Short Papers

Threading Fundamental Matrices

Shai Avidan and Amnon Shashua

Abstract—We present a new function that operates on Fundamental matrices across a sequence of views. The operation, we call "threading", connects two consecutive Fundamental matrices using the trifocal tensor as the connecting thread. The threading operation guarantees that consecutive camera matrices are consistent with a unique 3D model, without ever recovering a 3D model. Applications include recovery of camera ego-motion from a sequence of views, image stabilization (plane stabilization) across a sequence, and multiview image-based rendering.

Index Terms—Structure-form-motion, multiview geometry.

1 Introduction

CONSIDER the problem of recovering the camera trajectory from an extended sequence of images. Since the introduction of multilinear forms across three or more views, there have been several attempts to put together a coherent algebraic framework that would produce a sequence of camera matrices that are consistent with the same 3D (projective) world [14], [2], [13]. The consistency requirement is needed to ensure that all the camera matrices will be defined up to a single projective transformation. There are two basic approaches to the problem. One is to use 3D structure to enforce the consistency constrains, the other is to ensure that all the recovered camera matrices are due to the same reference plane.

The first approach is intuitive and fairly amenable to recursive estimation. An example of such an approach is the incremental method of [2], where fundamental matrices or trifocal tensors are recovered to ensure the quality of matching points and the camera matrices from the 3D structure, which is built-up incrementally, is estimated. Likewise, [13] recovers independently the fundamental matrices of every consecutive pair of images and relies on a 3D structure to put them all in a single measurement matrix which is then used to recover the camera parameters.

The second approach requires some investigation into the connections between camera matrices. If a simple connection exists then one can avoid having to recover the 3D structure of the scene as an intermediate variable in the process. The only attempt we know of is in [14], which seeks a sequence of camera matrices in which the homography matrices all correspond to the plane at infinity. However, the method resorts to a large nonlinear optimization problem, where one, alternatively, recovers 3D structure from motion and motion from structure (thus, not avoiding the 3D structure as an intermediate variable).

In order to obtain a better idea of the issues involved, we will first define what makes a collection of camera matrices consistent with each other. In the Euclidean case (cameras are calibrated), the definition is straightforward. We require that it would be possible to represent the camera parameters of each image in the sequence as a composition of the camera parameters of the previous images in the sequence. This is because a composition of two rotation matrices is a rotation matrix as well. In the projective case, the rotation matrix is generalized to the homography matrix. A homography matrix is a mapping of a 3D planar surface between a pair of images, thus, the rotation matrix is the homography of the plane at infinity. If we want the composition of two projective camera matrices to produce a consistent camera matrix, then we must ensure that multiplying homography matrices will produce a consistent (and valid) homography matrix. This is ensured if all the homography matrices are due to the same 3D plane (i.e., they form a subgroup). Further details on the formal treatment of this issue can be found in the appendix. Our problem statement is:

Problem Def. 1. (Consistent Trajectory). *Given a set of multiple matching points across multiple images, recover a set of camera matrices whose homography matrices are due to a single (arbitrary) reference plane.*

In this paper, we wish to find a consistent set of camera matrices in a *direct* manner, i.e., without reconstructing the 3D scene as an intermediate step. To this end, we introduce a new result on the connection between fundamental matrices and the trifocal tensor on a given triplet of images. Applying this result on a sliding window of triplets of images provides an efficient method for concatenating camera matrices along an extended sequence without resorting to 3D structure. The connection is based on a representation of the tensor as a function of the elements of two consecutive fundamental matrices and a homography matrix of some arbitrary reference plane. An interesting property of this representation is that each camera matrix is guaranteed to be *consistent* with the previously recovered camera matrices. That is, all camera matrices will be recovered due to the same reference plane. By repeated application of the basic result, we call a *threading operation*, on a sliding window of triplets of views, we obtain a consistent sequence of camera matrices (and the fundamental matrix and trifocal tensors as well). Since the threading operation is linear, we have an on-line structure-from-motion algorithm that has only a single nonlinear constraint—the rank-2 constraint of the first fundamental matrix. The rest of the camera matrices are recovered linearly one by one. Potential use of the threading operation includes ego-motion estimation or image stabilization.

