mapsto Mg(\boldsymbol{q})$ is a mapping from R^9 to R^{12}. This is a simple parameter-minimization problem that may be solved using the Levenberg–Marquardt (LM) method. The important point to note is the following:

> Given a set of n world-to-image correspondences, $\boldsymbol{x}_i \leftrightarrow \boldsymbol{u}_i$, the problem of finding a constrained camera matrix P that minimizes the sum of algebraic distances $\sum_i d_{\mathrm{alg}}(\boldsymbol{u}_i, P\boldsymbol{x}_i)^2$ reduces to the minimization of a function $R^9 \to R^{12}$, independent of the number n of correspondences.

If this problem is solved using the LM method, then an initial estimate of the parameters may be obtained by decomposing a camera matrix P found using the DLT algorithm. A central step in the LM method is the computation of the derivative matrix (Jacobian matrix) of the function being minimized, in this case $Mg(\boldsymbol{q})$. Note that $\partial Mg/\partial \boldsymbol{q} = M\partial g/\partial \boldsymbol{q}$. Thus, computation of the Jacobian reduces to computation of the Jacobian matrix of g, and subsequent multiplication by M.

Minimization of $\|Mg(\boldsymbol{q})\|$ takes place over all values of the parameters $\boldsymbol{q}$. Note, however, that if $P = K[R \mid -R\boldsymbol{t}]$ with K as in (2.9), then P satisfies the condition $p_{31}^2 + p_{32}^2 + p_3^2 = 1$, since these entries are the same as the last row of the rotation matrix R. Thus, minimizing $Mg(\boldsymbol{q})$ will lead to a matrix P satisfying the constraints $s = 0$ and $\alpha_u = \alpha_v$ and scaled such that $p_{31}^2 + p_{32}^2 + p_3^2 = 1$, and which in addition minimizes the algebraic error for all point correspondences.

(g) *Experimental evaluation*

Experiments were carried out with synthetic data to evaluate the performance of this algorithm. The data were created to simulate a standard 35 mm camera with a 35 mm focal length lens. A set of points were synthesized inside a sphere of radius 1 m, and the camera was located at a distance of about 2.5 m from the centre of the sphere. The image is sampled so that the magnification factors are $\alpha_u = \alpha_v = 1000.0$, the same in each direction. This corresponds to a pixel size of 35 μm for a 35 mm camera.

Experiments were carried out to find the camera matrix with four different assumptions on known camera parameters.

1. Zero skew: $s = 0$. The number of remaining degrees of freedom d for the camera matrix is equal to 10.
2. The pixels are square: $s = 0$ and $\alpha_u = \alpha_v$. This corresponds to the situation for a true pinhole camera where image coordinates are measured in a Euclidean coordinate frame. In this case, $d = 9$.
3. In addition to the above assumptions, the principal point (u_0, v_0) is assumed to be known. There remain $d = 7$ degrees of freedom.
4. The complete internal calibration matrix K in (2.9) is assumed to be known. However, the pose of the camera is unknown. Thus $d = 6$.

To evaluate the performance of the algorithm, the result was compared with the optimal estimate with different degrees of noise. Thus, Gaussian noise with a given variance was added to the image coordinates of each point, and the camera matrix

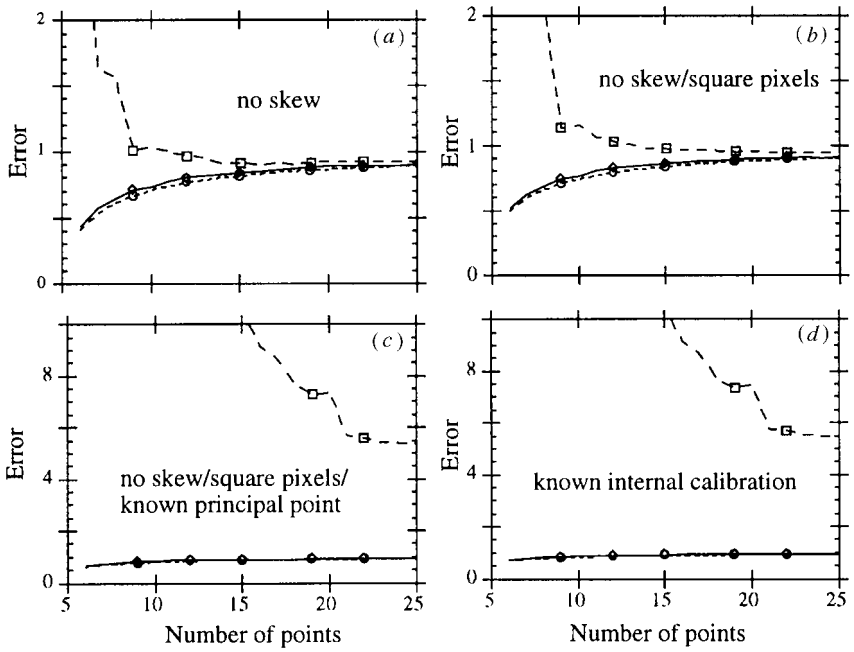

Figure 1. The residual error was RMS-averaged over 100 runs for each $n = 6, \ldots, 25$, where n is the number of points used to estimate the camera matrix. Four different levels of knowledge of the internal camera matrix were tried, corresponding to the four different graphs. In each of the graphs, the solid line represents the result of our iterative DLT algorithm, and the almost identical dotted line is the optimal estimate. In all four graphs, these two lines are barely distinguishable. For comparison, the results of a further method are also plotted. In this method, the camera matrix P is computed using the linear DLT algorithm, the complete calibration matrix K in (2.9) is then computed using the QR decomposition, and the known internal parameters are subsequently set to their known values. This method performs very poorly for small numbers of points, lying well off the graph, and is markedly inferior to the optimal estimation method, even for larger numbers of points.

was estimated. The residual error was then computed, that is the difference between the measured and projected pixel. Since the residual error appeared to grow proportionally to injected noise (at least for noise levels less than about 10 pixels), a value of $\sigma = 1$ pixel was used in the experiments. For each level of noise σ, the camera matrix was estimated 100 times with random noise. The residual error was RMS-averaged over all 100 runs and compared to the optimal value given by (2.8).

