Computation of the Quadrifocal Tensor

Richard I. Hartley

G.E. Corporate Research and Development
1 Research Circle,
Niskayuna, NY 12309,
USA

Abstract

This paper gives a practical and accurate algorithm for the computation of the quadrifocal tensor and extraction of camera matrices from it. Previous methods for using the quadrifocal tensor in projective scene reconstruction have not emphasized accuracy of the algorithm in conditions of noise. Methods given in this paper minimize algebraic error either through a non-iterative linear algorithm, or two alternative iterative algorithms. It is shown by experiments with synthetic data that the iterative methods, though minimizing algebraic, rather than more correctly geometric error measured in the image, give almost optimal results.

Keywords

Calibration and Pose Estimation, Stereo and Motion, Image Sequence Analysis, Algebraic error, Quadrifocal tensor

1 Introduction

In the study of the geometry of multiple views, the fundamental matrix and more recently the trifocal tensor have proven to be essential tools. The quadrifocal tensor which relates coordinates measured in four views is the natural extension of these techniques to four views. Because of the added stability of a fourth view, and the more tightly it constrains the position of reconstructed points in space, use of the quadrifocal tensor should lead to greater accuracy than two and three-view techniques. This hypothesis is supported by the results of Heyden ([6, 5]). However, the four-view tensor has not been given much attention in the literature. One of the main impediments to its use is the fact that the quadrifocal tensor is very greatly overparametrized, using 81 components of the tensor to describe a geometric confuguration that depends only on 29 parameters. This can lead to severe inaccuracies if additional constraints are not applied. One method of doing this was proposed by Heyden, who defined the reduced quadrifocal tensor. This has the effect of partially constraining the solution, but still leaves many (in fact 32) unresolved constraints.

In this paper, a method is given for computing the quadrifocal tensor and extracting camera parameters in a way so as to satisfy all constraints on the tensor, while at the same time minimizing the algebraic measurement error. The results obtained prove to be near optimal in terms of minimizing residual error in the measured image coordinates.

2 The Quadrifocal Tensor

The quadrifocal tensor was discovered by Triggs ([8]) and an algorithm for using it for reconstruction was given by Heyden ([6, 5]). However, because the quadrifocal tensor is not so widely understood as the fundamental matrix, or the trifocal tensor, a derivation of its properties and description of existing algorithms is given here. The only previously existing algorithm for computation of the quadrifocal tensor and extraction of the camera matrix is due to Heyden ([6, 5]). This algorithm will be described below, since most of the material will be needed for the description of a new algorithm.

Consider four cameras P, P', P'' and P''', and let $\mathbf{x}$ be a point in space imaged by the four cameras. Let the corresponding image points in the four images be denoted by $\mathbf{u}$, $\mathbf{u}'$, $\mathbf{u}''$ and $\mathbf{u}'''$ respectively. We write $\mathbf{u} \approx P\mathbf{x}$, and similarly for the other images. The notation $\approx$ indicated that the two sides of the equation are equal, up to an unknown scale factor. Taking account of the scale factor, there are constants k, k', k'' and k''' such that $k\mathbf{u} = P\mathbf{x}$, $k'\mathbf{u}' = P'\mathbf{x}$, and so on for the other two images. This set of equations

may be written as a single matrix equation as follows:

$$
\begin{bmatrix}
\mathbf{a}^1_. & u^1 & & \\
\mathbf{a}^2_. & u^2 & & \\
\mathbf{a}^3_. & u^3 & & \\
\hline
\mathbf{b}^1_. & & u'^1 & \\
\mathbf{b}^2_. & & u'^2 & \\
\mathbf{b}^3_. & & u'^3 & \\
\hline
\mathbf{c}^1_. & & & u''^1 \\
\mathbf{c}^2_. & & & u''^2 \\
\mathbf{c}^3_. & & & u''^3 \\
\hline
\mathbf{d}^1_. & & & & u'''^1 \\
\mathbf{d}^2_. & & & & u'''^2 \\
\mathbf{d}^3_. & & & & u'''^3
\end{bmatrix}
\begin{pmatrix}
\mathbf{x} \\
-k \\
-k' \\
-k'' \\
-k'''
\end{pmatrix} = 0
\tag{1}
$$

where vectors $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$ and $\mathbf{d}^i$ are the row vectors of the matrices P, P′, P″ and P″ respectively.

Since this equation has a solution, the matrix X on the left has rank at most 7, and so all 8×8 determinants are zero. Any determinant containing fewer than two rows from each of the camera matrices gives rise to a trilinear or bilinear relation between the remaining cameras. A different case occurs when we consider 8×8 determinants containing two rows from each of the camera matrices. Expansion of the determinant leads directly to a quadrilinear relationship of the form

$$
u^i u'^j u''^k u'''^l \epsilon_{ipw} \epsilon_{jqx} \epsilon_{kry} \epsilon_{lsz} Q^{pqrs} = 0_{wxyz}
\tag{2}
$$

where 0_{wxyz} is a zero tensor with four indices w, x, y and z, and the rank-4 tensor Q^{pqrs} is defined by

$$
Q^{pqrs} = \det
\begin{bmatrix}
\mathbf{a}^p_. \\
\mathbf{b}^q_. \\
\mathbf{c}^r_. \\
\mathbf{d}^s_.
\end{bmatrix}
\tag{3}
$$

The tensor ϵ_{ijk} is defined to be zero unless i, j and k are distinct, and otherwise equal to 1 or -1 depending on whether (ijk) is an even or odd permutation of (123). We use the tensor summation convention in which an index repeated in upper (contravariant) and lower (covariant) positions implies summation of all values $1, 2, 3$ of the index.

