3D Model Acquisition from Extended Image Sequences

29/10/95

Paul Beardsley, Phil Torr and Andrew Zisserman
Robotics Research Group

Report No. OUEL 2089/96

University of Oxford
Department of Engineering Science
Parks Road
Oxford OX1 3PJ
U.K.

Tel: +44 865 273127
Fax: +44 865 273908
Email: pab/phst/az@robots.oxford.ac.uk

3D Model Acquisition from Extended Image Sequences

Paul Beardsley, Phil Torr and Andrew Zisserman
Robotics Research Group
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, UK.
tel. +44 1865 273154, fax. +44 1865 273908
email [pab,phst,az]@robots.oxford.ac.uk

Abstract

This paper describes the extraction of 3D geometrical data from image sequences, for the purpose of creating 3D models of objects in the world. The approach is uncalibrated - camera internal parameters and camera motion are not known or required.

Processing an image sequence is underpinned by token correspondences between images. We utilise matching techniques which are both robust (detecting and discarding mismatches) and fully automatic. The matched tokens are used to compute 3D structure, which is initialised as it appears and then recursively updated over time. We describe a novel robust estimator of the trifocal tensor, based on a minimum number of token correspondences across an image triplet; and a novel tracking algorithm in which corners and line segments are matched over image triplets in an integrated framework.

Experimental results are provided for a variety of scenes, including outdoor scenes taken with a hand-held camcorder. Quantitative statistics are included to assess the matching performance and structure accuracy. Renderings of the 3D structure enable a qualitative assessment of the results.

Keywords: structure from motion, uncalibrated cameras, tracking.

1 Introduction

The aim of this work is to recover 3D models from long uncalibrated monocular image sequences. These models will be used for graphics and virtual reality applications. The sequences are generated by circumnavigating the object of interest (e.g. a house) acquiring images with a camcorder. Neither the internal calibration of the camera or the motion of the camera are known. In particular the motion is unlikely to be smooth. "Long" here is of the order of hundreds of images, with a finite displacement (several cms) between each frame. The focus of this paper is the matching and tracking of image primitives. This underpins structure recovery in such extended sequences.

We build on the work of a number of previous successful systems which have recovered structure and motion from tracked image primitives (tokens). Coarsely these systems can be divided into those that use sequential and those that use batch updates. Sequential includes in particular DROID (corner-based, calibrated camera) [11, 10], Ayache and Lustman (line-based, calibrated stereo) [2], Zhang and Faugeras (line-based, calibrated stereo) [30], and Robert, Buffa and Hebert (corner-based, uncalibrated stereo) [22]; with batch methods including Tomasi and Kanade [24] (corners, weak perspective), Mohr[19] (corners, uncalibrated) and Hartley[17] (corners, uncalibrated). In our case the matching and structure update are sequential (though over image triplets, rather than the more usual image pairs), with the possibility of batch update at the end.

The finite base line between views outlaws the utilisation of a simple token matching strategy between consecutive images, such as nearest neighbour. It is in this area of tracking technology [29] that we have made the most significant innovations:

1. Corner and line segment tokens are matched simultaneously over image triplets in an integrated framework, by employing the trifocal tensor [18].

2. A robust estimator for the trifocal tensor is developed, with the tensor instantiated over three views using a minimal set (6) of point matches and a least median squares scheme.

In a typical image triplet, 150+ corners and 10+ line segments are matched, with the robust scheme ensuring protection against outliers [25, 5] (mismatches) and independently moving objects [27].

Two important advantages of the method described here are that the camera model covers a full perspective projection, not its affine approximation (weak or para-perspective)

as in [24], and no knowledge of camera internal parameters or relative motion is required. However, a consequence is that the 3D structure recovered is projective, rather than Euclidean [6, 13]. A method for reducing this projective ambiguity to Euclidean is given in section 6.

In the remainder of the paper: Section 2 summarises the mathematical representation and notation used; Section 3 provides an overview of both tracking and structure recovery; Section 4 covers token matching and tracking in detail, including a comparison of matching strategies; Section 5 describes the robust computation of the trifocal tensor, and compares the performance of robust estimators based on 6 and 7 points; Section 6 describes structure initialisation and recovery. Results are presented for a variety of real scenes, with an assessment of matching performance (lifetime of tracked tokens, total number of matches), and examples of the recovered structure. All of the processing (matching, structure recovery etc), is automatic, involving no hand picked points, and is robust to mismatches.

2 Mathematical representation and notation

Notation for the camera model and multi-view geometry is now introduced. This material is mainly review, and draws heavily on [6, 7, 13, 20, 18].

Camera projection matrices Perspective projection from 3D projective space $\mathcal{P}^3$ to the image plane $\mathcal{P}^2$ is modelled by a 3×4 matrix $\mathtt{P}$

$$\mathbf{x} = \mathtt{P}\mathbf{X} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, x_3)^\top$ and $\mathbf{X} = (X, Y, Z, 1)^\top$ are the homogeneous coordinates of an image point and 3D point respectively. For homogeneous quantities '=' indicates equality up to a non-zero scale factor.

Image lines $\mathbf{l}$ are represented by 3-vectors, $\mathbf{l} = (l_1, l_2, l_3)^\top$. The line $\mathbf{l}$ backprojects to a plane π in $\mathcal{P}^3$ represented by the 4-vector $\pi = \mathtt{P}^\top\mathbf{l}$. A line in $\mathcal{P}^3$ is represented by the 4×2 matrix $\mathtt{L}$, where the columns of $\mathtt{L}$ are two points lying on the line. Any linear combination of the columns of $\mathtt{L}$ is a point on the line. Alternatively, $\mathtt{L}$ is the rank 2 null space of the 2×4 matrix formed from the 4-vectors of two planes which intersect on the line.

The camera optical centre $\mathbf{Q} = (\mathbf{t}^\top, 1)^\top$ projects as $\mathtt{P}\mathbf{Q} = \mathbf{0}$, and it is convenient to partition the projection matrix $\mathtt{P}$ as

$$\mathtt{P} = [\mathtt{M}| \ -\mathtt{M}\mathbf{t}] \tag{2}$$

This partitioning is valid provided the left 3×3 matrix $\mathtt{M}$ is not singular, which requires the optical centre not to lie on the plane at infinity.

Image pairs - bilinear relations For two cameras with $\mathbf{x} = \mathtt{P}\mathbf{X}$ and $\mathbf{x}' = \mathtt{P}'\mathbf{X}$, corresponding points in the two images satisfy the epipolar constraint

$$\mathbf{x}'^\top \mathtt{F}\mathbf{x} = 0 \tag{3}$$

where $\mathtt{F}$ is the 3×3 *fundamental matrix*, with maximum rank 2. This is the bilinear relation in the homogeneous coordinates of the corresponding points in two images. The epipolar line in image 2 corresponding to $\mathbf{x}$ is $\mathbf{l}' = \mathtt{F}\mathbf{x}$, and similarly in image 1 corresponding to $\mathbf{x}'$ is $\mathbf{l} = \mathtt{F}^\top\mathbf{x}'$, where $\mathbf{l}, \mathbf{l}'$ are homogeneous line vectors. The epipoles in the first and second image are obtained from the right and left null-spaces of $\mathtt{F}$ respectively, $\mathtt{F}\mathbf{e} = 0$ and $\mathtt{F}^\top\mathbf{e}' = 0$.

It is always possible [15] to choose

$$\mathtt{P} = [\mathtt{I}| \ \mathbf{0}] \qquad \mathtt{P}' = [\mathtt{M}'| \ \mathbf{e}']$$

which corresponds to the world coordinate system having its origin at the optical centre of the first camera, and its axes aligned with the camera axes. Then

$$\mathtt{F} = [\mathbf{e}']_\times \mathtt{M}' = \mathtt{M}'^{-\top}[\mathtt{M}'^{-1}\mathbf{e}']_\times$$

where $[\mathbf{u}]_\times$ is the matrix representation of the vector product, i.e. $[\mathbf{u}]_\times\mathbf{v} = \mathbf{u} \times \mathbf{v}$.