Results of the experiments are shown in figure 1. The results show that minimizing algebraic error gives an almost optimal estimate of the camera matrix. In fact, the residual error is scarcely distinguishable from the optimal value. This is true in each of the four calibration problem types tried.

3. Computation of the fundamental matrix

(*a*) *The 8-point algorithm*

We now turn to the computation of the fundamental matrix. It will turn out that very similar methods apply to its computation as were used in the DLT algorithm.

Given a set of correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}'_i$ between two images, the fundamental matrix is defined by the relation $\boldsymbol{u}_i'^{\mathrm{T}} F \boldsymbol{u}_i = 0$ for all i. In the presence of noise, this relation will not hold precisely, and so one seeks a least-squares solution. Note that the equation $\boldsymbol{u}_i'^{\mathrm{T}} F \boldsymbol{u}_i = 0$ is linear in the entries of F. From eight or more point matches, one may solve for the entries of F by finding the least-squares solution to a set of linear equations (Hartley 1995*a*). Let the set of equations be denoted by $M\boldsymbol{f} = 0$. The vector $M\boldsymbol{f}$ has components equal to $\boldsymbol{u}_i'^{\mathrm{T}} F \boldsymbol{u}_i$, and $\|M\boldsymbol{f}\|^2 = \sum_i (\boldsymbol{u}_i'^{\mathrm{T}} F \boldsymbol{u}_i)^2$. Thus, in this case, as before with the DLT algorithm, $M\boldsymbol{f}$ represents the algebraic error vector. Matrix F is found by minimizing $\|M\boldsymbol{f}\|$ subject to $\|\boldsymbol{f}\| = 1$.

The fundamental matrix F must, however, satisfy a constraint $\det F = 0$, and this constraint will not generally be satisfied by the matrix F found by this linear algorithm. One would therefore like to minimize the algebraic error $\|M\hat{\boldsymbol{f}}\|$ over all vectors $\hat{\boldsymbol{f}}$ corresponding to singular matrices $\hat{F}$.

In Hartley (1995*a*), the matrix $\hat{F}$ was taken to be the closest singular matrix to F under Frobenius norm, where F is the linear solution. This is not an especially good way of proceeding, since it weights errors in each of the entries of F equally. A preferable method is to proceed as with the DLT. One parametrizes the matrix $\hat{F}$ by a set of parameters $\boldsymbol{q}$ in such a way as to ensure it is singular. Then, letting $\hat{f} = g(\boldsymbol{q})$, one uses an iterative algorithm to minimize $\|Mg(\boldsymbol{q})\|$. This is the general scheme which will be followed, but there are details to be filled out, and a new twist will arise, which allows a parametrization with only three parameters.

(*b*) *Algebraic minimization*

Consider the fundamental matrix F, which can be written as a product $F = Q[\boldsymbol{e}]_\times$ where Q is a non-singular matrix, and $\boldsymbol{e}$ is the epipole in the first image. The symbol $[\boldsymbol{e}]_\times$ represents the skew-symmetric 3×3 matrix such that $[\boldsymbol{e}]_\times \boldsymbol{v} = \boldsymbol{e} \times \boldsymbol{v}$ for any vector $\boldsymbol{v}$.

Suppose we wish to compute the fundamental matrix F of the form $F = Q[\boldsymbol{e}]_\times$ that minimizes the algebraic error $\|M\boldsymbol{f}\|$ subject to the condition $\|\boldsymbol{f}\| = 1$. The vector $\boldsymbol{f}$ is the 9-vector containing the entries of F. It has been seen that the 8-point algorithm finds such an $\boldsymbol{f}$, without the condition that $F = Q[\boldsymbol{e}]_\times$. We now wish to enforce that condition.

Let us assume for now that the epipole $\boldsymbol{e}$ is known. Later we will let $\boldsymbol{e}$ vary, but for now it is fixed. The equation $F = Q[\boldsymbol{e}]_\times$ can be written in terms of the vectors $\boldsymbol{f}$ and $\boldsymbol{q}$, comprising the entries of F and Q as an equation $\boldsymbol{f} = E\boldsymbol{q}$, where E is a 9×9 matrix. Supposing that $\boldsymbol{f}$ and $\boldsymbol{q}$ contain the entries of the corresponding matrices in row-major order, then it can be verified that E has the form

$$E = \begin{bmatrix} [\boldsymbol{e}]_\times & & \\ & [\boldsymbol{e}]_\times & \\ & & [\boldsymbol{e}]_\times \end{bmatrix}. \tag{3.1}$$

Now, our minimization problem is as follows: minimize $\|ME\boldsymbol{q}\|$ subject to the condition $\|E\boldsymbol{q}\| = 1$.† This problem is solved as follows. Let the singular value decomposition of E be $E = UDV^{\mathrm{T}}$. It is easily seen that the matrix E has rank 6, since each of the diagonal blocks has rank 2. It follows that D has six non-zero