Note that the four indices of the four-view tensor are contravariant A discussion of tensors, contravariant and covariant indices is found in an appendix of [3]. Note that for this *quadrifocal tensor* there is no distinguished view as there is in the case of the trifocal tensor. There is only one quadrifocal tensor corresponding to four given views. More details of this derivation, and the equations that may be derived from it are given in [2].

Each point correspondence across four views gives rise via (2) to 81 equations in the entries of Q, one equation for each of the choices of indices w, x, y, z. Among these equations, there are 16 linearly independent equations in the 81 entries of the quadrifocal tensor Q. From a set of several point correspondences across four views one obtains a set of linear equations of the form $\mathbf{Aq} = 0$ where $\mathbf{q}$ is a vector containing the entries of

Q. Since Q is defined only up to scale, it may appear that from 5 point correspondences there are enough equations to compute Q. However, the sets of equations arising from different point correspondences have a linear relationship, as shown in [2]. It turns out that 6 points are required to solve for Q. With 6 points or more, one may solve a linear least-squares problem to find Q.

It is also possible to derive equations involving the quadrifocal tensor from line correspondences, or mixed line-point correspondences, as explained in [2]. All the techniques of this paper are applicable to line or mixed point-line correspondences as well, but we omit further mention of this point.

2.1 Algebraic Error

In the presence of noise, one can not expect to obtain an exact solution to an over-constrained set of equations of the form $\mathtt{Aq} = \mathbf{0}$ such as those that arise from a point correspondence in 4 views. The linear least-squares algorithm instead finds the unit-norm vector $\mathbf{q}$ that minimizes $||\mathtt{Aq}||$. The vector $\epsilon = \mathtt{Aq}$ 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 $\mathtt{A}$.

Consider an estimate for the quadrifocal tensor Q, represented by a vector $\mathbf{q}$, and let $\mathtt{A}$ be the matrix of equations corresponding to a set of point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i \leftrightarrow \mathbf{u}'_i \leftrightarrow \mathbf{u}'''_i$ across 4 views. The vector $\mathtt{Aq}$ is the *algebraic error vector* associated with the estimate $\mathbf{q}$, relative to the measurements. Thus, our goal is to find the unit norm vector $\mathbf{q}$ minimizing the algebraic error $||\mathtt{Aq}||$. In finding this minimum, the vector $\mathbf{q}$ may be allowed to vary freely over the whole linear space R^{81}, or be constrained to some subset of R^{81}. In any case, the principle is the same, namely that $\mathtt{Aq}$ is the algebraic error vector.

Constraints. The simplest manner of estimating the tensor Q is the linear least squares method, that finds the unit vector $\mathbf{q}$ that minimizes algebraic error, where $\mathbf{q}$ is allowed to vary over the whole space R^{81}. As with the fundamental matrix and the trifocal tensor, the tensor Q that one finds in this way will not in general correspond (as in (3)) to a set of four camera matrices. This will only be the case when Q is appropriately constrained. To be more precise, a quadrifocal tensor Q is determined by the four camera matrices. These have a total of 44 degrees of freedom. However, the quadrifocal tensor (just as the fundamental matrix and trifocal tensor) is unchanged by a projective transformation of space, since its value is determined only by image coordinates. Hence, we may subtract 15 for the degrees of freedom of a 3D projective transformation. The quadrifocal tensor depends on only 29 essential parameters. Thus, it must satisfy a total of $51 = 80 - 29$ constraints, in addition to the scale ambiguity. As one may see, the 81 entries of the quadrifocal tensor are very greatly redundant as far as describing the projective configuration of the cameras is concerned.

By constrast, the fundamental matrix has 8 entries up to scale, and must satisfy a single constraint ($\det F = 0$). The trifocal tensor has 27 entries and must satisfy 8 constraints, since the projective configuration of three cameras has $18 = 33 - 15$ degrees of freedom. Although one may perhaps hope to get reasonable results by ignoring the constraints on the fundamental matrix and even the trifocal tensor, it is clear that one can not ignore the 51 constraints of the quadrifocal tensor and hope to get reasonable results.

The method described in this paper for computation of the quadrifocal tensor is to find **q** that minimizes algebraic error, while at the same time being constrained to correspond to a set of four camera matrices, according to (3).

2.2 Minimization subject to constraints

In the case where the vector **q** is constrained to lie on a linear submanifold (an affine subspace) of R^{81}, a simple linear method exists for carrying out the minimization. The following algorithm was give in [4]. Since it is an essential part of our method for finding the quadrifocal tensor, it is repeated below for convenience.

Algorithm : Given a "measurement" matrix A and "constraint" matrix C find the unit norm vector **q** that minimizes $||\text{A}\text{q}||$ subject to constraints $\text{q} = \text{C}\text{a}$ for some vector **a**. Equivalently, minimize $||\text{ACa}||$ subject to $||\text{Ca}|| = 1$, then set $\text{q} = \text{Ca}$.

Solution :

1. Compute the SVD $\text{C} = \text{UDV}^\top$ such that the non-zero values of D appear first down the diagonal.

2. Let U′ be the matrix comprising the first r columns of U, where r is the rank of C. Further, let V′ consist of the first r columns of V and D′ consist of the r first rows and columns of D.

3. Find the unit vector **q′** that minimizes $||\text{AU}'\text{q}'||$. This is the singular vector corresponding to the smallest singular value of AU′.

4. The required vector **q** is given by $\text{q} = \text{U}'\text{q}'$, A vector **a** such that $\text{q} = \text{Ca}$ is given by $\text{a} = \text{V}'\text{D}'^{-1}\text{q}'$.