Image triplets - trilinear relations Corresponding points in three images, and corresponding lines in three images, satisfy trilinear relations which are encapsulated in the trifocal tensor.

The trifocal tensor is a $3 \times 3 \times 3$ homogeneous tensor $\mathtt{T}_{ijk}$. If the image of a point $\mathbf{X}$ in $\mathcal{P}^3$ is $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{x}''$ in each of three images, then

$$x''_l = x'_i \sum_{k=1}^{k=3} x_k \mathtt{T}_{kjl} - x'_j \sum_{k=1}^{k=3} x_k \mathtt{T}_{kil},$$

for all $i, j = 1 \ldots 3$. Given $\mathtt{T}_{ijk}$ a point can be transferred to the third image from correspondences, $\mathbf{x}$ and $\mathbf{x}'$, in the first and second.

Similarly, if the image of a line in $\mathcal{P}^3$ is $\mathbf{l}$, $\mathbf{l}'$ and $\mathbf{l}''$ in each of three images, then

$$l_i = \sum_{j=1}^{j=3} \sum_{k=1}^{k=3} l'_j l''_k \mathtt{T}_{ijk}$$

i.e. the same tensor can be used to transfer both points and lines.

The projection matrices for the three views $\mathtt{P}, \mathtt{P}', \mathtt{P}''$ uniquely determine T_{ijk}. In detail, if $\mathtt{P} = [\mathtt{I}|\ \mathbf{0}]$ and p'_{ij}, p''_{ij} are the elements of $\mathtt{P}'$ and $\mathtt{P}''$ respectively, then

$$T_{ijk} = p'_{ji} p''_{k4} - p'_{j4} p''_{ki} \tag{4}$$

Conversely, given T_{ijk}, then the three projection matrices, fundamental matrices between view pairs, and epipoles can be recovered.

3 Overview of token matching and structure recovery

In this section we summarise the overall processing - from token extraction, to matching, to structure recovery - where the basic unit is an image triplet, and the trifocal tensor is the work horse. It is assumed that the scene is largely static, with the camera moving. No *a priori* information on camera internal parameters or motion is assumed, other than a threshold on the maximum disparity of tokens between images. The approach builds on earlier systems (e.g. [3]) where the basic unit was an image pair.

The correspondences between image corners $\mathbf{x}$ and lines $\mathbf{l}$ in the sequence of images are used to recover structure $\mathbf{X}$ and $\mathtt{L}$ respectively of the scene. In the case of unknown calibration, as here, structure is recovered modulo a projective transformation. That is, if the Euclidean structure of the scene is $\mathbf{X}_E$, $\mathtt{L}_E$, the recovered structure is

$$\begin{aligned} \mathbf{X} &= \mathtt{H}\mathbf{X}_E \\ \mathtt{L} &= \mathtt{H}\mathtt{L}_E \end{aligned}$$

where $\mathtt{H}$ is a non-singular 4×4 matrix which is the same for all points and lines, but undetermined.

3.1 Feature extraction

Two types of image primitives are used - corners and line segments - extracted independently in each image. Corners are detected to sub-pixel accuracy using the Harris corner detector [12]. Lines are detected by the standard procedure of: Canny edge detection [4]; edge linking; segmentation of the chain at high curvature points; and finally, straight line fitting to the resulting chain segments. The straight line fitting is by orthogonal regression with a very tight tolerance. This tolerance ensures that only actual line segments are extracted, i.e. that curves are not piecewise linear approximated.

If the camera or object is moving during acquisition, features are motion blurred, and in particular are sampled at separate time instances by the interlacing of the image lines. For this reason it is important to separate each image frame into its two interlaced fields, and process the fields individually.

3.2 Matching primitives over image triplets

Corners and line segments are matched over an image triplet in an integrated scheme, with matched corners providing support for line segment matching and vice-versa.

Simple token matching based only on similarity of image primitive attributes and proximity, will inevitably produce mismatches. For image *pairs*, the fundamental matrix provides a constraint for identifying mismatches. The fundamental matrix encodes the epipolar geometry between the views — given a corner in image 1, its match in image 2 must lie on the corresponding (epipolar) line. For *triplets* of images, there is a more powerful constraint available, and that is the geometry encoded in the trifocal tensor. Given a primitive matched in two images, the position of that primitive in the third image is defined. Analogous to the two image case, mismatches across the three images can be eliminated because they disagree with the constraint.

Thus, there is a natural symbiosis between a 2-image matching scheme and the computation of the fundamental matrix, and a 3-image matching scheme and the computation of the trifocal tensor. Furthermore the trifocal tensor allows the use of both corners and lines, unlike the fundamental matrix which is applicable to corners only. This is a major motivation for adopting the 3-image scheme. The overall approach is:

1. Corners are matched between image pairs, and simultaneously `F` is computed, using a robust scheme.

2. These pairwised matched corners, together with putative line matches over the triplet, are used to match primitives across image triplets and simultaneously compute `T` robustly.

Details are given in section 4. The outcome is a set of corner and line matches over the image triplet that is consistent with the trifocal tensor. In turn the tensor is used to compute projection matrices for structure computation, as described below.

3.3 3D structure initialisation

At the start of a sequence, the trifocal tensor from the first triplet of images is used to generate three camera matrices for those images, and matched corners and line segments are then used to instantiate estimates of 3D point and line structure. In a similar manner, new points and line segments in 3D are initialised whenever new structure becomes visible in the course of the sequence. Details are given in section 6.

3.4 3D structure update

After initialisation, an update process is employed for each new image added to the sequence. Matching between the last image and the new image provides a correspondence between existing 3D structure and the new image tokens, enabling the computation of the camera matrix for the new image and thereby determining the new camera position relative to the existing world coordinate frame. Existing estimates of 3D structure are updated using an Extended Kalman Filter (EKF).

3.5 Tracking primitives over sequences

The matching of primitives described in section 3.2 applies at the start of a sequence when there is no existing estimate of 3D structure. Once 3D structure has been obtained, it can be used to aid the matching process for each new image added to the sequence. The augmented matching scheme is described in the next section.

Image primitives matched across a triplet which do not correspond to existing 3D structure, are images of structure which has only become visible at this point in the sequence. These matched primitives (corners and line segments) are used to initialise new 3D structure.

4 Matching strategies

Corners (representing the projection of 3D points) and lines have complementary properties for matching over images. Between image pairs there is a geometrical constraint (epipolar geometry) available for points, but not lines. However, a corner has position (geometry) but few other image attributes — corners are typically matched using cross-correlation alone. In contrast, a line segment has a number of attributes which can be used fairly reliably for matching: including its geometry (position, orientation, length) and photometric properties such as intensity gradient averaged along its length. Consequently, although there is not a powerful geometric constraint available for line matching over image pairs (as there is for corners) the number of mismatches is far less than that of unguided corner matching. However, over image triplets the trifocal tensor provides a powerful geometric constraint for lines.

The methodology for matching is essentially the same for the cases of: corners between image pairs, corners and lines between image triplets, and image corners with 3D point structure. There are three distinct stages. We illustrate these stages first for the matching of corners between image pairs, followed by the other two cases.

4.1 Matching corners between image pairs

1. **Seed correspondences by unguided matching**
   The aim is to obtain a small number of reliable seed correspondences. Given a corner at position (x, y) in the first image, the search for a match is centred on (x, y) in the second image with a threshold on maximum disparity. The strength of candidate matches is measured by cross-correlation. The threshold for match acceptance is deliberately conservative to minimise incorrect matches.

2. **Robust computation of a geometric constraint**
   There is a potentially a significant presence of mismatches amongst the seed matches. The aim here is to obtain set of "inliers" consistent with the geometric constraint using a robust technique — RANSAC has proved the most successful. In this case the geometric constraint is the epipolar geometry. A putative fundamental matrix (up to three solutions) is computed from a random set of 7 corner correspondences (the minimum number required to compute a fundamental matrix). The support for this fundamental matrix is determined by the number of correspondences in the

seed set within a threshold distance of their epipolar lines. The fundamental matrix with the largest support is accepted. The outcome is a set of corner correspondences consistent with the fundamental matrix, and a set of mismatches (outliers). The fundamental matrix is then reestimated from the inliers to improves its accuracy.