† It does not do to minimize $\|ME\boldsymbol{q}\|$ subject to the condition $\|\boldsymbol{q}\| = 1$, since a solution to this occurs when $\boldsymbol{q}$ is a unit vector in the right nullspace of E. In this case, $E\boldsymbol{q} = 0$, and hence $\|ME\boldsymbol{q}\| = 0$.

diagonal entries. Let U' be the 9×6 matrix consisting of the first six columns of U, and let V' consist of the first 6 columns of V and let D' be the top-left 6×6 minor of D, containing the non-zero diagonal entries. The minimization problem then becomes the following: minimize $\|MU'D'V'^{\mathrm{T}}\boldsymbol{q}\|$ subject to $\|U'D'V'^{\mathrm{T}}\boldsymbol{q}\| = 1$. This last condition is equivalent to $\|D'V'^{\mathrm{T}}\boldsymbol{q}\| = 1$, since U' has orthogonal columns. Now, writing $\boldsymbol{q}' = D'V'^{\mathrm{T}}\boldsymbol{q}$, the problem becomes: minimize $\|MU'\boldsymbol{q}'\|$ subject to $\|\boldsymbol{q}'\| = 1$, which is our standard minimization problem. The solution $\boldsymbol{q}'$ is the singular vector corresponding to the smallest singular value of MU'. Subsequently, we can compute $\boldsymbol{f} = E\boldsymbol{q} = U'D'V'^{\mathrm{T}}\boldsymbol{q} = U'\boldsymbol{q}'$, and the algebraic error is $M\boldsymbol{f} = MU'\boldsymbol{q}'$. In this way, we compute $\boldsymbol{q}'$. If $\boldsymbol{q}$ is required as well, then it is easily obtained. Because E is not a full rank matrix, there is not a unique solution for $\boldsymbol{q}$ from the equation $D'V'^{\mathrm{T}}\boldsymbol{q} = \boldsymbol{q}'$. However, one solution for $\boldsymbol{q}$ is given by $\boldsymbol{q} = V'D'^{-1}\boldsymbol{q}$. To see this, one verifies $D'V'^{\mathrm{T}}\boldsymbol{q} = D'V'^{\mathrm{T}}V'D'^{-1}\boldsymbol{q} = D'D'^{-1}\boldsymbol{q} = \boldsymbol{q}$ as required.

The complete algorithm is:

Algorithm 3.1. *Given the epipole* $\boldsymbol{e}$*, find the fundamental matrix* F *of the form* $F = Q[\boldsymbol{e}]_\times$ *that minimizes the algebraic error* $\|M\boldsymbol{f}\|$ *subject to* $\|\boldsymbol{f}\| = 1$.

Solution.

1. Compute the SVD $E = UDV^{\mathrm{T}}$, where E is given in (3.1), and the non-zero values of D appear first down the diagonal.
2. Let U' be the matrix comprising the first six columns of U, let V' consist of the first six columns of $\boldsymbol{V}$ and D' consist of the six first rows and columns of D.
3. Find the unit vector $\boldsymbol{q}'$ that minimizes $\|MU'\boldsymbol{q}'\|$.
4. The required matrix F corresponds to the vector $\boldsymbol{f} = U'\boldsymbol{q}'$, and the minimum algebraic error is $M\boldsymbol{f}$.
5. A factorization of $F = Q[\boldsymbol{e}]_\times$ is obtained by computing the vector $\boldsymbol{q} = V'D'^{-1}\boldsymbol{q}'$ corresponding to Q.

(*c*) *Iterative estimation*

The algorithm of the last section gives a way of computing an algebraic error vector $M\boldsymbol{f}$ given a value for the epipole $\boldsymbol{e}$. This mapping $\boldsymbol{e} \mapsto M\boldsymbol{f}$ is a map from R^3 to R^9. Note that the value of $M\boldsymbol{f}$ is unaffected by scaling $\boldsymbol{e}$. Starting from an estimated value of $\boldsymbol{e}$ derived as the generator of the right nullspace of an initial estimate of F, one may iterate to find the final F that minimizes algebraic error. The initial estimate of F may be obtained from the 8-point algorithm, or any other simple algorithm.

Note the advantage of this method of computing F is that the iterative part of the algorithm consists of a very small parameter minimization problem, involving the estimation of only three parameters. Despite this, the algorithm finds the fundamental matrix that minimizes the algebraic error for all matched points. The matched points themselves do not come into the final iterative estimation.

Simplifying the computation. Because of the simple form of the matrix E, it is easy to compute its SVD without having to resort to a full SVD algorithm. This may be