2.3 Geometric Distance

A common assumption is that measurement error is confined to image measurements, and image measurements conform to a gaussian error model. Given a set of measured correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i \leftrightarrow \mathbf{u}''_i \leftrightarrow \mathbf{u}'''_i$, the optimal estimate for the quadrifocal tensor under this error model is the one that satisifies equations of the form

$$\hat{u}_i^a \hat{u}_i'^b \hat{u}_i''^c \hat{u}_i'''^d \epsilon_{apw} \epsilon_{bqx} \epsilon_{cry} \epsilon_{dsz} Q^{pqrs} = 0_{wxyz} \tag{4}$$

as in (3) for each matched point, where $\hat{\mathbf{u}}_i, \dots, \hat{\mathbf{u}}'''_i$ are estimated image points that minimize the geometric error

$$\sum_i d(\hat{\mathbf{u}}_i, \mathbf{u}_i)^2 + d(\hat{\mathbf{u}}'_i, \mathbf{u}'_i)^2 + d(\hat{\mathbf{u}}''_i, \mathbf{u}''_i)^2 + d(\hat{\mathbf{u}}'''_i, \mathbf{u}'''_i)^2 \tag{5}$$

subject to the condition that (4) is satisfied exactly. Here, $d(\cdot, \cdot)$ represents Euclidean distance in the image. The quantity $d(\mathbf{u}_i, \hat{\mathbf{u}}_i)$ is known as the *geometric distance* between $\mathbf{u}_i$ and $\hat{\mathbf{u}}_i$. Thus the error to be minimized is the sum of squares of geometric distances between measured and projected points.

Geometric error may be minimized using the technique of bundle adjustment, to carry out a projective reconstruction of the scene. A method for doing this using sparse techniques to minimize time complexity is given in [1]. In general, minimization of geometric

error can be expected to give the best possible results (depending on how realistic the error model is). This algorithm is not implemented here for use in comparison with the algebraic minimization algorithm. Nevertheless, one may easily derive a theoretical bound (the Cramer-Rao lower bound) for the minimum error obtainable using the geometric error model. This bound is used to evaluate the algebraic minimization algorithms. Consider an estimation problem involving N measurements each of a Gaussian random variable with standard deviation σ, and d parameters, the minimum residual error (distance between measured and modelled values) is equal ([7]) to $\epsilon = \sigma(1 - d/N)^{1/2}$. In this particular case, with four images and n points, two coordinates per point, there are $8n$ measurement. Counting parameters, we count $3n$ for the coordinates of the points in space, plus 29 for the degrees of freedom of the quadrifocal tensor. Hence, the optimal residual error is

$$E = \sigma((5n - 29)/8n)^{1/2} \ . \tag{6}$$

2.4 The Reduced Measurement Matrix

In general, the matrix $\mathtt{A}$ may have a very large number of rows. As described in [4] it is possible to replace $\mathtt{A}$ by a square matrix $\hat{\mathtt{A}}$ such that $||\mathtt{A}\mathbf{q}|| = ||\hat{\mathtt{A}}\mathbf{q}||$ for any vector $\mathbf{q}$. Such a matrix $\hat{\mathtt{A}}$ is called a *reduced measurement matrix*. An efficient way of obtaining $\hat{\mathtt{A}}$ is to use the QR decomposition $\mathtt{A} = \mathtt{Q}\hat{\mathtt{A}}$, where $\mathtt{Q}$ has orthogonal columns and $\hat{\mathtt{A}}$ is upper-triangular and square.

In this way, all the information we need to keep about the set of matched data $\mathbf{u}_i \leftrightarrow \mathbf{u}_i' \leftrightarrow \mathbf{u}_i'' \leftrightarrow \mathbf{u}_i'''$ is contained in the single matrix $\hat{\mathtt{A}}$. If we wish to minimize algebraic error $||\mathtt{A}\mathbf{q}||$ as $\mathbf{q}$ varies over some restricted set of transforms, then this is equivalent to minimizing the norm $||\hat{\mathtt{A}}\mathbf{q}||$.

2.5 The reduced quadrifocal tensor

A method introduced by Heyden ([6, 5]) for reducing the number of unsatisfied constraints involving the quadrifocal tensor involves the *reduced quadrifocal tensor*. Heyden applied the same techniques to defined also a reduced fundamental matrix and reduced trifocal tensor, but these will not be considered in this paper.

Suppose that among the set of correspondences, three correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}_i' \leftrightarrow \mathbf{u}_i'' \leftrightarrow \mathbf{u}_i'''$ for $i = 1, \ldots, 3$ are selected. This selection should be done in such a way that points $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are not collinear, and neither are the corresponding points in the other images. Now, a projective transformation $\mathtt{T}$ exists that maps the points represented in homogeneous coordinates as $\mathbf{e}_1 = (1, 0, 0)^\top$, $\mathbf{e}_2 = (0, 1, 0)^\top$ and $\mathbf{e}_3 = (0, 0, 1)^\top$. to the points $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$. Note that $\mathtt{T}$ is not an affine transformation, since it does not keep the plane at infinity fixed. As such, it is not fully defined by three point correspondences. However, a simple choice of $\mathtt{T}$ is the one represented by a matrix

$$\mathtt{T} = \left[\begin{array}{ccc} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{array} \right] \tag{7}$$

where $(u_i, v_i, w_i)^\top$ are the coordinates of $\mathbf{u}_i$. One verifies easily that indeed $\mathtt{T}\mathbf{e}_i = \mathbf{u}_i$. We assume that each point $\mathbf{u}_i$ in the image is subjected to the transformation $\mathtt{T}^{-1}$, resulting in a new set of points. Let us assume that this transformation has been carried out,

and agree to denote the new transformed points by the same symbol $\mathbf{u}_i$. Thus, one has a set of image points $\mathbf{u}_i$ of which the first three points $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are equal to $\mathbf{e}_i$, $i = 1, \ldots, 3$.