3. **Guided matching**
   The aim here is to obtain additional matches consistent with the geometric constraint. The constraint provides a far more restrictive search region than that used for unguided matching. Consequently, a less severe threshold can be used on the matching attributes. In this case, matches are sought for unmatched corners searching only epipolar lines. This generates a larger set of consistent matches.

The final two steps are repeated until the number of matches stabilises.

Typically the number of corners in a 512×512 image of an indoor scene is about 300, the number of seed matches is about 100, and the final number of matches is about 200. Using corners computed to sub-pixel accuracy, the typical distance of a point from its epipolar line is $\sim$0.2-0.4 pixels.

4.2 Matching points and lines between image triplets

The same three steps are used over image triplets, with the geometric constraint provided by the trifocal tensor.

1. **Seed correspondences by unguided matching**
   For lines, seed correspondences over the three images are obtained by matching on a number of attributes (see section 4.4). For corners, seed correspondences between images 1 and 2, and 2 and 3 are obtained using the fundamental matrix as described above.

2. **Robust computation of a geometric constraint**
   A full description of this method is given in section 5. Briefly, a putative trifocal tensor (up to three solutions) is computed from a random set of six seed point correspondences. Least median squares is employed as the method of robust estimation. The support for the putative tensor is measured by the median error over correspondences in the seed set. The error involves contributions from both corners

and lines. The tensor with the least median error is chosen, and reestimated using the consistent point and line correspondences.

3. **Guided matching**
   Corner and line matching is then resumed, but now with a far more restrictive search area — for each pair of putative matches, only a region about the predicted position in the third image need be searched. This generates a larger set of consistent matches.

Here, both points and lines contribute to the estimate of the geometric constraint, and in turn the one constraint is used to search for both corner and line correspondences.

Typically the number of seed matches over a triplet is about corners, and 10 lines. The final number of matches is about 150 and 10 respectively. Using corners computed to sub-pixel accuracy, the typical distance of a corner/line from its transferred position is $\sim$1.5 pixels.

4.3 Matching between image corners and 3D points

The previous two matching schemes were for image to image matching. Once an estimate of 3D structure is available however (at any stage in the image sequence after the initialisation phase is completed) then it is possible to use the 3D structure to aid the matching. This augmented scheme is carried out whenever a new image arrives, and applies to the matching between the last image and the new image of the sequence. The outcome are matches over the triplet, and also between the 3D points and corners in the new image.

1. **Seed correspondences by unguided matching**
   As in the matching of corners between image pairs, section 4.1.

2. **Robust computation of a geometric constraint**

   a. Robust computation of the fundamental matrix
   As in the matching of corners between image pairs, section 4.1.

   b. Robust computation of the camera matrix
   The set of matches obtained above provide a correspondence between the existing 3D point structure and the new image corners. RANSAC is used to compute the

camera matrix `P`, which projects the 3D points of this set onto the new image. A putative projection matrix is computed from a random sample of 6 correspondences. The support for this matrix is measured by the number of correspondences in the set that are within a threshold distance between measured and projected position.

Existing 3D points and lines which have passed behind the focal plane of the camera, making it certain that they cannot appear in the new image, are identified [16] and not used in the computation.

3. **Guided matching**
   Corner matching is resumed. `P` is used to project any unmatched 3D points onto the new image, and a match is searched for around the projected position. This generates a larger set of consistent matches.

Typically, we find that the majority of matches are obtained in the initial matching stage when the fundamental matrix is used. However, the use of the camera matrix computation can add 5-10 matches in a total of 200. The r.m.s. error between projected 3D structure and actual image tokens in the new image is $\sim$0.5 pixels.

4.4 Implementation details

Unguided line segment matching Each line segment in the first image is matched against line segments in the second image subject to a threshold on maximum disparity - every point on the line segment in the second image must lie within the specified disparity of some point on the line segment in the first image. Matching is then subject to a threshold on maximum change of line orientation between the images, and a threshold on "proportion of overlap". This proportion is obtained by a perpendicular projection of the first line segment onto the second. This is repeated in the reverse direction, and the poorest proportion of overlap used to determine acceptance of the match. Finally, for each potential match accepted by the above tests, the percentage difference in mean gradient of the line segments is computed, and the closest match is taken subject to an acceptance threshold. Typically we use an angle threshold of 10°, an overlap threshold of 0.5, and the threshold on percentage difference in mean gradient is 10%.

Numerical conditioning Three basic computations which occur repeatedly are the determination of the fundamental matrix, the trifocal tensor, and the camera projection

matrix. In each case using "raw" image coordinates and, in particular, homogeneous image points of the form $\mathbf{x} = (u, v, 1)$ (where (u, v) are pixel coordinates) leads to poorly conditioned numerical computations. The problem arises because the first two components of the homogeneous vectors are typically two orders of magnitude larger than the final component, so the associated matrices will have a bad condition number. Conditioning is substantially improved by normalising the image measurements by transforming the measurements to a frame with origin at the centre of mass of the points, with scaling such that the third component of the homogeneous vector is similar in magnitude to the first two components [9]. A similar normalisation is also employed for 3D points $\mathbf{X}$.

4.5 Comparison of pairwise and triplet based matching

4.5.1 Experiment I: Number of matches/mismatches

Corner matches between three images can be established by two methods:

1. **Structure based** This is the matching scheme described in section 4.1 for image pairs, supported by the use of structure. Explicitly, corners are matched between image pairs 1 & 2 and 2 & 3 using a robust estimator for $\max F$; 3D point structure is instantiated from the matches between 1 & 2; and a robust estimator is used for `P` to match this structure to image 3, based on the 2 & 3 matches.

2. **Trifocal tensor based** This is the matching scheme described in section 4.2 above for image triplets.

We assess and compare the two matching schemes by two measures — the number of matches, and the number of mismatches.

Figure 1 shows three consecutive images of a model house, processed by the 2-image (i.e. structure based) and 3-image schemes (i.e. trifocal tensor based), and with matched corners superimposed on the images. For the 2-image scheme, only those points which survived over all three images are shown, to enable a proper comparison with the 3-image approach. There is very little difference in the number and distribution of the matches found. Furthermore, there are no mismatches under either scheme. Figure 2 shows the same information for an outdoor scene. There are a few mismatches (about 2 mismatches in about 80 points) under each scheme.