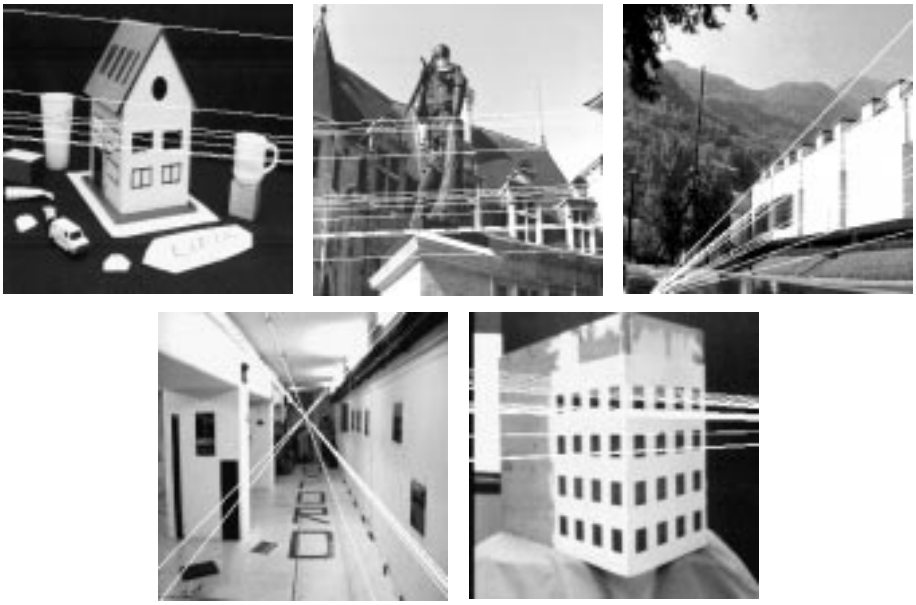

Figure 2. The images used in the experiments.

important in the iterative algorithm to achieve maximum speed, since this SVD is computed repeatedly during the minimization. As seen in (3.1), the matrix E has a diagonal block-structure consisting of three blocks $[\boldsymbol{e}]_\times$. The SVD consequently has a corresponding block-structure. Specifically, if $[\boldsymbol{e}]_\times = \hat{U}\hat{D}\hat{V}'$, then the SVD of $E = \mathrm{diag}([\boldsymbol{e}]_\times, [\boldsymbol{e}]_\times, [\boldsymbol{e}]_\times)$ is $E = UDV^{\mathrm{T}}$, where $U = \mathrm{diag}(\hat{U}, \hat{U}, \hat{U})$, and similarly for D and V.

The SVD of $[\boldsymbol{e}]_\times$ itself can be computed easily as follows. Suppose that $\hat{U}$ is an orthogonal matrix such that $\boldsymbol{e}\hat{U} = (0, 0, 1)$. Such a matrix $\hat{U}$ is a Householder transformation and is easily computed (Golub & Van Loan 1989). Then one sees that $[\boldsymbol{e}]_\times = \pm\hat{U}Z\hat{U}^{\mathrm{T}} = \pm\hat{U}\,\mathrm{diag}(1, 1, 0)\hat{Z}\hat{U}^{\mathrm{T}} = \pm\hat{U}\hat{D}\hat{V}^{\mathrm{T}}$, where

$$Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \qquad \hat{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This is easily verified by observing that both $[\boldsymbol{e}]_\times$ and $\pm\hat{U}Z\hat{U}^{\mathrm{T}}$ are skew-symmetric matrices with the same nullspace, generated by $\boldsymbol{e}$ in each case. We are interested in $\hat{U}'$ consisting of the first two columns of $\hat{U}$. Turning now to the SVD of $E = \mathrm{diag}([\boldsymbol{e}]_\times, [\boldsymbol{e}]_\times, [\boldsymbol{e}]_\times)$, we see that $U' = \mathrm{diag}(\hat{U}', \hat{U}', \hat{U}')$. If we partition the 9×9 matrix M into blocks $M = [M_1, M_2, M_3]$, where each M_i has 3 columns, then one computes that $MU' = [M_1\hat{U}', M_2\hat{U}', M_3\hat{U}']$. Thus, the computation of MU' required in algorithm 3.1 has two simple steps.

1. Compute the 3×3 Householder matrix $\hat{U}$ such that $\boldsymbol{e}^{\mathrm{T}}\hat{U} = (0, 0, 1)$, and let $\hat{U}'$ comprise its first two columns.

2. Set $MU' = [M_1\hat{U}', M_2\hat{U}', M_3\hat{U}']$.

(*d*) *Experimental evaluation of the algorithm*

A set of experiments were carried out similar to those in Hartley (1995*a*). One image from each pair of images used is shown in figure 2. These images contain a wide variation of measurement noise and placement of the epipoles. For each pair of images, a number n of matched points were chosen and the fundamental matrix was computed. The fundamental matrix computed was shown evaluated against the full set of all matched points, and the residual error was computed. This experiment was done 100 times for each value of n and each pair of images, and the average residual error was plotted against n. This gives an idea of how the different algorithms behave as the number of points is increased.

The results of these experiments are shown and explained in figure 3. They show that minimizing algebraic error gives essentially indistinguishable results from minimizing the geometric error, but both perform better than the linear normalized 8-point algorithm (Hartley 1995*a*). The optimal residual geometric error is computed from (2.8) to be in this case

$$E_{\mathrm{opt}} = \sigma \left(\frac{n-7}{4n} \right)^{1/2}.$$

4. Computation of the trifocal tensor

(*a*) *The linear solution*

The trifocal tensor (Hartley 1995*b*, 1997) relates the coordinates of points or lines seen in three views in a similar way to that in which the fundamental matrix relates points in two views.

The basic formula relates a point $\boldsymbol{u}$ in one image and a pair of lines λ' and λ'' in the other two images. Provided there is a point $\boldsymbol{x}$ in space that maps to $\boldsymbol{u}$ in the first image, and a point on the lines λ' and λ'' in the other two images, the following identity is satisfied:

$$u^i \lambda'_j \lambda''_k T_i^{jk} = 0. \tag{4.1}$$

Here we are using tensor notation, in which a repeated index appearing in covariant (lower) and contravariant (upper) positions implies summation over the range of indices (namely, $1, \ldots, 3$).