Let the points in space mapping to the points $\mathbf{u}_i$ be denoted $\mathbf{x}_i$. Thus, $\mathbf{u}_i = \mathtt{P}\mathbf{u}_i$. We focus on the first three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. Since image points $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are not collinear (by assumption), neither are the points $\mathbf{x}_i$ that project to them. It is possible, therefore to select a projective coordinate frame in which these points have coordinates $\mathbf{x}_1 = (1, 0, 0, 0)^\top$, $\mathbf{x}_2 = (0, 1, 0, 0)^\top$ and $\mathbf{x}_3 = (0, 0, 1, 0)^\top$. Since these points map to image points $\mathbf{e}_i$, one verifies that the form of the projection matrix must be

$$
\mathtt{P} = \begin{bmatrix} p_1 & & & q_1 \\ & p_2 & & q_2 \\ & & p_3 & q_3 \end{bmatrix} .
\tag{8}
$$

We may make the further assumption that the centre of the camera is at the point $(0, 0, 0, 1)^\top$, which means that $\mathtt{P}(0, 0, 0, 1)^\top = (0, 0, 0)^\top$. Consequently, $q_1 = q_2 = q_3 = 0$.

Now, applying a similar argument to each of the other cameras, one sees that each of $\mathtt{P}$, $\mathtt{P}'$, $\mathtt{P}''$ and $\mathtt{P}'''$ has a similar form, consisting of a left hand diagonal block, plus an arbitrary 4-th column. Finally, one may multiply each of the camera matrices on the right by the matrix

$$
\begin{bmatrix} p_1^{-1} & & & -p_1^{-1}q_1 \\ & p_2^{-1} & & -p_2^{-1}q_2 \\ & & p_2^{-1} & -p_3^{-1}q_3 \\ & & & 1 \end{bmatrix} .
$$

This reduces the first matrix $\mathtt{P}$ to the particular simple form $[\mathtt{I} \mid \mathbf{0}]$ while retaining the form (8) of the other matrices. This is the reduced form for the camera matrices.

Notation : To avoid continually having to count the number of primes to distinguish the elements from the three cameras, we will write the four cameras as

$$
\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \quad ; \quad \mathtt{P}' = \begin{bmatrix} a_1 & & & a_1' \\ & a_2 & & a_2' \\ & & a_3 & a_3' \end{bmatrix}
$$

$$
\mathtt{P}'' = \begin{bmatrix} b_1 & & & b_1' \\ & b_2 & & b_2' \\ & & b_3 & b_3' \end{bmatrix} \quad ; \quad \mathtt{P}''' = \begin{bmatrix} c_1 & & & c_1' \\ & c_2 & & c_2' \\ & & c_3 & c_3' \end{bmatrix} .
\tag{9}
$$

We now consider the quadrifocal tensor Q^{ijkl} defined by (3), in terms of the matrices in (9). This tensor is known as the *reduced quadrifocal tensor*. Each entry is defined as a determinant made up of one row from each of the four camera matrices. Note that if one of the three indices 1, 2 or 3 is absent from the indices of Q^{ijkl}, then the corresponding column of the determinant contains only zeros, and the entry of Q^{ijkl} is zero. Thus, the only nonzero entries are those in which all three indices occur. Since in total there are four indices, one index must appear twice. Simple counting reveals that there are only 36 non-zero entries in the reduced quadrifocal tensor – 6 to account for the choice of which pair of the four indices i, j, k, l are the same, and 6 to account for the permutations of the indices $1, 2, 3$.

We may now summarize an algorithm due to Heyden ([6, 5]) for computing the reduced quadrifocal tensor from a set of point correspondences.

1. Select three point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i \leftrightarrow \mathbf{u}''_i \leftrightarrow \mathbf{u}'''_i$ for $i = 1, \ldots, 3$ and compute transformations $\mathtt{T}$, $\mathtt{T}'$, $\mathtt{T}''$ and $\mathtt{T}'''$ that map points $\mathbf{e}_i$ $i = 1, \ldots, 3$ to these selected points. Apply the inverse transformation to the complete set of points in each image, to obtain a new set of transformed point correspondences.

2. Each transformed point correspondence gives a set of linear equations in the 36 non-zero entries of the quadrifocal tensor, according to (2). One ignores the first three point correspondences, since they give null equations. Solve these equations in a least-squares manner to find the entries of the quadrifocal tensor.