The paper is organized as follows: Section 2 provides the general background, notations, and conventions for the paper. The main results are stated and proven in Section 3. The outline of the algorithm is given in Section 4 and results are shown in Section 5.

2 Notations

We use a 3-view building block in the process of concatenating together a sequence of fundamental matrices, hence, our notations are geared for representing three views at a time. A triplet of camera matrices are denoted by $[I;0],[A;v'],[B;v'']$, where the left 3×3 minor is a homography matrix due to the (arbitrary) reference plane and the fourth column is the epipole which is the projection of the center of projection of the first camera onto the second and third image planes, respectively.

We will occasionally use tensorial notations, as described next. We use the covariant-contravariant summation convention: A point is an object whose coordinates are specified with superscripts, i.e., $p^i=(p^1,p^2,\ldots)$. These are called contravariant vectors. An element in the dual space (representing hyper-planes—lines in the 2D plane), is called a covariant vector and is represented by subscripts, i.e., $s_j=(s_1,s_2,\ldots.)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^is_i=p^1s_1+p^2s_2+\ldots+p^ns_n$. This is known as a contraction. An outer-product

• *The authors are with the Institute of Computer Science, The Hebrew University, Jerusalem 91904. E-mail: {avidan, shashua}@cs.huji.ac.il.*

Manuscript received 6 Jan. 1998; revised 15 July 1999; accepted 10 Feb. 2000. Recommended for acceptance by Y.-F. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107614.

of two 1-valence tensors (vectors), $a_i b^j$, is a 2-valence tensor (matrix), c_i^j, whose i,j entries are $a_i b^j$—note that in matrix form $C = ba^\top$. A 3-valence tensor has three indices, say H_i^{jk}. The positioning of the indices reveals the geometric nature of the mapping: for example, $p^i s_j H_i^{jk}$ must be a point because the i, j indices drop out in the contraction process and we are left with a contravariant vector (the index k is a superscript). Thus, H_i^{jk} maps a point in the first coordinate frame and a line in the second coordinate frame into a point in the third coordinate frame. A single contraction, say $p^i H_i^{jk}$, of a 3-valence tensor leaves us with a matrix. Note that when p is $(1,0,0)$, $(0,1,0)$, or $(0,0,1)$ the result is a "slice" of the tensor.

The tensor ϵ_{ijk} is the antisymmetric tensor defined such that $\epsilon_{ijk}a^i b^j c^k$ is the determinant of the 3×3 matrix whose columns are the vectors a, b, c. As such, ϵ_{ijk} contains $0, +1, -1$ where the vanishing entries correspond to arrangement of indices with repetitions (21 such entries), whereas the odd permutations of ijk correspond to -1 entries and the even permutations to $+1$ entries. Therefore, $\epsilon_{ijk}a^i b^j = c_k$ is the cross product of two points resulting in the line c_k. Likewise, $\epsilon^{ijk}a_i b_j = c^k$ represents the point intersection of the to lines a_i and b_j.

Matching image points across three views will be denoted by p, p', p''. When these points appear in a tensor equation indices are included, $p^i, p'j, p''^k$, $i, j, k = 1, 2, 3$, and always i is an index for elements (points or lines) in view one, j is an index for elements in view two, and k is an index for elements in view three. Occasionally, we will refer to a particular component of an image point, and in that case we will adopt the convention $p^i = (x, y, 1), p'^j = (x', y', 1), p''^k = (x'', y'', 1)$.

The trifocal tensor of the three camera matrices $[I; 0]$, $[A; v']$ and $[B; v'']$ is a $3 \times 3 \times 3$ tensor defined below:

$$\mathcal{T}_i^{jk} = v'^j b_i^k - v''^k a_i^j. \tag{1}$$

The tensor acts on a triplet of matching points in the following way:

$$p^i s_j^\mu r_k^\rho \mathcal{T}_i^{jk} = 0, \tag{2}$$

where s_j^μ are any two lines (s_j^1 and s_j^2) intersecting at p', and r_k^ρ are any two lines intersecting p''. Since the free indices are μ, ρ each in the range 1, 2, we have four trilinear equations (unique up to linear combinations). For more details, please refer to [9], [10], [4].

In the sequel, it will become useful to represent the 3×3 fundamental matrix [6], [3] as embedded into a $3 \times 3 \times 3$ tensor as follows [1]:

$$\mathcal{F}_i^{jk} = \epsilon^{ljk} F_{li},$$

where F_{li} is the Fundamental matrix and ϵ^{ljk} is the cross-product tensor. This can be verified as follows:

$$\begin{array}{l} p^i s_j r_k \mathcal{F}_i^{jk} = \\ p^i s_j r_k (\epsilon^{ljk} F_{li}) = \\ p_i \underbrace{(s_j r_k \epsilon^{ljk})}_{p'^l} F_{li} = 0, \end{array} \tag{3}$$

where s, r are any two lines coincident with p'. Finally, a triplet of images will be denoted by (n_1, n_2, n_3), where the numbers n_1, n_2, n_3 stand for the index of the three images in the sequence. Note that order is important. The tensor of the image triplet (n_1, n_2, n_3) is denoted by $< n_1, n_2, n_3 >$.

3 Threading Fundamental Matrices Using Trifocal Tensors

The essence of the paper is in the following two theorems that exhibit the relationship between a homography matrix, fundamental matrix, and trifocal tensor. By repeatedly applying these results on a sliding window of triplets of views, we obtain a camera trajectory which is consistent with a single 3D reconstruction of the world—because all the homography matrices correspond to a single reference plane.

Theorem 1. *The following equation holds:*

$$\mathcal{T}_i^{jk} = c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j, \tag{4}$$

where $\mathcal{T}_i^{jk}$ is the tensor of views 1,2,3, the matrix A, whose elements are a_i^j, is a homography from image 1 to 2 via some arbitrary plane π, $\mathcal{F}_i^{jl}$ is the 2-view tensor of views 1,2, and $C = [C; v''']$ is the camera motion from image 2 to 3 where c_l^k is a homography matrix from image 2 to 3 via the (same) plane π.

Proof. We know that $\mathcal{T}_i^{jk} = v'^j b_i^k - v''^k a_i^j$ where the parameters $[A, v'] = [a_i^j, v'^j]$ and $[B, v''] = [b_i^k, v''^k]$ are the camera matrices from 3D to views 2,3 respectively: $\lambda p' = Ap + \rho v'$ and $p'' \cong Bp + \rho v''$, where p, p', p'' are the matching points in views 1, 2, and 3, respectively, and A, B are homography matrices due to the *same* (arbitrary) reference plane π (uniqueness issue discussed in [9]). By substitution, $p'' \cong BA^{-1}p' + \frac{\rho}{\lambda}(v'' - BA^{-1}v')$ Therefore, the camera motion from view 2 to 3 is represented by, $[C; v'''] = [BA^{-1}; v'' - BA^{-1}v']$ and

$$\begin{array}{rcl} b_i^k & = & c_l^k a_i^l \\ v''^k & = & c_l^k v'^l + v'''^k. \end{array} \tag{5}$$

By substituting the expressions above instead of b_i^k and v''^k in $\mathcal{T}_i^{jk}$, we obtain:

$$\begin{array}{rcl} \mathcal{T}_i^{jk} & = & v'^j (c_l^k a_i^l) - (c_l^k v'^l + v'''^k) a_i^j \\ & = & c_l^k (v'^j a_i^l - v'^l a_i^j) - v'''^k a_i^j \\ & = & c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j, \end{array} \tag{6}$$

where $\mathcal{F}_i^{jl}$ is the trivalent tensor form of the Fundamental matrix, i.e., $\mathcal{F}_i^{jl} = \epsilon^{sjl} F_{si}$ where F_{li} is the Fundamental matrix and ϵ^{sjl} is the cross-product tensor. Finally, because of the group property of projective transformations, since A, B are transformations due to some plane π, then $C = BA^{-1}$. □

Theorem 2. *Given the Fundamental matrix of views 1, 2, and the tensor $\mathcal{T}_i^{jk}$, then the Fundamental matrix between views 2 and 3 can be recovered linearly from six matching points across the three views.*

Proof. The basic tensorial contraction, a trilinearity, is

$$p^i s_j r_k \mathcal{T}_i^{jk} = 0,$$

where s and r are lines coincident with p' and p'', respectively. Thus, the tensor and two views uniquely determine the third view (the reprojection equation) as follows:

$$p^i s_j \mathcal{T}_i^{jk} \cong p''^k,$$

where the choice of the line s is immaterial as long as it is coincident with p'. By substitution, we obtain

$$p^i s_j (c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j) \cong p''^k, \tag{7}$$

which provides two linear equations for the unknowns c_l^k and v'''. We next show that different choices of the line s do not produce new (linearly independent) equations and, thus, six matching points are required for a linear system for the unknowns.

Just as the trifocal tensor $\mathcal{T}_i^{jk}$ satisfies the reprojection equation, so does the 2-view tensor $\mathcal{F}_i^{jl}$:

$$p^i s_j \mathcal{F}_i^{jl} \cong p'^l,$$

where the choice of the orientation of the line s_j is immaterial. Thus, (7) reduces to (in matrix form):

$$p'' \cong Cp' + \rho(s)v''',$$

where $\rho(s)$ is a scalar (depends also on s) that determines the ratio between p'', Cp', and v''', is thus unique (invariant to the choice of s). □

It is worthwhile to note that the homography matrix A that appears in (6) can be generated using the following two observations: First, the space of all homography matrices between two fixed views lives in a 4-dimensional space [11], thus, we can span A from four primitive homography matrices. Second, three of the primitive homography matrices can be generated from the "homography contraction" property of the tensors, i.e., $\delta_k \mathcal{F}_i^{jk}$ is a homography matrix indexed by δ_k, thus, by setting δ_k to be $(1,0,0), (0,1,0)$, and $(0,0,1)$, we obtain three primitive homography matrices. These homography matrices correspond to planes coincident with the center of projection of the second camera (thus, are rank 2 matrices). The fourth primitive homography matrix is a rank-1 matrix whose columns are scaled versions of the epipole v' (which satisfies $F^\top v' = 0$). In particular, choose $v'n^T$ for $n = [1,0,0]^T$. This homography corresponds to a plane coincident with the center of projection of the first camera and (thus, is rank 1), therefore, is not linearly spanned by the three homography contractions of the 2-view tensor. Taken together, any linear combination of the above four matrices will provide an admissible homography matrix A that can be used in (6).

4 Threading and Extended Threading

It is possible to extend the basic 3-frame threading process to handle a sequence of views by using a sliding window of w frames in the following manner:

1. Recover the Fundamental matrix F_1 of the first pair of images in the sequence.
2. Recover the epipole v_1' from the null-space of ${F_1}^T$.
3. Construct the initial homography A_1 as a linear combination of the four homography matrices. For the sake of numerical stability, we wish to find a linear combination that will approximate the form of a rotation matrix. In particular, we use the method described in [8] which is suitable for small-angle rotations.
4. Set $M_0 = [I\ ;\ 0]$ and $M_1 = [A_1\ ;\ v_1']$.
5. Fix w to be the number of images in the sliding window.
6. For image n, perform:
7. For every image h, $n - w < h < n - 1$ in the sliding window compute the relative motion with respect to image $n-1$

$$[A_h\ ;\ v_h'] = M_h \begin{pmatrix} & M_{n-1} & \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1}. \tag{8}$$

8. Let $F_h = [v_h']_\times A_h$.
9. Use all A_h, F_h (with the respective matching points in images $h, n-1, n$) in (7) to compute $\mathbf{C} = [C, v''']$—the relative motion from image $n-1$ to image n.
10. Let

$$M_n = C \begin{pmatrix} & M_{n-1} & \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Normalize M_n by its Frobenius norm.

We use a typical sliding window of $w = 5$ frames for the extended threading. Note also that the recovered motion parameters (the matrix C) are relative to the last frame and in order to bring it into the coordinate system of the entire sequence, we need to multiply it by M_{n-1}. Finally, we normalize each new camera matrix by the Frobenius norm to avoid the entries in the matrix to shrink to zero or explode to infinity.

(a) (b)

Fig. 1. The "Helicopter" experiment consisting of 70 frames. We have subsampled the sequence by taking every other image. (a) and (b) are frames 0 and 35 in the subsampled sequence. The size of the images is 360×240.

5 Experiments

The threading process produces the set of projection matrices relative to some chosen reference plane. As such, we can view the threading process as relevant to both the task of recovering the "ego-motion" of a moving camera (the projection matrices) and to the task of stabilizing a planar surface along a sequence of views. In the latter case, the plane to be stabilized should be present in the first two views of the sequence but *need not be present* later on in the sequence. For example, in the sequence shown in Fig. 4, the plane chosen as a reference plane (say, for purpose of image stabilization) is the desk, which is fully present in the first image of the sequence, but then gradually moves out of the field of view as the image sequence progresses (the physical desk is still present but is cluttered by off-plane objects). The threading procedure creates a collection of camera matrices with respect to the chosen reference plane—which in particular means that we obtain, as a byproduct, the homography matrices induced by that plane even in images in which the planes is no longer visible.

The ego-motion aspect of the threading process is advantageous for situations in which the integrated field of view is much larger than the field of view of the camera at each time instant. In that case, the build-up of projection matrices from image measurements is incremental because new regions of the scene replace old scene parts as the camera moves. The threading process is designed to deal with an incremental integration of image measurements while avoiding the need to carry the 3D coordinates of corresponding points along the way. For example, the sequence of 35 frames in Fig. 1 was taken from a helicopter covering a large

(a) (b)

Fig. 2. Accuracy evaluation of plane stabilization. The ground was manually chosen as the reference plane in the first two frames. (a) and (b) display the overlay of frame 1 and frames 20 and 35, respectively, warped toward the first frame. The ground is aligned to a subpixel level, while the roof (above the ground) exhibits parallax effects. The car is duplicated on the road because it was moving while the sequence was taken.

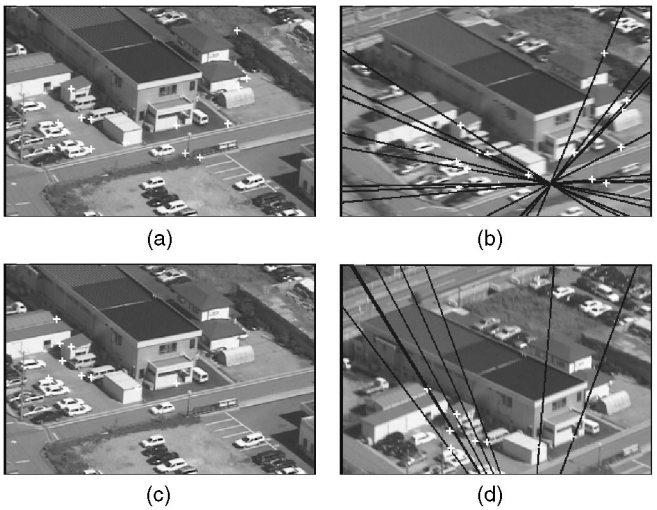

Fig. 3. Accuracy evaluation of recovered fundamental matrices from the threading process. (a) and (b) are images 1 and 20, respectively, with a number of tracked points superimposed. The distance to the epipolar lines is about 1.5 pixels. (c) and (d) are images 1 and 34, respectively, with a number of tracked points superimposed. The distance to the epipolar lines remains about 1.5 pixels.

integrated field of view. The recovery of ego-motion (fundamental matrices, trifocal tensors, and the camera projection matrices) is necessarily incremental.

Next, we describe the experimental setup and results on the two example sequences discussed above. In each example, we start with an image sequence as input and from which we recover a set of point matches. The point correspondence were extracted automatically using the KLT package [12]. Typically, we obtain around 150 matching points between frames. We replace lost feature points with new ones to maintain a constant number of about 150 point matches. The algorithm performs the threading operation in a robust statistics framework. Following the LMeDS (Least of Median Squares) approach [7], we sample groups of six points and recover the motion parameters from them. We then choose the group that had the largest support and recompute the motion parameters again in a least-squares manner.

Fig. 4 shows a number of frames from a sequence of 40 frames. The sequence was taken with a hand-held cam-corder that produced images of size 360×240. The reference plane was manually selected to correspond to the table in the scene. The associated homography matrix computed from views 1 and 2 of the sequence was fed into the threading process which then progressed through the remaining images of the sequence. Note that the table gradually moves out of the field of view. The threading process produces as a byproduct, the homography matrices between adjacent views along the sequence, which all should correspond to the selected plane arising from the table in the scene. Fig. 5 shows the warped images between pairs of images

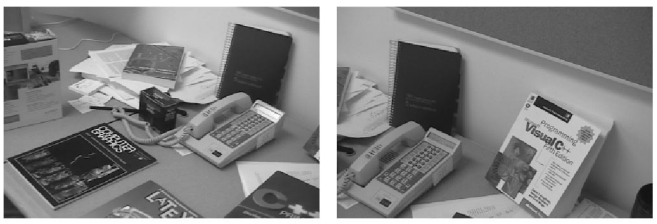

Fig. 4. The "Desk" experiment consists of 80 frames. We have subsampled the sequence by taking every other frame. (a) and (b) show frames 0 and 40 from the subsampled sequence. The size of the images is 360×240.

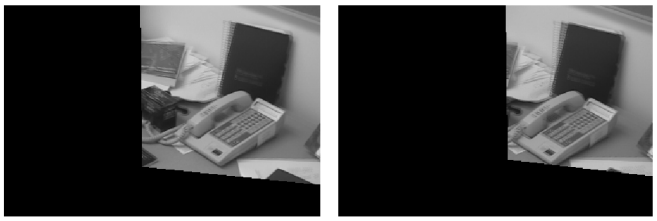

Fig. 5. Accuracy evaluation of plane stabilization. The desk was manually chosen as the reference plane in the first two frames. (a) and (b) display the overlay of frame 1 and frames 25 and 40, respectively, warped toward the first frame. The desk is aligned to a subpixel level while objects on the desk exhibit displacements porportional to their parallax.

cascaded through from the last to the first images—note that the plane is stabilized in the warped sequence, while the objects outside the table plane suffer from parallax effects. The plane stabilization application (and accuracy of) is demonstrated also in the "helicopter" sequence (Fig. 1). This sequence consists of 35 frames of size 360×240. Here, the ground is selected as the reference plane and its homography matrix is fed into the threading process which then progressed through the remaining 33 images. Fig. 2 shows the stabilized images. Note that the roof suffers from parallax effects. Also, the car on the road is duplicated because it was actually moving while the sequence was taken.

The ego-motion aspect of the threading process is demonstrated in Fig. 3 using the helicopter sequence. The threading process provides the camera matrices, fundamental matrices, and the trifocal tensors. For example, the quality of the fundamental matrix can be measured by the proximity of the epipolar lines to respective matching points. The extended threading algorithm was applied with a sliding window of five images. For the demonstration, we computed the fundamental matrices between the first image and several intermediate frames, directly from the recovered camera matrices, and used points that the KLT algorithm tracked throughout the sequence. One can observe, that the epipolar lines pass through the matching points. The median distance of points to their corresponding epipolar line reaches a peak of about 1.5 pixels at frame 15 and remains similar from there until the end of the

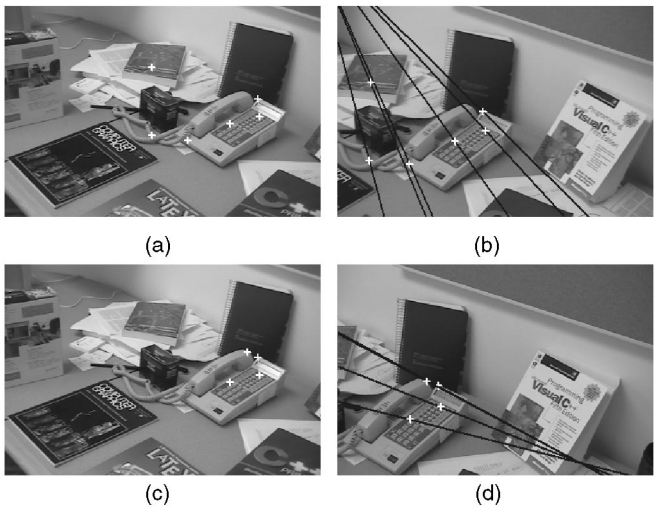

Fig. 6. Accuracy evaluation of recovered fundamental matrices from the threading process. (a) and (b) are images 1 and 20, respectively, with a number of tracked points superimposed. The distance to the epipolar lines is about 0.7 pixels. (c) and (d) are images 1 and 35, respectively, with a number of tracked points superimposed. The distance to the epipolar lines is about 1.4 pixels.

sequence. This accuracy provides an indication to the error accumulation of the threading process over the integration of many views (in this case 35 views). A similar example, for the desk sequence, is given in Fig. 6. In this case, the error is less than one pixel for the first 30 frames and only then does it start to rise up to about 1.4 pixels in frame 40.

6 Conclusion

We have introduced an efficient method for concatenating fundamental matrices along a sequence of images using the trifocal tensor as an intermediate "glue"—hence, the term "threading." The process is captured by a simple equation (4) which relies on an already computed fundamental matrix and a choice of gauge (coordinate frame) captured by a homography matrix of a reference plane. The threading equation then uses matching points with the next view in the sequence (at least six matching points) to produce the basic elements for constructing projection matrices, fundamental matrices, and trifocal tensors. Thus, the projection matrices constructed are guaranteed to comply with the chosen gauge, i.e., the homography matrices, thus computed, all correspond to a single reference plane—we have referred to that property as a establishing a "consistent" camera trajectory (see Appendix). We have also shown how to extend the basic scheme to handle larger sequence segments (beyond three images) for better numerical stability.

We have highlighted two aspects of the threading process. First, is the image stabilization aspect arising from the byproduct of recovering homography matrices arising from a single plane. For example, Figs. 5 and 2 demonstrate the stabilization of a selected plane across a sequence in which the plane gradually moves out of the field of view. Recent approches for plane stabilization across sequences [15] attempt also to perform under these conditions, but here, the stabilization is established in the context of tracking 3D points rather than coplanar points. In other words, the stabilization process uses all the information in the scene (captured by matching points) and, thereby, need not deal with the issue of maintaining a continues segmentation of the chosen plane from the scene throughout the sequence. The second aspect that was highlighted was the recovery of projection matrices.

Appendix

On Consistent Camera Trajectory

A set of projective camera matrices is consistent if all its homography matrices are due to the same (arbitrary) reference plane. Here, we show why this is true.

Let $M_j = [A_j, v_j]$, $j = 0, ..., J$, be a set of J camera matrices and let P_k, $k = 1, ..., K$ be a set of 3D points represented under some projective frame. The J images of the K points are described by $p_{kj} \cong M_j P_k$. Without loss of generality and to save on adding further notations, we will continue using the same symbols, i.e., $M_0 = [I; 0]$ and $M_j = [A_j, v_j]$. The matrices A_j are not unique as they belong to a 4-parameter family as follows: Consider three arbitrary scalars, a, b, c, arranged in a vector n and a fourth scalar μ. The coordinate transform represented by

$$W = \begin{bmatrix} I & 0 \\ n^\top & \mu \end{bmatrix}. \tag{9}$$

clearly satisfies $[I; 0]W = [I; 0]$, thus, the choice of canonical representation is not affected by the change of coordinates induced by W. Also, $W^{-1}P_k$ changes only the fourth coordinate λ_k which becomes $\frac{1}{\mu}(\lambda_k - n^\top p_{ko})$. Likewise, $[A_j; v_j]W = [\mu A_j + v_j n^\top; \mu v_j]$. Therefore, for every choice of n, μ, we have a corresponding coordinate frame that consists of $\mu A_j + v_j n^\top$ instead of A_j and $\frac{1}{\mu}(\lambda_k - n^\top p_{io})$ instead of λ_k.

The family of matrices $\mu A_j + v_j n^\top$ are 2D collineations (homographies) mapping the first view onto view j through a $2D$ plane, a reference plane, whose position in space is represented by n, μ. For example, if P is some point on the plane corresponding to $n = 0$ and p, p' are its projections onto views $0, j$, then $A_j p \cong p'$ and vice-versa.

Since n, μ are shared among all the camera matrices, then a necessary and sufficient condition for a set of camera matrices $[I; 0], [A_j, v_j]$, $j = 1, ..., J$, to be *consistent* with a single 3D scene reconstruction is that the homography matrices A_j form a subgroup, i.e., they all correspond to the same reference plane.

References

[1] S. Avidan and A. Shashua, "Tensor Embedding of the Fundamental Matrix," *3D Structure from Multiple Images of Large Scale Environments: European Workshop SMILE '98,* R. Koch and L. VanGool, eds., June 1998.

[2] P.A. Beardsley, A. Zisserman, and D.W. Murray, "Sequential Updating of Projective and Affine Structure from Motion," *Int'l J. Computer Vision,* vol. 23, no. 3, pp. 235–260, 1997.

[3] O.D. Faugeras, "Stratification of Three-Dimensional Vision: Projective, Affine and Metric Representations," *J. the Optical Soc. Amer.,* vol. 12, no. 3, pp. 465–484, 1995.

[4] R.I. Hartley and A. Zisserman, *Multiple View Geometry.* Cambridge Univ. Press, 2000.

[5] M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion using Image Stabilization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 454–460, June 1994.

[6] H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature,* vol. 293, pp. 133–135, 1981.

[7] P. Meer, D. Mintz, D. Kim, and A. Rosenfeld, "Robust Regression Methods for Computer Vision: A Review," *Int'l J. Computer Vision,* vol. 6, no. 1, pp. 59–70, 1991.

[8] B. Rousso, S. Avidan, A. Shashua, and S. Peleg, "Robust Recovery of Camera Rotation from Three Frames," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1996.

[9] A. Shashua, "Algebraic Functions for Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 779–789, 1995.

[10] A. Shashua, "Trilinear Tensor: The Fundamental Construct of Multiple-View Geometry and its Applications," *Algebraic Frames For The Perception Action Cycle,* G. Sommer and J.J. Koenderink, eds., Sept. 1997.

[11] A. Shashua and S. Avidan, "The Rank4 Constraint in Multiple View Geometry," *Proc. European Conf. Computer Vision,* Apr. 1996.

[12] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 593–600, June 1994.

[13] P. Sturm and B. Triggs, "A Factorization Based Algorithm for Multiimage Projective Structure and Motion," *Proc. European Conf. Computer Vision,* 1996.

[14] T. Vieville, O. Faugeras, and Q.T. Loung, "Motion of Points and Lines in the Uncalibrated Case," *Int'l J. Computer Vision,* vol. 17, no. 1, pp. 7–42, 1996.

[15] L. Zelnik-Manor and M. Irani, "Multi-Frame Alignment of Planes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 1999.