This equation may be used to generate equations given either point or line correspondences across three images. In the case of a line correspondence, $\lambda \leftrightarrow \lambda' \leftrightarrow \lambda''$ one selects two points $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ on the line λ, and for each of these points one obtains an equation of the form (4.1). In the case of a point correspondence $\boldsymbol{u} \leftrightarrow \boldsymbol{u}' \leftrightarrow \boldsymbol{u}''$ one selects any lines λ' and λ'' passing through $\boldsymbol{u}'$ and $\boldsymbol{u}''$, respectively. Then (4.1) provides one equation. Four equations are generated from a single 3-viewpoint correspondence by choosing two lines through each of $\boldsymbol{u}'$ and $\boldsymbol{u}''$, each pair of lines giving rise to a single equation.

The equations (4.1) give rise to a set of equations of the form $M\boldsymbol{t} = 0$ in the 27 entries of the trifocal tensor. From these equations, one may solve for the entries of the tensor. As before, for any tensor T_i^{jk} the value of $M\boldsymbol{t}$ is the algebraic error vector associated with the input data.

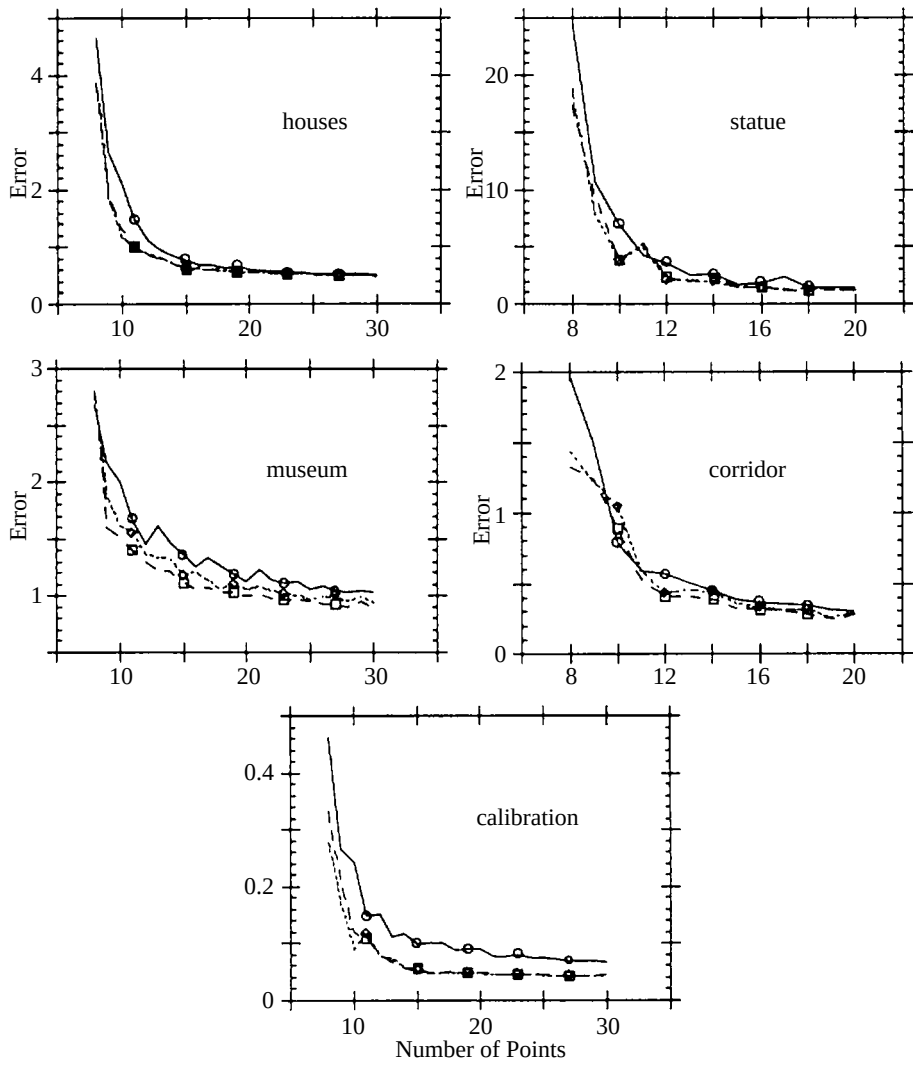

Figure 3. Results of the experimental evaluation of the algorithms. In each case, three methods of computing F were compared. In each graph, the top (solid) line shows the results of the normalized 8-point algorithm. Also shown are the results of minimizing geometric error and algebraic error, using the algorithm of this paper. In most cases, the result of minimizing algebraic error is almost indistinguishable from minimizing geometric error. Both are noticeably better than the non-iterative 8-point algorithm, although that algorithm gives reasonable results.

Consider the analogy with the 8-point algorithm for computing the fundamental matrix in the two-view case. The fundamental matrix has a constraint $\det F = 0$ that is not in general precisely satisfied by the solution found from the linear algorithm. In the case of the trifocal tensor, there are 27 entries in the tensor, but the camera geometry that it encodes has only 18 degrees of freedom. This means that the trifocal tensor must satisfy eight constraints, apart from scale ambiguity to make up the 27 degrees of freedom of a general $3 \times 3 \times 3$ tensor. The exact form of these constraints is not known precisely. Nevertheless, they must be enforced in order that the trifocal

tensor should be well behaved. It will now be shown how this can be done, while minimizing algebraic error.

(*b*) *Enforcing the constraints*

We denote the camera matrices P' and P'' by a^i_j and b^i_j, respectively, instead of by p'^i_j and p''^i_j. Thus, the three camera matrices P, P' and P'' may be written in the form $P = [I \mid 0]$, $P' = [a^i_j]$ and $P'' = [b^i_j]$.