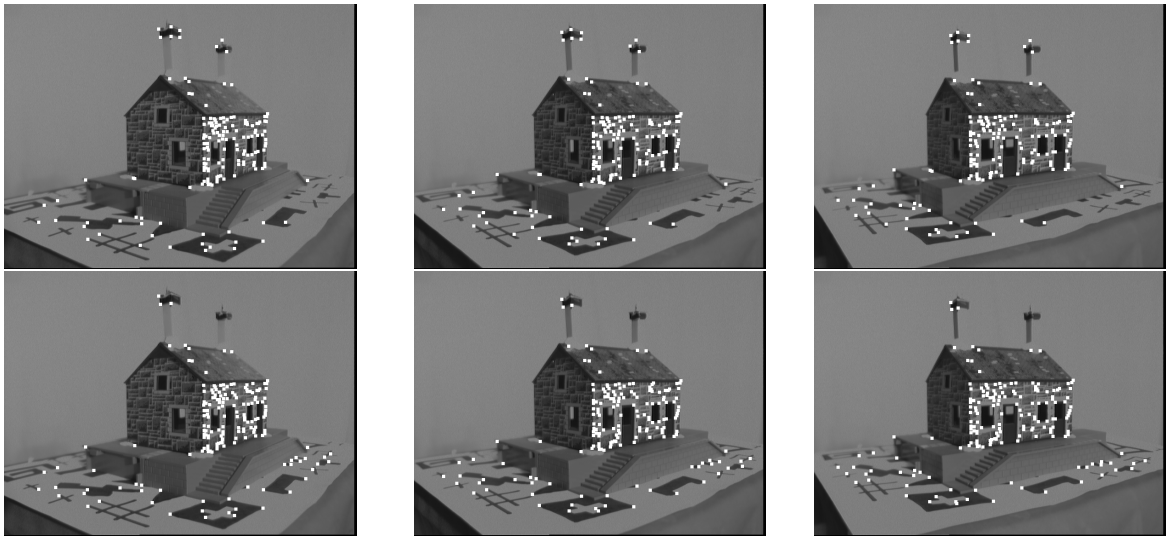

Figure 1: Three images from a sequence of a model house, with corner matches from the 2-image matching scheme (top) and the 3-image scheme (bottom). There is a lateral motion of the camera of about 3-4cm between frames. The image size is 760x550 pixels, and about 400 corners are detected in each image. **Upper 3 images:** In the 2-image scheme, about 300 matches are obtained between each pair, image 1-2 and image 2-3. The r.m.s. perpendicular distance of points from epipolar lines for these matches is about 0.4 pixels. About 160 of these matches survive across all three frames. R.m.s. error between projected 3D points and corresponding image corners is about 0.5 pixels. **Lower 3 images:** In the 3-image scheme, about 180 matches are obtained across all three images. The r.m.s. error between transferred points (using the trifocal tensor) and actual points in the third image is about 1.5 pixels. R.m.s. projected error is again about 0.5 pixels.

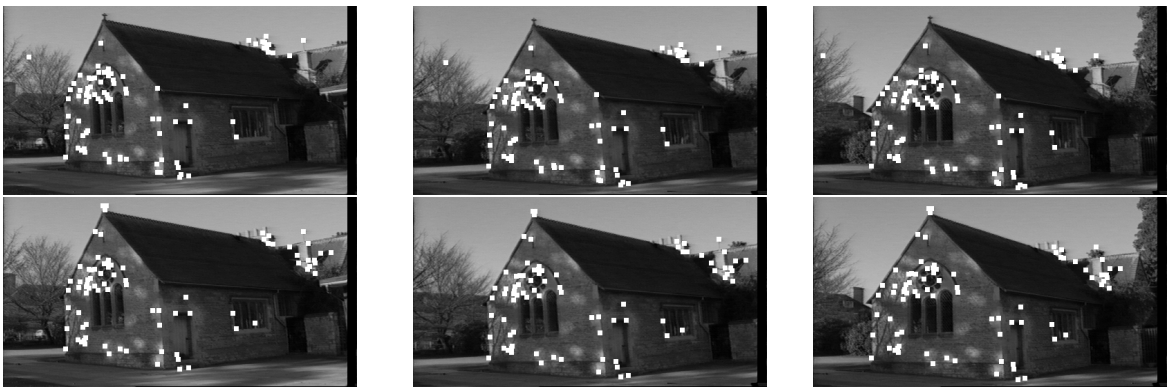

Figure 2: Three images from a sequence of a chapel, acquired by a hand-held camcorder. There is a lateral motion of the camera of a few cm between frames. Image size and corner count as in Figure 1. **Upper 3 images:** In the 2-image scheme, about 150-200 matches are obtained between each pair, with r.m.s. distance of points to epipolar lines about 0.4 pixels. About 80 matches survive across all three frames. **Lower 3 images:** In the 3-image scheme, the number of tokens matched across all three images is again about 80. The r.m.s. error between transferred points (using the trifocal tensor) and actual points in the third image is about 2 pixels. R.m.s. projected error is again about 0.5 pixels.

Of course, the 2-image matching scheme gives rise to some image matches which exist only between image 1-2 and image 2-3. This is because a number of the proposed matches (which are actually mismatches) accidentally agree with the epipolar geometry. Such a mismatch generates a meaningless 3D point, which cannot project to a potential match in the third image, so the corner is not matched across the triplet. Figure 3 shows the full set of matches for images 1-2 of the outdoor scene (a superset of the matches in Figure 2, all consistent with the estimated fundamental matrix), and the mismatches present in this set. Epipolar mismatches of this type occur particularly in an area of texture which gives many corners of similar appearance along the epipolar line (such as trees and bushes in outdoor scenes). Mismatches are most easily detected by examining the 3D structure and identifying those points which lie away from the main groupings of 3D structure.

Another reason for matches existing only between image 1-2 and image 2-3 is that the corner detector does not continue to find the point, or the physical point moves out of view.

Figure 4 shows typical results obtained from the trifocal tensor computation in the 3-image matching scheme. The set of seed point and line matches provided for the trifocal tensor computation contains both correct matches and mismatches. These are distin-

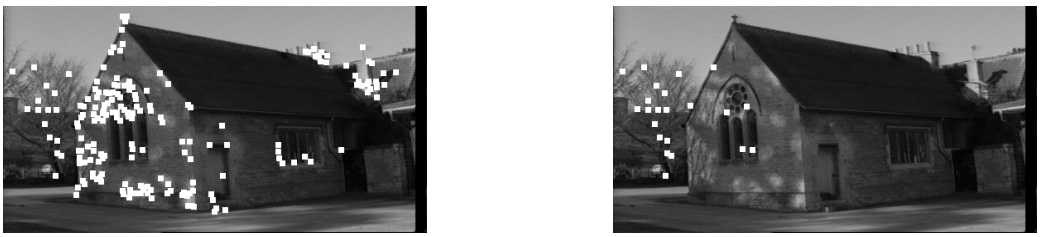

Figure 3: For the sequence in Figure 2 under the 2-image matching scheme, the left image shows the full set of matches obtained between the first two images superimposed on the first image. The right image shows those matches which are in actuality outliers. They are congregated mainly on the ambiguous texture area created by the trees in the background, and are accepted by the 2-image matching because they accidentally agree with the epipolar geometry. Only one of these outliers survives when a third image is processed, however - see Figure 2.

guished by their consistency with the estimated trifocal tensor.

In summary, the experimental results do not suggest that the 3-image matching scheme produces a marked improvement over the 2-image scheme. There are still good reasons for favouring the 3-image approach, however. Firstly, it is computationally much more elegant and efficient to use the trifocal tensor to match over three images and eliminate mismatches, rather than creating 3D structure from the first two images and then projecting it to the third. Secondly, mismatches in the 2-image scheme which accidentally agree with the epipolar geometry are only detected after processing has moved to the third image. By this stage, it is cumbersome to return to the first two images, remove the mismatch and attempt to rematch the points. More importantly, the mismatches might have adversely affected the computation of the fundamental matrix, leading to missed matches; in contrast, the 3-image scheme offers the possibility of detecting suspect pairwise matches immediately by using the trifocal tensor. Finally, the 3-image approach and the trifocal tensor allow the integrated use of points and lines, unlike the 2-image approach where corners drive the processing.

4.5.2 Experiments on track lifetime

Figure 5 shows two comparisons of the 2-image and 3-image schemes, in terms of overall matching statistics along a sequence. Echoing the conclusion of the previous section, the new comparison shows no significant difference in performance between the two ap-

Figure 4: Results for matching a triplet of images using the trifocal tensor, superimposed onto the last image. Top (left to right) **left:** point correspondences over the three images, with small squares indicating the previous positions. **centre:** matches consistent with the computed trifocal tensor, **right:** outlying matches (mismatched points). At bottom, the same information for line matches. In all, there are 101 seed point correspondences over the three images, 76 are indicated as inliers and, 25 as outliers. There are 15 seed line correspondences, 11 are indicated as inlying 4 as outlying.

proaches. However, we re-iterate that the the trifocal tensor based computation has significant advantages, in terms of elegance of use, and its ability to handle points and lines in an integrated way.