Transforming the quadrifocal tensor.   The quadrifocal tensor corresponding to the original set of point correspondences (not the transformed set) may be retrieved by transforming the reduced quadrifocal tensor found by the above algorithm. To do that, we need to consider how the quadrifocal tensor transforms. Note that the quadrifocal tensor has 4 contravariant (upper) indices. Similarly, points, such as $\mathbf{u}_i$ have a contravariant index. This means that $\mathtt{Q}$ transforms in the same way as points. In particular, denote by $\hat{\mathbf{u}}_i$ the set of transformed points, $\hat{\mathbf{u}}_i = \mathtt{T}^{-1}\mathbf{u}_i$, and let $\hat{\mathbf{u}}'_i$, $\hat{\mathbf{u}}''_i$ and $\hat{\mathbf{u}}''_i$ be defined similarly. Suppose $\hat{\mathtt{Q}}$ is the quadrifocal tensor corresponding to the set of correspondences $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i \leftrightarrow \hat{\mathbf{u}}''_i \leftrightarrow \hat{\mathbf{u}}'''_i$, and Q is derived from $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i \leftrightarrow \mathbf{u}''_i \leftrightarrow \mathbf{u}'''_i$. Since $\mathbf{u}_i = \mathtt{T}\hat{\mathbf{u}}_i$, the tensor $\hat{\mathtt{Q}}$ also transforms via transform $\mathtt{T}$. More precisely, one finds that

$$Q^{ijkl} = \hat{\mathtt{Q}}^{abcd} T_a^i T_b'^j T_c''^k T_d'''^l \tag{10}$$

If $\hat{\mathtt{Q}}$ is the tensor computed from the transformed points, then (10) allows us to retrieve the tensor corresponding to the original points.

2.6   Retrieving the Camera Matrices

Following Heyden, we give a method for computing the camera matrices once the reduced tensor has been computed. In contrast with the methods for computing the camera matrices from the fundamental matrix or the trifocal tensor, in which one computes first the epipoles, in the method to be described, one computes first the diagonal elements of the matrices in (9), and then computes the final columns, which represent the epipoles. To understand this, we consider the two entries Q^{2311} and Q^{3211}. From (3) one finds that

$$Q^{2311} = \det \begin{bmatrix} & 1 & \\ & a_3 & a'_3 \\ b_1 & & b'_1 \\ c_1 & & c'_1 \end{bmatrix} = a_3(b_1 c'_1 - c_1 b'_1)$$

On the other hand, similarly, one finds that $Q^{3211} = -a_2(b_1 c'_1 - c_1 b'_1)$. Hence, one sees that the ratio $a_3 : a_2 = Q^{2311} : -Q^{3211}$. Continuing in this manner, one may solve for the diagonal elements of the matrices (9) by solving the following equations :

$$\begin{bmatrix} 0 & Q^{2311} & Q^{3211} \\ Q^{1322} & 0 & Q^{3122} \\ Q^{1233} & Q^{2133} & 0 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \mathbf{0}$$

$$\begin{bmatrix} 0 & Q^{2131} & Q^{3121} \\ Q^{1232} & 0 & Q^{3212} \\ Q^{1323} & Q^{2313} & 0 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \mathbf{0} \qquad (11)$$

$$\begin{bmatrix} 0 & Q^{2113} & Q^{3112} \\ Q^{1223} & 0 & Q^{3221} \\ Q^{1332} & Q^{2331} & 0 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \mathbf{0}$$

These equations have a pattern to them, which the reader may easily discover. (The first index of Q^{ijkl} corresponds to the column of each matrix in (11), and the repeated index corresponds to the row.) In this way, one computes the diagonal elements of (9). Each of the solution vectors $(a_1, a_2, a_3)^\top$, $(b_1, b_2, b_3)^\top$ and $(c_1, c_2, c_3)^\top$ may be chosen to have unit norm.

2.7 Retrieving the epipoles.

Once one knows the values of $(a_1, a_2, a_3)^\top$, $(b_1, b_2, b_3)^\top$ and $(c_1, c_2, c_3)^\top$, the values of the entries of Q are linear in the remaining entries $a_1', a_2', a_3', b_1', b_2', b_3', c_1', c_2', c_3'$. This is because these entries will all appear in the last column of the determinant (3) representing Q^{ijkl}. Hence the determinant expression can not contain higher order terms in these entries. The exact form of the linear relationship can be computed by cofactor expansion of the determinant expression (3) for Q^{ijkl} down the last column. We will be content to express it in the form $\hat{\mathbf{q}} = \mathtt{M}\mathbf{a}'$, where $\hat{\mathbf{Q}}$ is the vector of entries of the reduced quadrifocal tensor, and $\mathbf{a}'$ is the vector $(a_1', a_2', a_3', b_1', b_2', b_3', c_1', c_2', c_3')^\top$. The unit-norm least-squares solution of this set of equations gives the entries of vector $\mathbf{a}'$. Along with the values of $\mathbf{a} = (a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3)^\top$ previously computed, this gives a complete solution for the camera matrices, according to (9).

The complete algorithm (due to Heyden) for computing the camera matrices from the reduced quadrifocal tensor is therefore as follows.

1. Compute the diagonal entries of the camera matrices (9) by solving (the unit norm least-squares solution) of the equations (11). Denote the vector of diagonal entries by $\mathbf{a}$.

2. Express the entries of the reduced quadrifocal tensor in terms of the vector $\mathbf{a}'$ of last-column entries of the matrices in (11). This gives a set of equations $\hat{\mathbf{q}} = \mathtt{M}\mathbf{a}'$, where the entries of $\mathtt{M}$ are quadratic expressions in the entries of $\mathbf{a}$. Solve this set of equations (once more the unit-norm least-squares solution) to find $\mathbf{a}'$.

3. If required, the reduced quadrifocal tensor may be computed using (3). Finally, the quadrifocal tensor corresponding to the original data may be computed using (10). Alternatively, if one requires only the original camera matrices, they may be obtained by transforming the reduced camera matrices found in steps 1 and 2.

It is important to note that this method, along with the algorithm for finding the reduced quadrifocal tensor together give a method for computing a quadrifocal tensor corresponding to a valid choice of camera matrices, and hence satisfying all needed constraints. This is done, of course, without explicitly finding the form of the constraints.

3 Minimizing Algebraic Distance

The algorithm given in the last section, though apparently performing adequately ([6, 5]) has various weaknesses, from a computational viewpoint.

1. In computing the transformations $\mathtt{T}, \ldots, \mathtt{T}'''$ according to (7) one risks encountering the case where the three points used to define the transforms are nearly collinear. In this case $\mathtt{T}$ may be close to being invertible. This means that the positions of the transformed points computed by applying $\mathtt{T}^{-1}$ to the original data points may not be very stable.

2. In solving for the epipoles, using (11) the matrices involved will not generally be singular. The solution is found in effect by computing the nearest singular matrix (in Frobenius norm) and taking the null space of that matrix. However, there is really no theoretical justification for finding the nearest singular matrix in Frobenius norm. The problem is analogous with that encountered when enforcing the singularity constraint for the fundamental matrix.

3. In the computation of the vector $\mathbf{a}'$, we solve a set of equations once more using a least-squared method. However, once more the quantity that we are minimizing has no meaningful interpretation in terms of errors in the original data.

It will now be shown how we may avoid these problems in a way so as to minimize algebraic error. The result is that in carrying out the three steps of the algorithm one is minimizing always the same algebraic cost function, and numerical performance is improved.

3.1 Minimizing Algebraic Error

Solving for the fundamental matrix.   The transformation formula (10) is seen to be linear in the entries of $\hat{\mathtt{Q}}$. More specifically, we may write $\mathbf{q} = \mathcal{T}\hat{\mathbf{q}}$. In this equation $\hat{\mathbf{q}}$ and $\mathbf{q}$ are vectors of entries of the reduced and full quadrifocal tensors, and $\mathcal{T}$ is an 81×36 matrix. Let $\mathtt{A}$ be the reduced measurement matrix computed from the *original* point correspondences. One wishes to compute the unit-norm vector $\mathbf{q}$ that minimizes the algebraic error $||\mathtt{A}\mathbf{q}||$ subject to the constraint that $\mathbf{q}$ is derived from a reduced quadrifocal tensor according to the constraint $\mathbf{q} = \mathcal{T}\hat{\mathbf{q}}$. In other words, we wish to minimize $||\mathtt{A}\mathcal{T}\hat{\mathbf{q}}||$ subject to the condition $||\mathcal{T}\hat{\mathbf{q}}|| = 1$. This is the type of constrained minimization problem considered in section 2.2. It is important to note here, that the transformation $\hat{\mathtt{Q}} \mapsto Q$ given by (10) depends on the matrices $\mathtt{T}, \ldots, \mathtt{T}'''$ given by a formula such as (7) which is built from the coordinates of the matched points. It is not necessary ever to invert this matrix to find $\mathtt{T}^{-1}, \ldots, \mathtt{T}'''^{-1}$. In addition, the reduced measurement matrix $\mathtt{A}$ is formed from the coordinates of the original untransformed points. In fact, it is never necessary to invert the matrices $\mathtt{T}, \ldots, \mathtt{T}'''$ at all. Thus, one computes the reduced quadrifocal tensor, *while completely avoiding* the problem of inverting matrices, which could potentially be near-singular.

Solving for the epipoles.   It does not seem to be possible to find the diagonal entries of the reduced camera matrices in a way so as to minimize algebraic error, in a

linear manner. Therefore, we adopt the method of Heyden of finding the least-squares solution to equations (11). This leaves the problem of finding the final columns of the matrices, namely the vector $\mathbf{a}'$. As before, the reduced quadrifocal tensor $\hat{\mathsf{Q}}$ may be expressed linearly in terms of the entries of $\mathbf{a}'$, writing $\hat{\mathbf{q}} = \mathsf{M}\mathbf{a}'$, assuming as we do that the diagonals of the reduced camera matrices are known.

Now, we wish to find $\mathbf{q}$ of unit norm that minimizes $||\mathbf{A}\mathbf{q}||$ subject to the additional constraints that $\mathbf{q} = \mathcal{T}\hat{\mathbf{q}}$ and $\hat{\mathbf{q}} = \mathsf{M}\mathbf{a}'$. These last two constraints become a single constraint $\mathbf{q} = \mathcal{T}\mathsf{M}\mathbf{a}'$. Thus, the problem may be cast as follows : minimize $||\mathbf{A}(\mathcal{T}\mathsf{M})\mathbf{a}'||$ subject to the condition $||(\mathcal{T}\mathsf{M})\mathbf{a}'|| = 1$. Once more this is a minimization problem of the sort solved in section 2.2. The solution method provides values for either $\mathbf{a}'$ or $\mathbf{q}$. Thus, we may compute the quadrifocal tensor directly by computing $\mathbf{q}$, or we may retrieve the reduced camera matrices by computing $\mathbf{a}'$.

The complete proposed algorithm is therefore as follows.

1. Form the reduced measurement matrix $\mathbf{A}$ from the original data.

2. Obtain the transformation matrices $\mathtt{T}, \ldots, \mathtt{T}'''$ from three of the correspondences, using formula (7).

3. Compute the 81×36 transformation matrix $\mathcal{T}$ such that $\mathbf{q} = \mathcal{T}\hat{\mathbf{q}}$ from the transformation rule (10)

4. Solve the minimization problem : minimize $||\mathbf{A}\mathcal{T}\hat{\mathbf{q}}||$ subject to $||\mathcal{T}\hat{\mathbf{q}}|| = 1$ to find $\hat{\mathsf{Q}}$, an initial estimate of the reduced fundamental matrix.

5. Find the diagonal elements (vector $\mathbf{a}$) of the reduced camera matrices by solving (11).

6. Compute the 36×9 matrix M such that $\hat{\mathbf{q}} = \mathsf{M}\mathbf{a}'$, where $\mathbf{a}'$ is the 9-vector containing the elements of the last columns (representing epipoles relative to the first camera) of the reduced camera matrices.

7. Solve the minimization problem : minimize $||\mathbf{A}\mathcal{T}\mathsf{M}\mathbf{a}'||$ subject to $||\mathcal{T}\mathsf{M}\hat{a}'|| = 1$. From this we may derive the vector $\mathbf{q} = \mathcal{T}\mathsf{M}\hat{a}'$ corresponding to a full quadrifocal tensor.

8. If desired, we may compute the reduced camera matrices (9) from the vectors $\mathbf{a}$ and $\mathbf{a}'$. These may be transformed to camera matrices for the original set of data by left-multiplication by the transforms $\mathtt{T}, \ldots, \mathtt{T}'''$.

The quadrifocal tensor found in this way is a valid tensor satisfying all appropriate constraints, since it is derived from a set of camera matrices. In addition, it minimizes algebraic error subject to the two conditions :

1. The camera matrix diagonals have the value given by the 9-vector $\mathbf{a}$, computed at step 5 of the algorithm.

2. The three points used to compute transforms $\mathtt{T}, \ldots, \mathtt{T}'''$ correspond precisely.

3.2 Iterative Estimation

The algorithm of the last section gives a way of computing an algebraic error vector $\mathbf{Aq}$ assuming that the 9-vector $\mathbf{a}$ representing the diagonals of the reduced camera matrices is known. This mapping $\mathbf{a} \mapsto \mathbf{Aq}$ is a map from R^9 to R^81. This suggests an iterative approach in which the goal is to vary $\mathbf{a}$ in a way such as to minimize the algebraic error $\|\mathbf{Aq}\|$. Such an iteration may be carried out using a parameter minimization program such as the Levenberg-Marquardt algorithm.

One may observe that the solution that one obtains in this manner is a minimum subject to the transformations $\mathtt{T}, \ldots, \mathtt{T}'''$ being fixed. These transformations are derived (using (7)) from the coordinates of the first three measured points. One may further let the entries of the three transformation matrice $\mathtt{T}', \mathtt{T}'', \mathtt{T}'''$ vary to find an absolute minimum of algebraic error. It is not necessary to let the transformation $\mathtt{T}$ vary. Since $\mathtt{T}', \ldots, \mathtt{T}'''$ are determined by the coordinates $\mathbf{u}'_i, \mathbf{u}''_i, \mathbf{u}'''_i$ for $i = 1, \ldots 3$, this accounts for a further variable parameters, a total of 27 in all.

Note the advantage of this method of computing Q is that the iterative part of the algorithm is of fixed size independent of the number of point correspondences involved.

3.3 4 points on a plane

It may be observed that if 4 of the points are known to lie on a plane, then the minimization of algebraic error can succeed without iteration in a single step. (We do not consider the case where three of these points are collinear.) To be specific, one may choose the first three correspondences to have homogeneous image coordinates $\mathbf{e}_1 = (1, 0, 0)^\top$, $\mathbf{e}_2 = (0, 1, 0)^\top$, $\mathbf{e}_3 = (0, 0, 1)^\top$ as before and the fourth point can be chosen as $\mathbf{e}_4 = (1, 1, 1)$. In a real computation, one needs to find transformations $\mathtt{T}, \ldots, \mathtt{T}'''$ that map these basis points to the actual image points. Now, since the points are known to be coplanar, one may assume that they lie on the plane at infinity, as the points $(1, 0, 0, 0)^\top$, $(0, 1, 0, 0)^\top$, $(0, 0, 1, 0)^\top$ and $(1, 1, 1, 0)^\top$. One assumes further that the first camera centre is at the origin $(0, 0, 0, 1)^\top$. In this case, it is easily verified that the camera matrices have the form (9) in which the diagonal elements are all equal to 1. This means that we can skip steps 4 and 5 of the algorithm of section 3.1. The complete algorithm is as follows :

1. Compute the reduced measurement matrix from the original data.

2. Compute the transforms $\mathtt{T}, \ldots, \mathtt{T}'''$ that take the image basis points $\mathbf{e}_1, \ldots, \mathbf{e}_4$ to the measured coordinate values.

3. Compute the 81×36 matrix $\mathcal{T}$ from (10) as before, such that $\mathbf{q} = \mathcal{T}\hat{\mathbf{q}}$.

4. Compute the 36×9 matrix $\mathtt{M}$ such that $\hat{\mathbf{q}} = \mathtt{M}\mathbf{a}'$, assuming that the diagonals of the reduced camera matrices (9) are all 1. This matrix has a given fixed form independent of the data.

5. Compute $\mathbf{a}'$ by solving the problem : minimize $\|\mathtt{A}\mathcal{T}\mathtt{M}\mathbf{a}'\|$ subject to $\|\mathcal{T}\mathtt{M}\mathbf{a}'\| = 1$. The quadrifocal tensor is given by $\mathbf{q} = \mathcal{T}\mathtt{M}\mathbf{a}'$.

The tensor derived in this way minimizes algebraic error, subject to the first four point correspondences (those for the coplanar points) being correct.

One should note that essentially the same technique works for the fundamental matrix and the trifocal tensor. If there exist four points known to lie in a plane, then there exists a straight-forward linear method that finds the matrix, or tensor, satisfying all constraints and minimizing algebraic error.
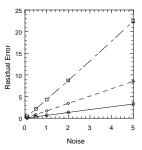
4 Experimental Results

The new algorithm for computation of the quadrifocal tensor was was tested with synthetic data. The data were created to simulate a standard 35mm camera with a 35mm focal length lens. A set of points were synthesized inside a sphere of radius 1m, and cameras were located at a distance of about 2.5m from the centre of the sphere aimed at the point set from random directions. 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 $35\mu m$ for a $35mm$ camera. The expected optimum value of the residual error was computed using formula ((6)). The results are shown in Fig 1.

The results given in Fig 1 are for a set of arbitrarily placed cameras and arbitrarily chosen points. An experiment was carried out to determine how the algorithms performed in near-critical configurations. A similar configuration of points and cameras was chosen as in the previous test. However, the first camera centre was located on the line through the first two world points $\mathbf{x}_1$ and $\mathbf{x}_2$. The result of this is that the points $\mathbf{u}_1$ and $\mathbf{u}_2$ (the images with respect to the first camera) are close together (in fact without noise, they coincide). Thus the transformation $\mathtt{T}$ is close to being singular. The result of this experiment is shown in Fig 2.

Iterative Algorithms   We now compare the iterative algorithms with the non-iterative algorithm. The results are shown if Fig 3. Two iterative algorithms are compared. In the first algorithm, the transformations $\mathtt{T}, \ldots \mathtt{T}'''$ are kept fixed, and the 9 diagonal entries of the reduced camera matrices vary. Thus, there are 9 varying parameters and 81 measured values. In the second method the transformation matrices $\mathtt{T}', \mathtt{T}'', \mathtt{T}'''$ are allowed to vary also. There are 27 varying parameters in this case. This second method should be expected to give better results.

5 Conclusion

The method of computing the quadrifocal tensor by minimizing algebraic error gives results almost indistinguishable from minimizing geometric error, and almost optimal. It avoids singular configurations in which Heyden's algorithm fails entirely, and in general gives numerically superior results. It is true that it is possible to be careful in selecting the three points used in Heyden's method for defining the reduced tensor. If this is done, then it is possible to avoid catastrophic failure of the algorithm. However this involves additional uncertainties in the algorithm. The present algorithm avoids such difficulties by avoiding the problems with singular configurations of points altogether.
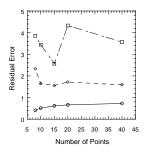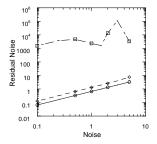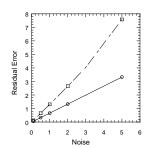
Figure 1: *The residual error achieved by the algorithm of this paper is compared with the optimal value as well as with the algorithm of Heyden. Each of the graphs shows the residual error RMS-averaged over 200 runs. In the left hand graph, the residual error is plotted against the amount of injected noise, for a set of 20 points. The right hand graph shows the effect of varying the number of points used to compute the tensor. It starts with 7 points, since none of the algorithms seems to perform well with just 6 points. The noise level is fixed at 1 pixel (in each coordinate direction). In each case, the lower plot is the optimum error given by (6)) while the two upper plots are the residual error for the algorithm of this paper (middle curve) and for Heyden's algorithm (top curve). One sees that the presently considered algorithm performs significantly better than Heyden's algorithm, but misses the optimal result by about a factor of 2. Note that the error is closely proportional to the amount of injected noise.*
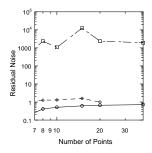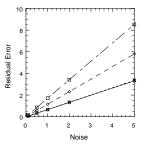
Figure 2: *In near singular cases, the present algorithm still performs well, as shown in these graphs. On the left is shown the residual as a function of increasing noise, and on the right the residual is plotted against the number of points. These are shown as log-log plots, so that the top curve will fit on the graph. It is seen that in this case Heyden's algorithm (represented by the top curve) fails completely. The center graph is a linear plot of just the optimal and the present non-iterative algorithm results.*
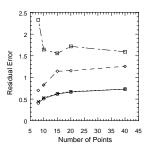
Figure 3: *These graphs show the performance of iterative algorithms for computing the quadrifocal tensor. The data set used is the same as for Fig 1. Each graph shows 4 curves. From top to bottom they are (i) the non-iterative algorithm, (ii) the 9-parameter iterative method, (iii) the 27 parameter iterative method and (iv) the theoretical optimal method. As may be seen, the 27-parameter iterative method gives very nearly optimal results (in fact the two last curves are indistinguishable). The two graphs shown are for varying noise and 20 points (on the left), and for fixed noise of 1 pixel and varying numbers of points (right).*

References

[1] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Applications of Invariance in Computer Vision : Proc. of the Second Joint European - US Workshop, Ponta Delgada, Azores – LNCS-Series Vol. 825, Springer Verlag*, pages 237–256, October 1993.

[2] Richard I. Hartley. Multilinear relationships between coordinates of corresponding image points and lines. In *Proceedings of the Sophus Lie Symposium, Nordfjordeid, Norway (never published)*, 1995.

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

[4] Richard I. Hartley. Minimizing algebraic error in geometric estimation problems. In *Proc. International Conference on Computer Vision*, pages 469 – 476, 1998.

[5] Anders Heyden. *Geometry and Algebra of Multiple Projective Transformations*. PhD thesis, Department of Mathematics, Lund University, Sweden, December 1995.

[6] Anders Heyden. Reconstruction from multiple images using kinetic depths. *International Journal of Computer Vision*, to appear.

[7] K. Kanatani. *Statistical Optimization for Geometric Computation : Theory and Practice*. North Holland, Amsterdam, 1996.

[8] Bill Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. unpublished report, 1995.