In this notation, the formula for the entries of the trifocal tensor is (Hartley 1997)

$$T^{jk}_i = a^j_i b^k_4 - a^j_4 b^k_i. \tag{4.2}$$

Our task will be to compute a trifocal tensor T^{jk}_i of this form from a set of image correspondences. The tensor computed will minimize the algebraic error associated with the input data. The algorithm is quite similar to the one given for computation of the fundamental matrix. Just as with the fundamental matrix, the first step is the computation of the epipoles.

Retrieving the epipoles. Consider the task of retrieving the epipoles from the trifocal tensor. If the first camera has matrix $P = [I \mid \mathbf{0}]$, then the epipoles $\boldsymbol{e}_{21}$ and $\boldsymbol{e}_{31}$ are the last columns a^i_4 and b^i_4 of the two camera matrices $P' = [a^i_j]$ and $P'' = [b^i_j]$, respectively. These two epipoles may easily be computed from the tensor T^{jk}_i according to the following proposition (Hartley 1995*b*).

Proposition 4.1. *For each $i = 1, \ldots, 3$, the matrix $T^{\cdot\cdot}_i$ is singular. Furthermore, the generators of the three left nullspaces have a common perpendicular, the epipole $\boldsymbol{e}_{21}$. Similarly, epipole $\boldsymbol{e}_{31}$ is the common perpendicular of the right nullspaces of the three matrices $T^{\cdot\cdot}_i$.*

This proposition translates easily into an algorithm for computing the epipoles (Hartley 1995*b*, 1997). This algorithm may be applied to the tensor T^{jk}_i obtained from the linear algorithm to obtain a reasonable approximation for the epipoles.

Constrained minimization. From the form (4.2) of the trifocal tensor, it may be seen that once the epipoles $\boldsymbol{e}_{21} = a^j_4$ and $\boldsymbol{e}_{31} = b^k_4$ are known, the trifocal tensor may be expressed linearly in terms of the remaining entries of the matrices $[a^j_i]$ and $[b^k_i]$. We may write $\boldsymbol{t} = H\boldsymbol{a}$ where $\boldsymbol{a}$ is the vector of the remaining entries a^i_j and b^i_j, $\boldsymbol{t}$ is the vector of entries of the trifocal tensor, and H is the linear relationship expressed by (4.2). We wish to minimize the algebraic error $\|M\boldsymbol{t}\| = \|MH\boldsymbol{a}\|$ over all choices of $\boldsymbol{a}$ constrained such that $\|\boldsymbol{t}\| = \|H\boldsymbol{a}\| = 1$.

Writing $\hat{\boldsymbol{t}} = H\boldsymbol{a}$, where $\boldsymbol{a}$ is the solution vector, we see that $\hat{\boldsymbol{t}}$ minimizes algebraic error $\|M\hat{\boldsymbol{t}}\|$ subject to the condition that T^{jk}_i is of the correct form (4.2), for the given choice of epipoles.

(*c*) *Iterative solution*

The two epipoles used to compute a correct constrained tensor T^{jk}_i are computed using the estimate of T^{jk}_i obtained from the linear algorithm. Analogous to the case of the fundamental matrix, the mapping $(\boldsymbol{e}_{21}, \boldsymbol{e}_{31}) \mapsto M\hat{\boldsymbol{t}} = MH\boldsymbol{a}$ is a mapping $R^6 \to R^{27}$. An application of the LM 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 six parameters, the homogeneous coordinates of the epipoles, are involved in the iteration problem.