5 Robust Computation of the trifocal tensor

In this section we describe the robust computation of the trifocal tensor given a set of putative corner and line correspondences over the three images, as described in the previous section. The trifocal tensor has 27 tensor entries, but only their ratio is significant, leaving 26 that must be specified. Each triplet of point correspondences $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ provides nine *linear* equations on the entries of $\mathtt{T}_{ijk}$, four of which are independent. Each triplet of line correspondences $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ provides two linear equations in the entries of $\mathtt{T}_{ijk}$. Therefore provided that $2n_l + 4n_p \geq 26$ (where n_l is the number of lines, and n_p is the number of points), $\mathtt{T}_{ijk}$ can be determined uniquely (up to scale) using a linear algorithm. This requires a minimum of 7 points or 13 lines or a combination of the two. If more points or lines are included a least squares solution can be employed in the same manner as the linear solution of the fundamental matrix, i.e. a 27×27 moment matrix is formed and the solution obtained using SVD [18]. This is the standard linear method

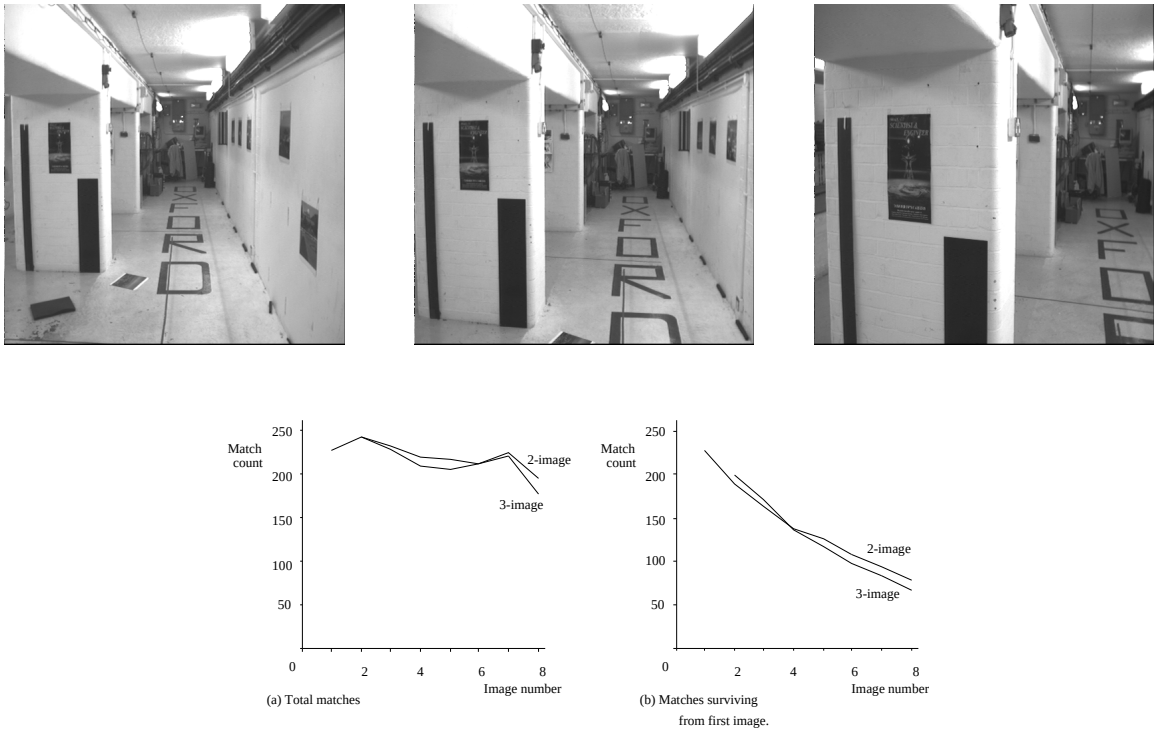

Figure 5: Three images (first, middle, last) from a sequence of ten, taken as the camera moves a total distance of about 2m in an indoor scene. The left graph shows the total number of matches at each stage, as an indicator of overall performance (the drop in numbers at the end of the sequence is because of the large blank area of wall in the scene). The right graph shows the number of matches which have been tracked continuously from the start of the sequence at each stage. There is no significant difference between the 2-image and 3-image schemes.

proposed by Hartley.

In [26] it was shown that random sampling methods, such as least median squares [23] or RANSAC [8] give the best estimate in outlier corrupted data. In these methods it is extremely important that the minimum number of correspondences are used, to reduce the probability of a mismatch being included in the random sample of correspondences used in RANSAC. The trifocal tensor has only 18 independent degrees of freedom, so for 6 points it is possible to determine T_{ijk}, though not uniquely (1 or 3 possible solutions are obtained from the solution of a cubic). The correct solution may be verified by the whole set of point and line correspondences. The number of points and lines are totalled together to determine the feasibility of a given solution. The decision process to render lines inlying or outlying is described below. When the final parameter estimates are recovered they are improved on by iterated least squares using both points and lines.

Choice of estimator. There are two types of robust estimator: RANSAC and least median squares (LMS), in the former, the number of points and lines inlying are totalled together to determine the feasibility of the solution. In the latter, the solution is taken that minimises the median. The former requires *a priori* knowledge of the variance of the residuals, so may not be used in the general case. The latter requires no such knowledge and provides an estimate of this variance, as described below.

The error measure for least median squares will now be discussed in more detail. As there are two different errors, one for points and lines, what is the best way to estimate the median error? When estimating parameters using multiple data representations, thought must be given to a suitable weighting procedure. For the general use of LMS the solution selected minimises the median error, in this specific case we are faced with the problem that there are two sets of error measurements both with different variances. The approach taken is to use a weighted combination of the medians, the weighting requires some *a priori* knowledge of the *relative* variances of the point and line errors. The weighting used is

$$e_m = n_p \frac{e_m^p}{\sigma_p} + n_l \frac{e_m^l}{\sigma_l} \tag{5}$$

where n_p is the number of points, n_l the number of lines and σ_p, σ_l are the standard deviations of the two error terms, which are unknown. Note as the scale of e_m is unimportant only the ratio $\sigma_p : \sigma_l$ need be known. This is estimated off line by running least median squares to estimate the trifocal tensor for several sequences and calculating the ratio of the medians for the best result in each case.

1. Repeat for m samplings as determined in Table 2:

   (a) Select a random sample of the minimum number of six token correspondences to estimate the trifocal tensor T. This provides 1 or 3 solutions.

   (b) For each of these solutions:

      i. Calculate the error e_i^p for the ith point correspondence.

      ii. Calculate the error e_j^l for the jth line correspondence.

      iii. Find the median of each error term: e_m^p for the point correspondences, and e_l^p for the line correspondences.

      iv. Calculate the weighted median for the two data sets: $e_m = n_p \frac{e_m^p}{\sigma_p} + n_l \frac{e_m^l}{\sigma_l}$.

   (c) Select the best solution over all the samples i.e. that with the lowest e_m. In the case of ties select the solution which has the lowest standard deviation of inlying residuals.

2. Re-estimate the parameters using all the data that has been identified as consistent, using iterated least squares.

Table 1: *A brief summary of random sampling algorithm*

The novel contribution here is to demonstrate a method based on six points. It will be seen that the new method gives a more accurate solution than for seven, and that the amount of computation is reduced.

The algorithm is summarised in Table 5

5.1 Implementation details

Using six matches to solve for the trifocal tensor The method for finding the trifocal tensor from six points is inspired by the method of Quan [21] for computing the structure of 6 points from 3 views. It is described in appendix A.

Error measures for points and lines For LMS the best solution is that which minimises the median residual. To determine this median residual two error criteria are defined: one each for lines and points. Given the trifocal constraint and the location of a scene point in two images its location in the third image may be predicted. From empirical tests the best results have been obtained by the minimisation of the distance of a predicted point from its actual location in the image plane. The error criterion that is used is the average of this distance over the three images. A trio of token correspondences is deemed consistent with a given constraint if the error criterion is below a threshold of 1.96σ, where σ is the estimated standard deviation of the error in locating a point. The estimation of the standard deviation of the error is discussed below. For lines, the root mean square of the distances of the end points of the actual line to the predicted line is used as the error criterion.

Robust estimation of the standard deviation of the error term Robust techniques to eliminate outliers are all founded upon some knowledge of the standard deviation σ of the error. Generally, given σ, outliers are calculated as follows:

$$z = \begin{cases} \text{non outlier} & |d| \leq t = 1.96\sigma \\ \text{outlier} & \text{otherwise,} \end{cases} \tag{6}$$

where $t = 1.96\sigma$ is a user defined threshold. In the case of the trifocal tensor there are three errors for each correspondence—distances of the predicted point to the actual point d_1, d_2, d_3 in each image. There are two options: either both all be tested by rule (6) and if either d_1, d_2, d_3, is greater than t then the correspondence is considered outlying; or, the three may be combined for a single test. The latter approach is followed, noting that $d_1^2 + d_2^2 + d_3^2$ is approximated by a χ^2 variable with two degrees of freedom leads to the following 95% confidence test:

$$z_i = \begin{cases} \text{non outlier} & d_1^2 + d_2^2 + d_3^2 \leq t = 5.99\sigma^2 \\ \text{outlier} & \text{otherwise,} \end{cases} \tag{7}$$

The standard deviation is related to the characteristics of the image, the token detector and the matcher. Often the value of σ is unknown, in which case it must be estimated from the data [23]. Following the precedents established for point correspondences, the standard deviation for lines σ_l may be calculated in a similar way.

Features	Fraction of Contaminated Data, ϵ						
p	5%	10 %	20 %	25 %	30 %	40 %	50 %
6	3	4	10	16	24	63	191
7	3	5	13	21	35	106	382

Table 2: *The number m of subsamples required to ensure $\Upsilon \geq 0.95$ for given p and ϵ, where Υ is the probability that all the data points selected in one subsample are non-outliers.*

5.2 Comparison of 6 and 7 point robust schemes

The number of subsamples required is now calculated. Fischler and Bolles [8] and Rousseeuw [23] proposed slightly different means of calculation, but both give broadly similar numbers. Here, the method of calculation given in [23] is used. Ideally every possible subsample would be considered, but this is usually computationally infeasible, and so m the number of samples, is chosen sufficiently high to give a probability Υ in excess of 95% that a good subsample is selected. The expression for this probability Υ is

$$\Upsilon = 1 - (1 - (1 - \epsilon)^p)^m, \tag{8}$$

where ϵ is the fraction of contaminated data, and p the number of tokens in each sample. Table 2 gives some sample values of the number m of subsamples required to ensure $\Upsilon \geq 0.95$ for given p and ϵ. It can be seen that the smaller the data set needed to instantiate a model, the less samples are required for a given level of confidence. If the fraction of data that is contaminated is unknown, as is usual, an educated worst case estimate of the level of contamination must be made in order to determine the number of samples to be taken, this can be updated as larger consistent sets are found e.g. if the worst guess is 50% and a set with 80% inliers is discovered, then ϵ could be reduced from 50% to 20%. It can be seen that as the proportion of outliers increases many more samples need to be taken for the seven point algorithm then for six.

Discussion The six point algorithm gives better results than the seven point algorithm described in [27], when tested on both real data and synthetic data (where the ground truth is known). This is for two reasons, the first being that that the six point algorithm requires less correspondences to estimate and so has less chance of including an outlier, as evinced by Table 2; the second, and perhaps more important, is that the six point algorithm exactly encodes the constraints on the parameters of the trifocal tensor. The

seven point algorithm on the other hand has too many degrees of freedoms, 27, when there should only be 18. This means that the tensor is overparameterised and a least squares solution will usually give a result that violates the constraints on its coefficients leading to a solution that is implicitly incorrect.

The six point algorithm is also considerably faster. In the case of the seven point algorithm the eigenvector of a 27×27 matrix must be found, which is slower than the solution of a cubic. Furthermore far fewer six point samples need to be taken to get a given degree of confidence in the result.

6 Structure Recovery

At the start of a sequence, structure is instantiated using the matches obtained from the first three images, as described in section 6.1. After initialisation, an update process is employed for each new image added to the sequence. Matching between the last image and the new image was discussed in section 4.3. The matches provide a correspondence between the existing 3D structure and the new image tokens. This correspondence enables the 3×4 camera matrix $\mathtt{P}$ for the new image to be determined, using a robust computation eliminating mismatches. Then all existing 3D structure can be projected to the new image, and unmatched 3D points and lines are matched using a search area around the projected position.

Given the matches and camera matrix for the new image, existing estimates of 3D structure can be updated by applying the new observations in an Extended Kalman Filter (EKF). This technology has been used in several systems including [1, 11, 30]. After the 3D structure has been updated, $\mathtt{P}$ is recomputed, using a non-linear computation which minimises the squared distance between projected 3D structure and actual image observations on the image plane. The recomputed $\mathtt{P}$ is used to determine the position of the camera focal point $\mathbf{Q}$ by $\mathtt{P}\mathbf{Q} = 0$.

Finally, new matches which do not yet have associated 3D structure are used to generate new 3D features, thereby handling new parts of the scene which sweep into view.

6.1 Implementation details

Structure initialisation from three views A triplet of images is used to generate camera matrices $\mathtt{P}_1, \mathtt{P}_2, \mathtt{P}_3$ for the three images [18]. Image corners and line segments,

which have been matched over this initial triplet, generate points and line segments in 3-space. Points in 3D are initialised determining the 3D point whose image projection, using the computed camera matrices, lies closest to the measured image corners. Line segment initialisation in 3D is carried out from the lines $\mathrm{l}_1, \mathrm{l}_2$ and l_3 in the three images, where l is a 3-vector containing the homogeneous coordinates of the line. Each line backprojects to a plane in space by

$$\pi_i = \mathtt{P}_i^\top \mathrm{l}_i$$

These planes intersect along the required 3D line, so $\pi_i \cdot \mathbf{X} = 0, i = 1, 2, 3$ is true for all 3D points $\mathbf{X}$ on the line. It follows that the 3×4 matrix $\mathtt{A}$ whose rows are $\pi_i^\top$ has a two-dimensional nullspace, and if the null vectors are $\mathbf{N}_1$ and $\mathbf{N_2}$, the required 3D line has the form $\mathbf{L} = \mathbf{N}_1 + \lambda \mathbf{N}_2$ where λ is a scalar. Allowing for the fact that the data is not perfect and the three planes will not intersect along a single line, we use SVD to find the null vectors. This method fails if the camera motion is parallel to the 3D line in the scene, because the generated planes are parallel.

To define the endpoints of the 3D line, $\mathbf{L}$ is projected to l_{Pi} in each image. In the first image, the perpendicular projections of the endpoints of the line segment on l_{P1} are computed, and these points are backprojected and the rays intersected with $\mathbf{L}$. This is repeated in images 2 and 3, giving in all six intersection points with $\mathbf{L}$. The two extreme intersection points define the endpoints of $\mathbf{L}$.

Obtaining good 3D instantiation Computed structure is poor for 3D points which lie along the direction of camera motion, because the rays backprojected from the image are nearly coincident and hence the estimation of their intersection point is unstable. Similarly (as stated above) for lines which are nearly parallel to the direction of camera motion, because the backprojected planes are nearly coincident. We use the fundamental matrix to eliminate from structure instantiation those corners which lie close to the epipoles, and those line segments which are nearly coincident with the epipoles. Finally, fixing a threshold on the number of times a new token is observed and matched before it is used to initialise 3D structure prevents the use of unreliable tokens.

Numerical conditioning The type of approach here, in which $\mathtt{P}$ is computed from existing 3D structure, effectively means that the 3D world coordinate frame is fixed and does not change over time. We have found, however, that certain computations are more

stable if the world coordinate frame is set so that its origin is at the centre of mass of the currently visible 3D structure. The normalisation described in section 4.4 is effectively a transformation of the world coordinate frame to the form required for greater stability, and this normalisation is employed whenever needed. (It is more straightforward to do this normalisation on the fly for the current 3D data under consideration, than to transform the whole coordinate frame and all accumulated data at each new camera position).

Merging line segments A single line in the physical world may be detected as a set of collinear but unconnected line segments on the image plane. The initialisation method detailed in section 6.1 indicates how, given a set of matched image line segments, a line in 3D corresponding to the maximum extent of any of the image lines can be initialised. Merging of line segments in 3D also provides a method for overcoming the problem of incorrectly segmented lines [30].

6.2 Results

This section contains experimental results for the estimated 3D structure. The structure is "Quasi-Euclidean" (a form which is close to being true Euclidean) which can be obtained given approximate knowledge of the camera intrinsic parameters as described in [3]. If desired, the structure can be transformed to a true Euclidean coordinate frame by employing known relationships between features in the scene, such as perpendicularity.

Figures 6 and 7 show results for the sequence of a model house, and the outdoor scene of a chapel. Point and line structure is shown. The recovered structure is best illustrated by using Delaunay triangulation to obtain image connectivity of the structure, and then mapping image intensity onto the triangular facets in 3D. Lines significantly improve the rendering since they often demarc object planes.

7 Conclusions and further work

A fundamental requirement for automatic structure from motion algorithms is a robust tracker. Tracking must be dependable over long sequences and obtain as many matches as possible on the target object without allowing the entry of mismatches into the match set. We have described a novel matching strategy based on the trifocal tensor which achieves this requirement, using the trifocal tensor both to guide matching and to eliminate mis-

Figure 6: Top-left shows the point and line structure recovered for the model house of figure 1, and the remaining images are obtained by rendering image intensity onto the 3D structure and viewing it from novel viewpoints (viewpoints which were never seen in the original sequence).

Figure 7: Results for the outdoor scene of a chapel of Figure 2. Details are as for the previous figure. The top-right figure shows a plan view of the recovered structure (perpendicular lines at upper-left) and camera positions (arc at lower-right). The camera positions lie at irregular intervals along a rough arc, which is a true reflection of the motion - the sequence was taken with a hand-held camcorder and the motion was not smooth. The dappling effect on the front of the chapel is sunshine through trees.

matches. A further notable and novel feature of the matching scheme is its integrated use of corner and line tokens.

The next stage of the work will extend the types of tokens used. In particular the inclusion of snake-based curve trackers will greatly broaden the range of objects which can be modelled. Finally, an important area to be addressed when working with image sequences is how the accuracy of computation of both $\mathtt{F}$ and T degrades as the baseline is reduced [28], and to develop strategies that deal with this.

A Computation of the trifocal tensor from six point correspondences

Six points are taken at random across the image. Assuming that the six space points are in general position, otherwise the trifocal tensor cannot be uniquely determined, then they can be assigned canonical projective coordinates as follows: $(1,0,0,0)^\top$, $(0,1,0,0)^\top$, $(0,0,1,0)^\top$, $(0,0,0,1)^\top$, $(1,1,1,1)^\top$ and $(X,Y,Z,W)^\top$ where X,Y,Z,W are unknown. Similarly, and without loss of generality the image coordinates of the first four points in each image are assigned to the projective basis in each image, i.e. the coordinates of the six image points are $(1,0,0)^\top$, $(0,1,0)^\top$, $(0,0,1)^\top$, $(1,1,1)^\top$, $(x_5,y_5,w_5)^\top$, and $(x_6,y_6,w_6)^\top$; and $\mathtt{B}^{(i)}$ is the 3×3 projectivity for the canonical frame. It is a simple matter to efficiently calculate $\mathtt{B}^{(i)}$: if $\vec{x}_1$, $\vec{x}_2$, $\vec{x}_3$ and $\vec{x}_4$ are to be transformed to a canonical frame then

$$\mathtt{B}^{(i)} = \begin{bmatrix} \lambda_1\vec{x}_1 & \lambda_2\vec{x}_2 & \lambda_3\vec{x}_3 \end{bmatrix} \tag{9}$$

where

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} = \begin{bmatrix} \vec{x}_1 & \vec{x}_2 & \vec{x}_3 \end{bmatrix}^{-1} \vec{x}_4 \tag{10}$$

Once the canonical system is set up,

$$\begin{bmatrix} 1 & 0 & 0 & 1 & x_5^{(i)} & x_6^{(i)} \\ 0 & 1 & 0 & 1 & y_5^{(i)} & y_6^{(i)} \\ 0 & 0 & 1 & 1 & w_5^{(i)} & w_6^{(i)} \end{bmatrix} = \begin{bmatrix} \alpha^{(i)} & 0 & 0 & \delta^{(i)} \\ 0 & \beta^{(i)} & 0 & \delta^{(i)} \\ 0 & 0 & \gamma^{(i)} & \delta^{(i)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & X \\ 0 & 1 & 0 & 0 & 1 & Y \\ 0 & 0 & 1 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 & 1 & W \end{bmatrix} \tag{11}$$

for each image i. Thus recovery of the trifocal tensor is equivalent to recovering the coordinates of $(X,Y,Z,W)^\top$ and $(\alpha^{(i)},\beta^{(i)},\gamma^{(i)},\delta^{(i)})$ for each camera.

From (11) the values of the sixth space point and camera parameters may be obtained in terms of the fifth and sixth image coordinate as follows:

$$\begin{aligned}
\frac{x_5^{(i)}}{w_5^{(i)}} &= \frac{\alpha^{(i)} + \delta^{(i)}}{\gamma^{(i)} + \delta^{(i)}} \\
\frac{y_5^{(i)}}{w_5^{(i)}} &= \frac{\beta^{(i)} + \delta^{(i)}}{\gamma^{(i)} + \delta^{(i)}} \\
\frac{x_6^{(i)}}{w_6^{(i)}} &= \frac{\alpha^{(i)} X + \delta^{(i)} W}{\gamma^{(i)} Z + \delta^{(i)} W} \\
\frac{y_6^{(i)}}{w_6^{(i)}} &= \frac{\beta^{(i)} Y + \delta^{(i)} W}{\gamma^{(i)} Z + \delta^{(i)} W}
\end{aligned}$$

or

$$\begin{bmatrix} w_5^{(i)} & 0 & -x_5^{(i)} & w_5^{(i)} - x_5^{(i)} \\ 0 & w_5^{(i)} & -y_5^{(i)} & w_5^{(i)} - y_5^{(i)} \\ w_6^{(i)} X & 0 & -x_6^{(i)} Z & w_6^{(i)} W - x_6^{(i)} W \\ 0 & w_6^{(i)} Y & -y_6^{(i)} Z & w_6^{(i)} W - y_6^{(i)} W \end{bmatrix} \begin{pmatrix} \alpha^{(i)} \\ \beta^{(i)} \\ \gamma^{(i)} \\ \delta^{(i)} \end{pmatrix} = 0 \ . \tag{12}$$

The 4×4 matrix on the left is has rank 3 and is given purely in terms of the image coordinates and the sixth space point. As it is of rank 3 the determinant must be zero:

$$\begin{aligned}
(-x_5^{(i)} y_6^{(i)} + x_5^{(i)} z_6^{(i)})(WX - YZ) + (x_6^{(i)} y_5^{(i)} - y_5^{(i)} w_6^{(i)})(WY - YZ) + \\
(-x_6^{(i)} w_5^{(i)} + y_6^{(i)} w_5^{(i)})(WZ - YZ) + (-x_5^{(i)} w_6^{(i)} + y_5^{(i)} w_6^{(i)})(XY - YZ) + \\
(x_5^{(i)} y_6^{(i)} - y_6^{(i)} w_5^{(i)})(XZ - YZ) &= 0 \ .
\end{aligned}$$

This is true in each of the three images:

$$\begin{bmatrix} (-x_5^{(1)} y_6^{(1)} + x_5^{(1)} z_6^{(1)}) & (x_6^{(1)} y_5^{(1)} - y_5^{(1)} w_6^{(1)}) & (-x_6^{(1)} w_5^{(1)} + y_6^{(1)} w_5^{(1)}) & (-x_5^{(1)} w_6^{(1)} + y_5^{(1)} w_6^{(1)}) & (x_5^{(1)} y_6^{(1)} - y_6^{(1)} w_5^{(1)}) \\ (-x_5^{(2)} y_6^{(2)} + x_5^{(2)} z_6^{(2)}) & (x_6^{(2)} y_5^{(2)} - y_5^{(2)} w_6^{(2)}) & (-x_6^{(2)} w_5^{(2)} + y_6^{(2)} w_5^{(2)}) & (-x_5^{(2)} w_6^{(2)} + y_5^{(2)} w_6^{(2)}) & (x_5^{(2)} y_6^{(2)} - y_6^{(2)} w_5^{(2)}) \\ (-x_5^{(3)} y_6^{(3)} + x_5^{(3)} z_6^{(3)}) & (x_6^{(3)} y_5^{(3)} - y_5^{(3)} w_6^{(3)}) & (-x_6^{(3)} w_5^{(3)} + y_6^{(3)} w_5^{(3)}) & (-x_5^{(3)} w_6^{(3)} + y_5^{(3)} w_6^{(3)}) & (x_5^{(3)} y_6^{(3)} - y_6^{(3)} w_5^{(3)}) \end{bmatrix} \begin{pmatrix} WX - YZ \\ WY - YZ \\ WZ - YZ \\ XY - YZ \\ XZ - YZ \end{pmatrix} = 0$$

therefore the vector $\mathbf{t} = (WX - YZ, WY - YZ, WZ - YZ, XY - YZ, XZ - YZ)$ lies in the null space of the matrix on the left, which, if the points are in general position has rank 3. The two dimensional null space of this matrix may be recovered by a singular value decomposition, let $\mathbf{t}_1$ and $\mathbf{t}_2$ be the two vectors spanning this null space. It will now be explained how $\mathbf{t} = (t_1, t_2, t_3, t_4, t_5)$ is recovered from $\mathbf{t}_1$ and $\mathbf{t}_2$. Equation (13) defines

a quadratic in terms of the image coordinates of the fifth and sixth image point:

$$\begin{pmatrix} x_5^{(i)} & y_5^{(i)} & w_5^{(i)} \end{pmatrix} \begin{bmatrix} 0 & t_5 - t_1 & t_1 - t_4 \\ t_2 & 0 & t_4 - t_2 \\ -t_3 & t_3 - t_5 & 0 \end{bmatrix} \begin{pmatrix} x_5^{(i)} \\ y_5^{(i)} \\ w_5^{(i)} \end{pmatrix} = 0 \tag{13}$$

which is satisfied in each of the images. It can be seen that the determinant of the matrix in (13) is zero giving the following constraint on the elements of $\mathbf{t}$,

$$t_1 t_2 t_5 - t_2 t_3 t_5 - t_2 t_4 t_5 = t_1 t_3 t_4 - t_2 t_3 t_4 - t_3 t_4 t_5 \tag{14}$$

from this constraint $\mathbf{t}$ may be recovered from $\mathbf{t}_1$ and $\mathbf{t}_2$ via the solution of a cubic equation, leading to one or three real solutions.

Given $\mathbf{t}$ then $(X, Y, Z, W)^\top$ may be recovered as follows:

$$\begin{aligned} \frac{X}{W} &= \frac{t_4 - t_5}{t_2 - t_3} \\ \frac{Y}{W} &= \frac{t_4}{t_1 - t_3} \\ \frac{Z}{W} &= \frac{t_5}{t_1 - t_2} \end{aligned}$$

assuming that $W \neq 0$, if $W = 0$ then the sixth point is on the plane at infinity it is trivial to use an alternate set of equations to recover $(X, Y, Z, W)^\top$. Given $(X, Y, Z, W)^\top$ the parameters of the camera matrices $(\alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}, \delta^{(i)})$ may be recovered in a linear manner from (12).

From the camera matrices the structure may be initialised directly in the original coordinate system the camera matrices are:

$$\mathtt{P}^{(i)} = \mathtt{B}^{-1} \begin{bmatrix} \alpha^{(i)} & 0 & 0 & \delta^{(i)} \\ 0 & \beta^{(i)} & 0 & \delta^{(i)} \\ 0 & 0 & \gamma^{(i)} & \delta^{(i)} \end{bmatrix} . \tag{15}$$

To recover the trifocal tensor we may use Hartley's [14] equations, if the first camera is set to $\left[\mathtt{I}|\vec{0}\right]$ (effected by a simple transformation of the coordinates) then the trifocal tensor's coefficients are given by Equation (4).

References

[1] N. Ayache. *Artificial vision for mobile robots.* MIT Press, Cambridge, 1991.

[2] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. *Proc. International Conference on Computer Vision*, 1987.

[3] P.A. Beardsley, A.P. Zisserman, and D.W. Murray. Navigation using affine structure and motion. In *Proc. 3rd European Conference on Computer Vision*, pages 85–96. Springer-Verlag, 1994.

[4] J.F. Canny. A computational approach to edge detection. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 8:769–798, 1986.

[5] R. Deriche, Z. Zhang, Q.T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Proc. 3rd European Conference on Computer Vision*, pages 567–576. Springer-Verlag, 1994.

[6] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd European Conference on Computer Vision*, pages 563–578. Springer-Verlag, 1992.

[7] O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, pages 951–956, Cambridge, MA, June 1995.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, vol. 24:381–95, 1981.

[9] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1989.

[10] C.G. Harris. Determination of ego-motion from matched points. In *Third Alvey Vision Conference*, pages 189–192, 1987.

[11] C.G. Harris and J.M. Pike. 3D positional integration from image sequences. In *Third Alvey Vision Conference*, pages 233–236, 1987.

[12] C.G. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[13] R. Hartley. Invariants of points seen in multiple images. GE internal report, to appear in PAMI, GE CRD, Schenectady, NY 12301, USA, 1992.

[14] R. I. Hartley. Projective reconstruction from line correspondences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[15] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proc. 2nd European Conference on Computer Vision*, pages 579–587. Springer-Verlag, 1992.

[16] R.I. Hartley. Cheirality invariants. In *Proc. DARPA Image Understanding Workshop*, pages 745–753, 1993.

[17] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 237–256. Springer-Verlag, 1994.

[18] R.I. Hartley. A linear method for reconstruction from points and lines. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, pages 882–887, Cambridge, MA, June 1995.

[19] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. *Proc. Conference Computer Vision and Pattern Recognition*, pages 543–548, 1993.

[20] J.L. Mundy and A.P. Zisserman. *Geometric invariance in computer vision.* MIT Press, 1992.

[21] L. Quan. Invariants of 6 points from 3 uncalibrated images. In J. O. Eckland, editor, *Proc. 3rd European Conference on Computer Vision*, pages 459–469. Springer-Verlag, 1994.

[22] L. Robert, M.Buffa, and M.Hebert. Weakly-calibrated stereo perception for robot navigation. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, pages 46–51, Cambridge, MA, June 1995.

[23] P.J. Rousseeuw. *Robust Regression and Outlier Detection.* Wiley, New York, 1987.

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorisation method. *International Journal of Computer Vision*, pages 137–154, 1992.

[25] P.H.S. Torr. *Motion segmentation and outlier detection.* PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.

[26] P.H.S. Torr, P.A. Beardsley, and D.W. Murray. Robust vision. In *Proc. British Machine Vision Conference 94*, 1994.

[27] P.H.S. Torr, A. Zisserman, and D.W. Murray. Motion clustering using the trilinear constraint over three views. In R. Mohr and C. Wu, editors, *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 118–125. Springer–Verlag, 1995.

[28] T. Vieville and O.D. Faugeras. Motion analysis with a camera with unknown, and possibly varying intrinsic parameters. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, pages 750–756, Cambridge, MA, June 1995.

[29] X.Hu and N.Ahuja. Matching point features with ordered geometric, rigidity and disparity constraints. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 16(10):1041–1048, 1994.

[30] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis.* Springer-Verlag, 1992.

[31] A. Zisserman, A. Blake, and C.A. Rothwell. Eliciting qualitative structure from image curve deformations. In *Proc. 4th International Conference on Computer Vision*, Los Alamitos, CA, 1993. IEEE Computer Society Press.