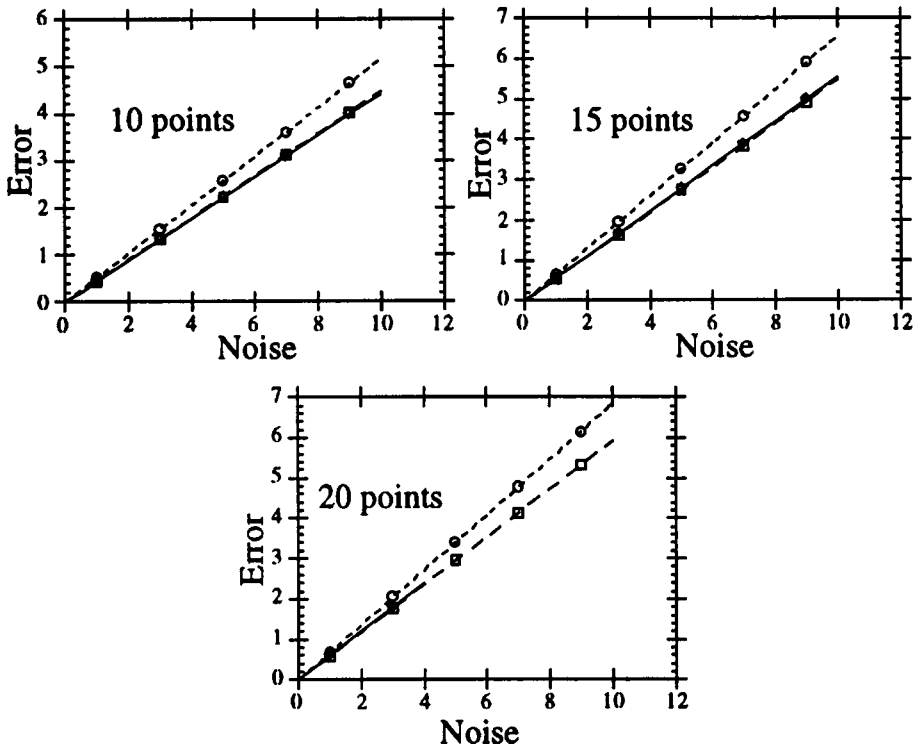

Figure 4. The residual error RMS-averaged over 100 runs is plotted against the number of points. Note that these plots are different from the plots for the DLT algorithm in that the horizontal axis represents the noise level, not the number of points as in the DLT case. The three plots are for 10, 15 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 iteration to minimize geometric error, using the algebraic minimization as a starting point (Hartley 1995*b*, 1997). Note that the residual errors are almost exactly proportional to added noise. We learn two things from this. 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. These results are notably better than results obtained in Hartley (1995*b*).

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

(*d*) *Experimental results*

Once more, the iterative algorithm for computing the trifocal tensor was tested with synthetic data. The configuration of the points and cameras was similar to that used for the DLT algorithm, but in this case there were three cameras aimed at the point set from random angles. Data sets of 10, 15 and 20 points were used to test the algorithm. Residual errors were compared with the optimal values of the residual errors, given by the formula (2.8). 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 (less 15 to account for projective ambiguity) and $3n$ represents the number of degrees of freedom of n points in space. Thus, in this case,

$$E_{\text{opt}} = \sigma \left(\frac{3n - 18}{6n} \right)^{1/2}.$$

The results are shown in figure 4.

5. Conclusion

Experimental evidence backs up the assertion that minimizing algebraic distance can usually give good results at a fraction of the computation cost associated with minimizing geometric distance. The great advantage of the method for minimizing algebraic error given in this paper is that even for problems that need an iterative solution the size of the iteration problem is very small. Consequently, the iteration is very rapid and there is reduced risk of falling into a local minimum, or otherwise failing to converge.

The method has been illustrated by applying it to three problems. For the computation of the fundamental matrix, iteration is over only three homogeneous parameters. For the trifocal tensor, iteration is over six parameters. This leads to more efficient methods than have been known.

The general technique is applicable to problems other than those treated here. It may be applied to the computation of the quadrifocal tensor and also to estimation of projective transformations between two- or three-dimensional point sets. In this latter problem, iteration is necessary if one restricts the class of available transformations to a subgroup of the projective group, such as planar homologies (used in Zisserman (1995)), or conjugates of rotations (Hartley 1994).

References

Golub, G. H. & Van Loan, C. F. 1989 *Matrix computations*, 2nd edn. Baltimore and London: The Johns Hopkins University Press.

Hartley, R. I. 1994 Self-calibration from multiple views with a rotating camera. In *Computer vision – ECCV '94*, vol. I (LNCS Series vol. 800), pp. 471–478. Springer.

Hartley, R. I. 1995*a* In defence of the 8-point algorithm. In *Proc. Int. Conf. on Computer Vision*, pp. 1064–1070. Los Alamitos, CA: IEEE Computer Society Press.

Hartley, R. I. 1995*b* A linear method for reconstruction from lines and points. In *Proc. Int. Conf. on Computer Vision*, pp. 882–887. Los Alamitos, CA: IEEE Computer Society Press.

Hartley, R. I. 1997 Lines and points in three views and the trifocal tensor. *Int. J. Computer Vision* **22**, 125–140.

Kanatani, K. 1996 *Statistical optimization for geometric computation: theory and practice*. Amsterdam: North Holland.

Zisserman, A., Forsyth, D. A., Mundy, J. L., Rothwell, C. A., Liu, J. & Pillow, N. 1995 3D object recognition using invariance. *AI Jl* **78** 239–288.

Discussion

P. H. S. Torr (*Department of Engineering Science, University of Oxford, UK*). Initially Dr Hartley stated that for conic fitting it was undesirable to minimize the

algebraic distance, but he then went on to say that this would be somehow good for the fitting of the fundamental matrix. I believe that this is a contradiction. The fundamental matrix is a quadratic surface in the four-dimensional space of the image coordinates: x, y in the first image, x, y in the second image, making a four-dimensional space. Thus if minimizing the algebraic distance produces poor results in conic fitting, it will certainly produce poor results for fitting the fundamental matrix.

To explain why such apparently good results are obtained, consider something that Takeo Kanade said earlier: that the affine camera was appropriate for many image pairs. In this case there is a linear relationship between the four image coordinates in the two images, and minimizing the algebraic distance for the centred and scaled coordinate system is the same as minimizing the geometric distance (as explained by Larry Shapiro in his book), which explains these good results. Furthermore, if there are significant perspective effects, such that the affine camera is not a good model, then minimizing the algebraic distance is a bad thing and mediocre results will be achieved. Has he any comments on this?

R. I. Hartley. The analogy between conic fitting and fundamental matrix estimation is an interesting one. It is not quite an exact analogy, since in the case of a conic the defining relationship expressed by the fundamental matrix is bilinear in the two sets of indices, whereas in the case of a conic section the equation is an arbitrary quadratic. Nevertheless, I am not sure that this makes a significant difference and I admit the apparent contradiction. However, I have not really tried these techniques for conic fitting, and have no personal knowledge of how well they will work. Perhaps this requires re-evaluation. For the fundamental matrix, they work well. I do not think that it is a matter of being close to an affine approximation. The algorithm was evaluated by using real images, as seen in figure 2 of the paper. Most of the images have a large depth ratio, and the affine approximation is not tenable. For the trifocal tensor estimation case, synthetic data were used for which the depth of the point set was as much as 40% of the distance of the centre of the points to the camera. For this case as well, the affine approximation is not valid, yet the algorithm gives good results.

Perhaps a key to the success of the algebraic minimization method is that the 'algebraic error' function should be related to geometric error. For instance, in the DLT resection case (§ 2 of my paper) it is shown that the algebraic and geometric error differ only by the factor ww'. Since w and w' are proportional to the depth of points from the camera centre, this means that algebraic error is geometric image error weighted by depth, and hence is closely related to error in position of the 3D points. Thus in this case, algebraic error is closely related to meaningful geometric error, which would explain the good results obtained. Possibly something similar happens in the case of the other estimation problems. Conversely, for some problems it must be possible to formulate the equations in such a way that the 'algebraic error' that arises has no close relationship to geometric error, in which case one would expect poor results.

A. Fitzgibbon (*Department of Engineering, University of Oxford, UK*). Almost an implementation detail, but often very important. How are the epipoles parametrized?

R. I. Hartley. I parametrize each epipole by its three homogeneous coordinates, rather than by using the minimum parametrization with just two coordinates. This

has the advantage that one need not worry about epipoles close to infinity, for which the third coordinate is close to zero and non-homogeneous coordinates are unstable. In particular it is possible during iteration for the epipole to pass smoothly through infinity and swap from one side of the image to the other, which would not be possible if non-homogeneous coordinates (two per epipole) are used.

A. Fitzgibbon. So the epipole is allowed to float.

R. I. Hartley. Yes, but I do try to keep each epipole close to having norm 1 by renormalizing it at the end of each iteration of the Levenberg–Marquardt algorithm. This probably has minimal effect, but keeps the coordinates from becoming too large.

There are a couple of other little tricks in the implementation that I've glossed over. An important one concerns the error vector. The error vector $M\boldsymbol{f}$ is only defined up to sign, so you have to make sure that you consistently select the correct sign so that the computed error vector varies as a continuous function of the parameters. This may be done by choosing the sign so that the error has positive inner product with the previously computed error vector. Otherwise, a small change of parameters may cause a large change of error as the error vector swaps sign, as an artifact of the SVD or other numeric procedures.

S. Carlsson (*KTH, Stockholm, Sweden*). When one tries to estimate conics, it is generally known that unless a large proportion of the conic is covered it will fail. Is there something similar here in the fundamental matrix? Is there some kind of maximum angular spread of the points?

R. I. Hartley. I think that the failure modes for fundamental matrix computation are best understood in terms of the critical sets for its calculation: there is ambiguity precisely when the set of points and the two camera matrices lie on a quadric surface in space. This includes the case of degenerate quadrics, such as a pair of intersecting planes. For configurations close to a critical configuration, the extraction of the fundamental matrix will be poorly conditioned.

P. H. S. Torr. Actually, the 'angular spread' of the points to which Stefan Carlsson refers, corresponds to the spread of points observed in the image together with the distance that they move between images, i.e. the amount of variation of the x, y, x' and y' coordinates. If the points are imaged over many different depths and over a wide baseline, then this will lead to large perspective effects. In the situation without these effects, all that can be estimated is the affine camera, which is a linear version of the fundamental matrix. This last situation is directly analogous to the situation to which Stefan alluded, where a conic cannot be well estimated.

R. I. Hartley. I think that Phil has hit the nail on the head here. One may also note the distinction between computation of the fundamental matrix and determination of the configuration of the cameras relative to the points. Consider the case where a camera rotates, and also makes a very small (but non-zero) motion relative to the set of points being imaged. In this case, the two images differ by a planar projectivity, plus a small correction occasioned by the motion of the camera. This may be too small to measure accurately. In this case, one can characterize the camera rotation, and correctly infer that the motion of the camera is insignificant, at least on the scale of the points being imaged. However, if one wishes to compute the fundamental matrix, one is out of luck completely. It is notable in this case that with more than 8 points,

the two camera centres and the points need not lie on or close to a quadric (critical) surface, which means that critical surface configurations are not the only ones that can cause instabilities of the fundamental matrix.

In these cases, as with the conic computation, the problem is not with a choice of minimizing algebraic or geometric error to compute the matrix, rather that the problem instance itself is not well conditioned.

O. Faugeras (*INRIA, France*). Has Dr Hartley experienced any problem in the estimation of the trifocal tensor in the case where the three optical centres are in a line?

R. I. Hartley. I've never actually tried it, not explicitly, no.

J. Lasenby (*Department of Engineering, University of Cambridge, UK*). Has Dr Hartley had any problems in the trifocal tensor in calculating the epipoles? They are defined as the null vector, but the matrix might not have a determinant of zero.

R. I. Hartley. I haven't had a problem. Naturally, one computes the null vector in a least-squares sense as the singular vector corresponding to the smallest singular value. One expects problems only when the smallest and next-smallest singular values are close to being equal. This may occur either as a result of extreme noise distortion, or because the matrix actually has rank 1. Even then, there is a safety belt, since the epipole is computed as the common perpendicular to the nullspaces of the three cross-sections of the trifocal tensor (see proposition 4.1 of my paper). If one of the nullspaces is not well defined, then the other two still serve to define the epipole.

However, in a straight linear method for computing the camera matrices from the trifocal tensor, it is estimating the epipoles that is definitely the weak point of the algorithm. That is why, to get the best results, I then iterate, moving the epipoles to try to minimize the total error. This provides added robustness in cases where the epipoles are not stable.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES