Threading Fundamental Matrices

Shai Avidan Amnon Shashua
Institute of Computer Science,
The Hebrew University,
Jerusalem 91904, Israel
e-mail: {avidan,shashua}@cs.huji.ac.il

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 Trilinear 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 multi-view image-based rendering.

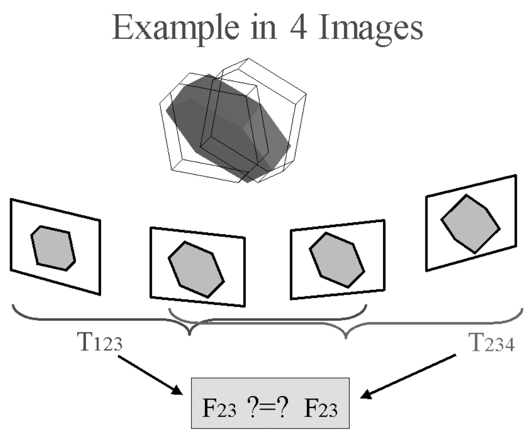
1 Introduction

Consider the problem of recovering the (uncalibrated) camera trajectory from an extended sequence of images. Since the introduction of multi-linear forms across three or more views (see Appendix) 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 [25, 4, 23]. The consistency requirement arises from the simple fact that from an algebraic standpoint a camera trajectory must be *concatenated* from pairs or triplet of images. Therefore, a sequence of independently computed Fundamental matrices or Trilinear tensors, maybe optimally consistent with the image data, but not necessarily consistent with a unique camera trajectory (see Figure 1). There are two basic approaches to the problem:

1. Recover (incrementally or batch-wise) the most (statistically) optimal 3D structure from the image measurements across the extended sequence. Then, given the 3D and 2D correspondences recover the corresponding camera matrix.

2. Recover a sequence of camera matrices whose homography matrices all correspond to the same reference plane.

The first approach is intuitive and fairly amenable to recursive estimation. Example of recent implementations of this approach for uncalibrated camera include the incremental method of [4] who recover Fundamental matrices or Trilinear tensors to ensure the quality of matching points, and then estimate the camera matrices from the 3D structure which is built-up incrementally. Likewise, [23] 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 that is used to recover the camera parameters.



(a)

Figure 1: One can compute two tensors T_{123}, T_{234} from the four images of the 3D scene. However, each tensor can give rise to a different reconstruction of the 3D structure due to noise or errors in measurments, and therefor the camera trajectory between images 2 and 3, as captured by the fundamental matrix F_{23}, is inconsistent between the two tensors. The "threading" operator described in the text guarantees a consistent recovery of the camera trajectory.

The second approach is more challenging since it requires a deeper investigation into the connections between camera matrices. If a simple connection exists then there is the advantage of avoiding 3D structure, as an intermediate variable in the process. The only attempt we know of is of [25] who 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 non-linear optimization problem, where one alternatively recovers 3D structure from motion and motion from structure (thus not avoiding the 3D structure as an

[0] This manuscript, Hebrew University Computer Science CS98-1, has been submitted for publication to IEEE PAMI on Jan. 1998.

intermediate variable).

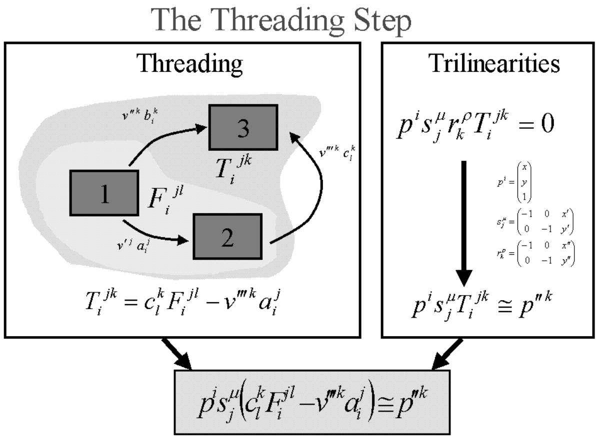
In this paper we introduce a new result on the connection between Fundamental matrices and Trilinear tensors. As a byproduct, this result provide a principaled method for concatenating camera matrices along an extended sequence without resorting to 3D structure. The connection is based on a representation of the tensor as a function of the elements of two consecutive Fundamental matrices and a homography matrix of some arbitrary reference plane (Eqn. 1). An interesting byproduct of this representation is that we are guaranteed to recover (linearly) two *consistent* camera matrices. 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 Trilinear tensors as well). The immediate byproducts (applications) of the threading operation include:

- **Ego-Motion**
  The algorithm recovers a consistent camera trajectory along the image sequence without recovering 3D structure.
- **Image stabilization**
  The algorithm recovers a sequence of camera matrices that are due to the same plane. By selecting a reference plane in the first pair of images, we ensure that the same plane is stabilized throughout the sequence.
- **Multi-view Image-Based Rendering**
  The algorithm puts all the images in a single projective coordinate framework and therefor all the images can contribute to the synthesis of a novel image, using a technique such as [3].

The paper is organized as follows. Section 2 provides the general notations and conventions used in 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. The Appendix contains a brief overview of the necessary elements assumed including the Fundamental matrix, Plane + Parallax representation, the Trilinear tensor, and the tensorial form of the Fundamental matrix.

2 Notations

A point $\boldsymbol{x}$ in the 3D projective space $\mathcal{P}^3$ is projected onto the point p in the 2D projective space $\mathcal{P}^2$ by a 3×4 camera projection matrix $\mathbf{A} = [A, v']$ that satisfies $p \cong \mathbf{A}\boldsymbol{x}$, where $\cong$ represents equality up to scale. The left 3×3 minor of $\boldsymbol{A}$, denoted by A, stands for a 2D projective transformation of some arbitrary plane (the reference plane) and the fourth column of $\boldsymbol{A}$, denoted by v', stands for the epipole (the projection of the center of camera 1 on the image plane of camera 2). In a calibrated setting the 2D projective transformation is the rotational component of camera motion (the reference plane is at infinity) and the epipole is the translational component of camera



(a)

Figure 2: The threading step is plugged into the trilinearities to obtain the threading equation.

motion. Since only relative camera positioning can be recovered from image measurements, the camera matrix of the first camera position in a sequence of positions can be represented by $[I; 0]$.

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., $p^i = (p^1, p^2, ...)$. These are called contravariant vectors. An element in the dual space (representing hyper-planes — lines in $\mathcal{P}^2$), is called a covariant vector and is represented by subscripts, i.e., $s_j = (s_1, s_2, ...)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^i s_i = p^1 s_1 + p^2 s_2 + ... + p^n s_n$. This is known as a contraction. An outer-product 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 $C = ba^\top$. Further details on the necessary background can be found in the Appendix.

3 Threading Fundamental Matrices Using Trilinear Tensors

Given an extended sequence of images we wish to recover a unique camera trajectory which is the most consistent with the image measurements. We also wish to do so in a principaled manner, i.e., see first what can be done at the algebraic level, then figure out what is the best statistical model for incorporating image noise. On the algebraic level, a necessary condition for trajectory consistency is that the recovered camera matrices all refer to the same common reference plane (which could be virtual), see Appendix. The two theorems below are the essence of this paper and include:

- Providing an equation for representing the Trilinear tensor as a function of the Fundamental matrix (represented in its trivalent tensorial form), the reference plane homography between views 1 and 2, and the camera motion between images 2 and 3.

- Given the equation discussed above, the Fundamental matrix between views 1 and 2, and at least 6 matching points across images 1,2,3, one can linearly recover the camera motion between views 2 and 3.
- The recovered camera motion between views 2 and 3, is guaranteed to be consistent (i.e., the corresponding homography matrix is associated with the same reference plane).

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 (see Figure 2).

Theorem 1 *The following equation holds:*

$$\mathcal{T}_i^{jk} = c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j \tag{1}$$

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 $\mathbf{C} = [C; v''']$ 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:

$$\begin{aligned} \lambda p' &= Ap + \rho v' \\ p'' &\cong Bp + \rho v'' \end{aligned}$$

where p, p', p'' are the matching points in views 1,2,3 respectively, and A, B are homography matrices due to the *same* (arbitrary) reference plane π (uniqueness issue discussed in [14]). Clearly,

$$p'' \cong BA^{-1}p' + \frac{\rho}{\lambda}(v'' - BA^{-1}v')$$

Therefore, the camera motion from view 2 to 3 is represented by,

$$[C; v'''] = [BA^{-1}; v'' - BA^{-1}v']$$

and,

$$\begin{aligned} b_i^k &= c_l^k a_i^l \\ v''^k &= c_l^k v'^l + v'''^k. \end{aligned} \tag{2}$$

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

$$\begin{aligned} \mathcal{T}_i^{jk} &= v'^j(c_l^k a_i^l) - (c_l^k v'^l + v'''^k)a_i^j \\ &= c_l^k(v'^j a_i^l - v'^l a_i^j) - v'''^k a_i^j \\ &= c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j, \end{aligned} \tag{3}$$

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^{ljk} is the cross-product tensor (see Appendix). Finally, because of the group property of projective transformations, since A, B are transformations due to some plane π, then so is $C = BA^{-1}$. $\square$

Theorem 2 *Given the Fundamental matrix of views 1,2 and the tensor $\mathcal{T}_i^{jk}$, then the Fundamental matrix between views 2,3 can be recovered linearly from 6 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 (see Appendix). 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,

$$p^i s_j (c_l^k \mathcal{F}_i^{jl} - v'''^k a_i^j) \cong p''^k \tag{4}$$

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 6 matching points are required for a linear system for the unknowns.

Just as the Trilinear 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 (see Appendix). Thus, Eqn. 4 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'' and Cp' and v''', thus is unique (invariant to the choice of s). $\square$

It is worthwhile to note that the homography matrix A that appears in Eqn. 3 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 [17], thus we can span A from 4 primitive homography matrices. Second, three of the primitive homography matrices can be generated from the "homography contraction" property of the tensors (see Appendix), 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, that in matrix form are:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} F \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} F$$
$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} F \tag{5}$$

where F is the Fundamental matrix. 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:

$$\begin{bmatrix} v'_1 & 0 & 0 \\ v'_2 & 0 & 0 \\ v'_3 & 0 & 0 \end{bmatrix} \tag{6}$$

where v' is the epipole satisfying $F^\top v' = 0$. This homography corresponds to a plane coincident with the center of projection of the first camera (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 Eqn. 3.

4 The Online algorithm

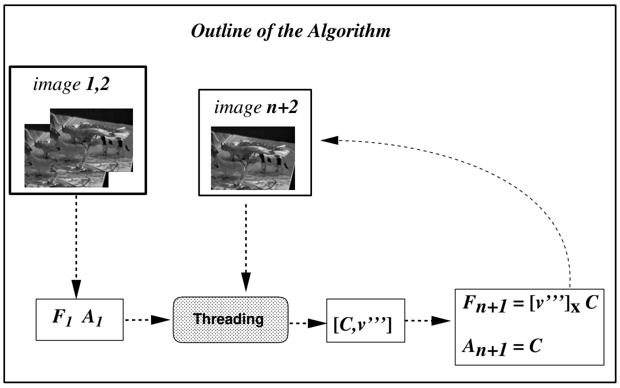
The online algorithm threads together the Fundamental matrices of consecutive images, by applying the threading operation on a sliding window of triplets of images. The algorithm starts with computing the Fundamental matrix of the first pair of images and recovering an initial homography matrix. The initial homography matrix can be recovered either from the primitive homography matrices constructed from the Fundamental matrix, or by using any method for the recovery of a homography matrix by plane stabilization [11]. The rest of the images are added one by one by applying the threading operation on a sliding window of triplets of images. Figure 3 gives a block diagram of the proposed algorithm.

In detail, the algorithm is as follows:

1. Recover the Fundamental matrix F_1 of the first pair of images in the sequence.

2. Recover the epipole v' from the null-space of ${F_1}^T$.

3. Construct the initial homography A_1 as a linear combination of the four homography matrices in Eqns. 5 and 6. 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 [13] which is suitable for small-angle rotations.



(a)

Figure 3: The algorithm starts with recovering the Fundamental matrix F_1 of the first pair of images in the sequence. The Fundamental matrix is then used to construct the initial homography matrix A_1. For image $n+2$, the current Fundamental matrix F_n and homography A_n are used to recover the camera parameters of the new image - $[C, v''']$, using the threading operation. This parameters are then used to construct $F_{n+1} = [v''']_\times C$ and $A_{n+1} = C$.

For image $n+2$:

1. Apply the threading operation for recovering the camera matrix $\mathbf{C} = [C, v''']$. The input to the operation is 6 point matches across three images, the Fundamental matrix F_n and homography matrix A_n, using Equation 4.

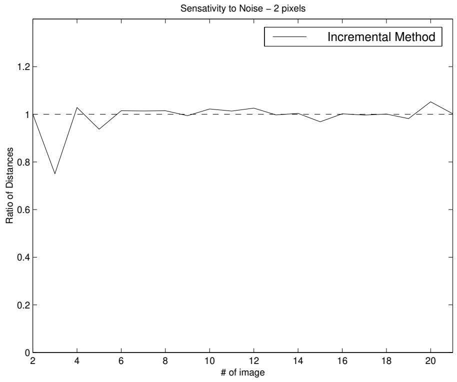
2. The new Fundamental matrix $F_{n+1} = [v''']_\times C$ and homography $A_{n+1} = C$ are the parameters for the recovery of the camera matrix of the next image.

5 Experiments

Experiments were conducted on synthetic data and several different real images with different cameras and different motion parameters. No information about camera internal parameters or motion is known or used. The point correspondence, for the real images, were extracted automatically by our system. In a nutshell, the system computes a bi-directional optical flow and searches for points with high gradient that have a matching optical flow in both direction. Typically we obtain around 200 matching points.

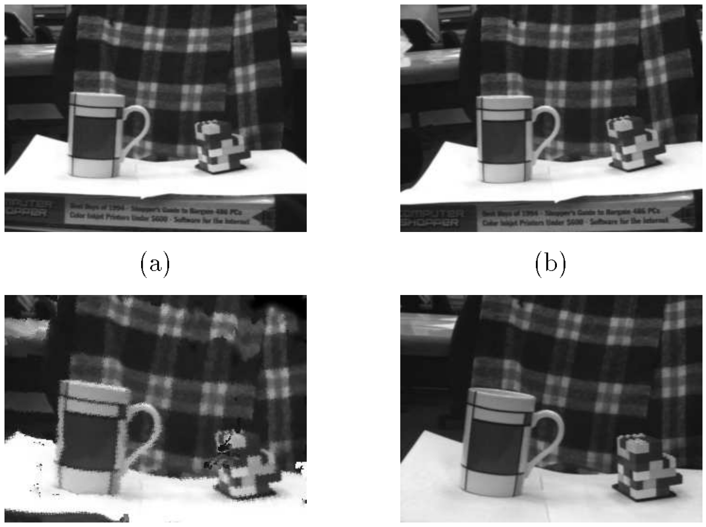
5.1 Test on Synthetic Data

We measured the error of the threading operation along an image sequence. The 3D world consisted of a set of 50 points that were projected on a sequence of 21 images, using randomly generated camera matrices. All image measurements were normalized to the range [0..1] and white noise (of up to 2 pixels in a 512×512 pixels image) was added. We recovered the Fundamental matrix of the first pair of images, using the 8-point algorithm,



(a)

Figure 4: The quality of the recovered camera matrices on a sequence of 21 synthetic images of size 512×512 with added white noise of 2 pixels. The graph shows the ratio between the error of the threading operation and the error of the Fundamental matrix of every consecutive pair of images. The error term is defined as the distance, in parameter space, between the recovered epipole and the ground truth epipole. A value smaller than 1 means the threading operation performed better than the Fundamental matrix. Note that the threading operation does not accumulate errors and remains close to the error rate obtained when using the Fundamental matrix.



(a) (b) (c) (d)

Figure 5: Accumulating the camera matrices along a sequence of 6 images to construct the tensor of images $< 1, 2, 6 >$. This tensor is then used to reproject image number 6 from images 1 and 2. (a),(b) shows images 1 and 2, respectively. (c) shows the reprojected image 6. (d) is the original image 6, shown here for comparison.

and used the fourth homography matrix in Equation 5 as the initial homography matrix. The rest of the camera matrices were recovered according to the algorithm described in Section 4 and we denote the recovered epipoles by e_T. For comparison, we computed the Fundamental matrix from every consecutive pair of images and recovered the epipole from it and denote it by e_F. Figure 4 shows the ratio $\frac{d(e_T)}{d(e_F)}$, where $d(\cdot)$ measures the distance, in parameter space, between the recovered epipole and correct epipole. The test was repeated for 30 times and the median of the errors, for white noise of 2 pixels, is shown. As can be seen, the error rate of the threading operation is almost identical to that achieved by the Fundamental matrix, and does not degrade with the number of images.

5.2 Real Data

Tests on three real sequences were conducted. The first test measured the quality of the recovered parameters on real data. The second test presents a possible use of our method for the purpose of multi-camera image-based rendering mechanism and the third test demonstrates the ability to stabilize a plane.

5.2.1 Test 1

A sequence of 6 images of size 320×240 pixels was used with the camera moving mainly forward. The Fundamental matrix of the first two images was computed after [8] and we used the method described in [13, 2] to recover the initial homography matrix. The rest of the images were added one by one, as described in Section 4. To measure the quality of the recovered parameters we have constructed the Trilinear tensor $< 1, 2, 6 >$, from the recovered camera matrices, and used it to reproject image 6 from images 1 and 2, using the technique described in [3]. The result is shown in Figure 5

5.2.2 Test 2

A sequence of 28 images of size 320×240 pixels was used with the camera moving in a semi-circle motion forward and to the right. As before, the initial homography matrix was recovered with the method described in [13, 2]. The camera matrices of images 2 through 28 were recovered with the threading operation and used to construct two tensors - $< 1, 2, 28 >, < 26, 27, 28 >$. The image pairs $(1, 2)$ and $(26, 27)$, together with their respective tensors, were used to reproject image 28. The results are shown in Figure 6.

5.2.3 Test 3

A sequence of 7 images of size 384×288 pixels was used, with the camera moving mainly to the left. The images contain a collection of toy animals placed on a table covered with a picture of a fruit salad. We manually selected the plane of the table as our initial homography matrix and applied the threading operation to recover the camera matrices. From Theorem 1, the homography matrix

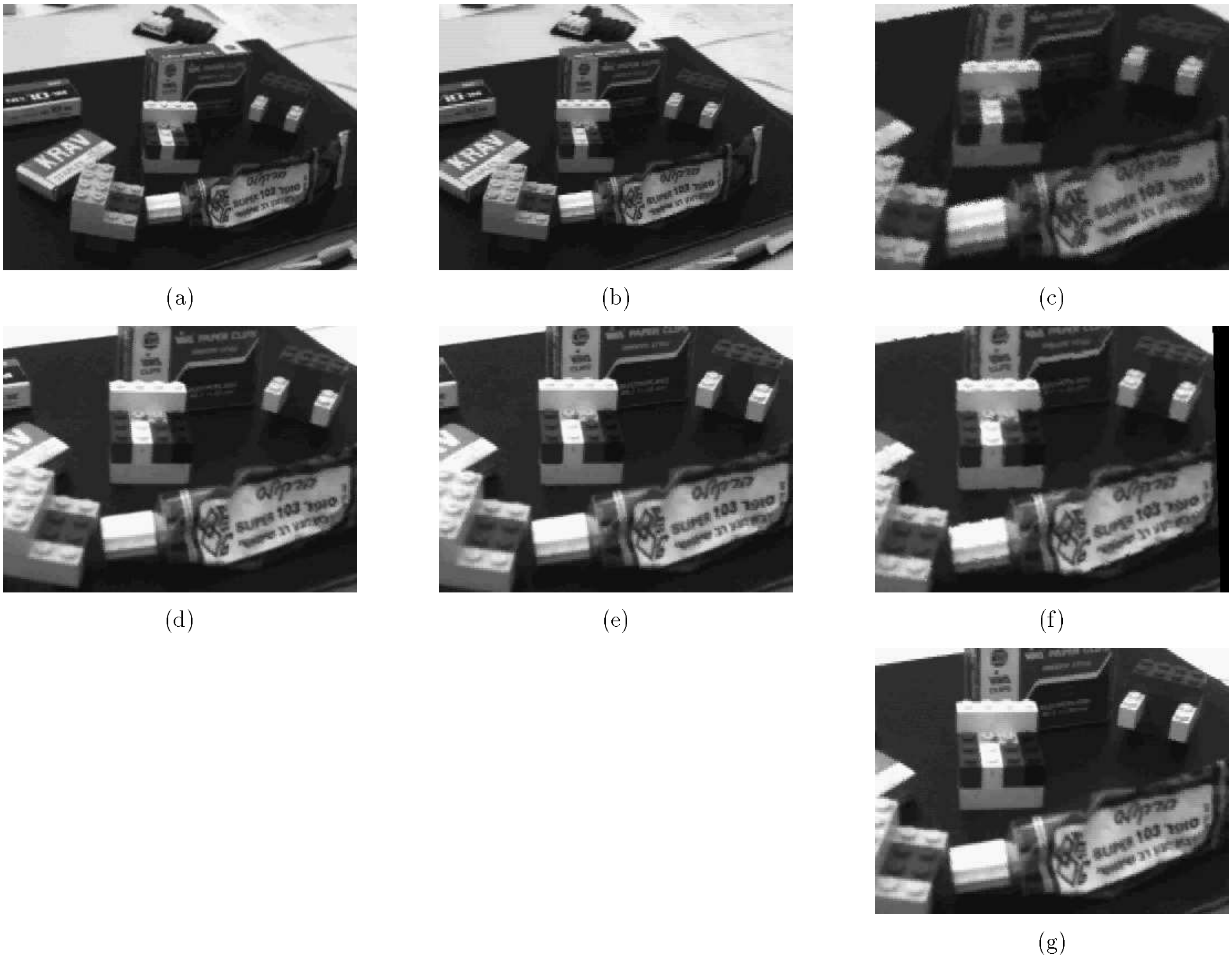


Figure 6: Accumulating the camera matrices along a sequence of 28 images to construct the tensors $< 1, 2, 28 >, < 26, 27, 28 >$. The two tensors are then used to reproject image 28. (a),(b) show images 1 and 2, respectively and (d),(e) show images 26 and 27, respectively. (c),(f) show how image 28 was reprojected from the two pairs, respectively, using the reconstructed tensors. (g) is the original image 28, shown here for comparison.

between images 6 and 7 should be due to the same plane defined in images 1 and 2. To verify this, we marked the plane in image 6 by comparing the optical flow between images 6 and 7 with the recovered homography matrix. All the pixels with optical flow not equal to the homography matrix are considered as coming from outside the reference plane and are marked with black. Note that the threading operation does not need the plane to be present in the sequence and that the plane is used here only for the purpose of verifying the consistency of the recovered camera matrices. Figure 7 shows the results of this test.

6 Conclusion

We have presented a new result on the connection between Fundamental matrices and Trilinear tensors. This result is used to thread Fundamental matrices of consecutive images into a consistent camera trajectory. The threading operation is applied on a sliding window of triplets of images to construct a consistent camera trajectory along an extended sequence of (uncalibrated) images, without recovering 3D structure. Immediate application of the threading operation are:

- **Ego-Motion**
  The algorithm recovers a consistent camera trajectory along the image sequence without recovering 3D structure.

- **Image stabilization**
  The algorithm recovers a sequence of camera matrices that are due to the same plane. By selecting a reference plane in the first pair of images, we ensure that the same plane is stabilized throughout the sequence.

- **Multi-view Image-Based Rendering**
  The algorithm puts all the images in a single projective coordinate framework and therefor all the images can contribute to the synthesis of a novel image, using a technique such as [3].

References

[1] S. Avidan and A. Shashua. Tensorial transfer: On the representation of $n > 3$ views of a 3D scene. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[2] S. Avidan and A. Shashua. Unifying two-view and three-view geometry. In *ARPA, Image Understanding Workshop*, 1997.

[3] S. Avidan and A. Shashua. View synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[4] P.A. Beardsley, A. Zisserman, and D.W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision, 23(3):235–260*, 1997.

[5] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the European Conference on Computer Vision*, pages 563–578, Santa Margherita Ligure, Italy, June 1992.

[6] O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995.

[7] R. Hartley. Lines and points in three views — a unified approach. In *Proceedings of the ARPA Image Understanding Workshop*, Monterey, CA, November 1994.

[8] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the International Conference on Computer Vision*, 1995.

[9] R. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the International Conference on Computer Vision*, pages 882–887, Cambridge, MA, June 1995.

[10] A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the International Conference on Computer Vision*, pages 1058–1063, Cambridge, MA, June 1995.

[11] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, Washington, June 1994.

[12] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[13] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[14] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[15] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. Submitted for journal publication. A short version has appeared in International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC97), Kiel Germany Sep. 8–9, 1997, 1997.

[16] A. Shashua and P. Anandan. The generalized trilinear constraints and the uncertainty tensor. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[17] A. Shashua and S. Avidan. The rank4 constraint in multiple view geometry. In *Proceedings of the European Conference on Computer Vision*, Cambridge, UK, April 1996.

[18] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, 1996.

[19] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the International Conference on Computer Vision*, June 1995.

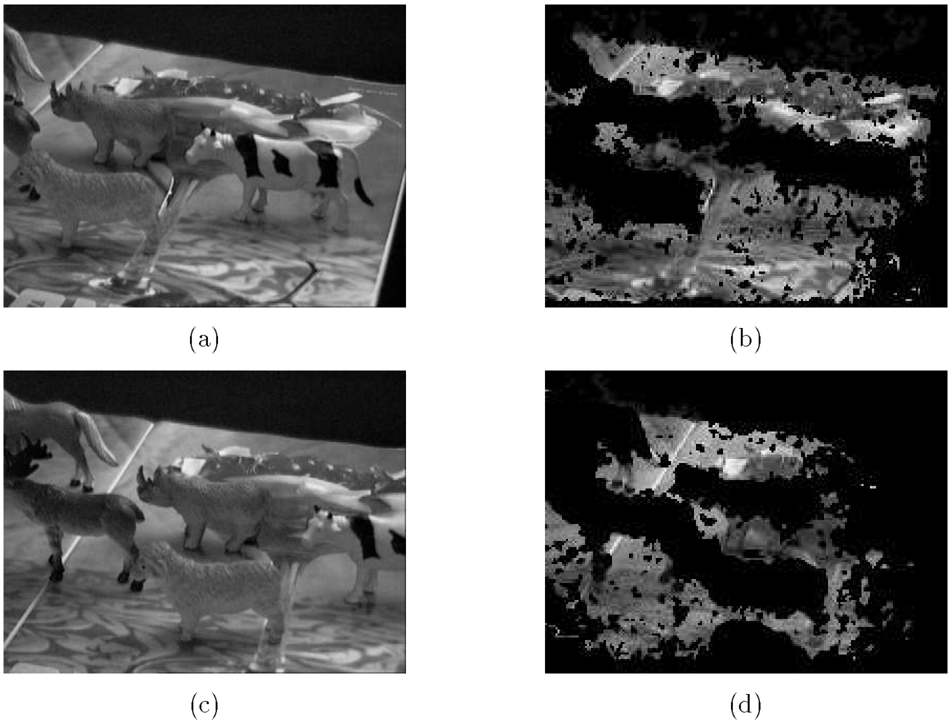


Figure 7: Stabilizing the plane in the first image over a sequence of 7 images. (a),(c) are images 1 and 6 in the sequence. (b), (d) are the same images with the pixels outside the plane marked with black pixels.

[20] M.E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–183, 1990.

[21] M.E. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *Proceedings of the ARPA Image Understanding Workshop*, 1990.

[22] G. Stein and A. Shashua. Model based brightness constraints: On direct estimation of structure and motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[23] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, 1996.

[24] B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, pages 338–343, Cambridge, MA, June 1995.

[25] T. Vieville, O. Faugeras, and Q.T. Loung. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision, 17(1):7–42*, 1996.

[26] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondences: Closed form solution, uniqueness and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3), 1992.

A Background

The background material of this paper includes (i) the Fundamental matrix, (ii) the "plane + parallax" representation, (iii) the Trilinear tensor and its contraction properties, and (iv) the reduction of the Trilinear tensor into the 2-view tensor whose components include the elements of the Fundamental matrix.

A.1 The Fundamental Matrix of Two Views

Two views $p = [I;0]\boldsymbol{x}$ and $p' \cong \mathbf{A}\boldsymbol{x}$ are known to produce a bilinear matching constraint whose coefficients are arranged in a 3×3 matrix F known as the "Essential matrix" of [12] described originally in an Euclidean setting, or the "Fundamental matrix" of [5] described in the setting of Projective Geometry (uncalibrated cameras):

$$F = [v']_{\times} A \tag{7}$$

where $\mathbf{A} = [A; v']$ (a^l_j are the elements of A - the left 3×3 minor of $\mathbf{A}$, and v' is the fourth column, the epipole, of $\mathbf{A}$). $[v']_{\times}$ denotes the skew-symmetric matrix of v'. i.e., the product with some vector u, $[v']_{\times} u$, produces the cross product between v' and u, $v' \times u$. The minor A is a 2D projective transformation from the first view onto the second via some *arbitrary* plane. In an affine setting the plane is at infinity, and in an Euclidean setting it is the rotational component of camera motion. The epipole v' is the projection of the camera center of the first camera onto the second view, and in an Euclidean setting it is the translational component of camera motion.

The Fundamental matrix satisfies the constraint $p'^{\top} F p = 0$ for all pairs of matching points p, p' in views 1 and 2, respectively. This bilinear form in image coordinates arises from the fact that the points v', Ap and p' are collinear, thus $p'^{\top}(v' \times Ap) = 0$. The matrix F can be recovered linearly from 8 matching points, and $F^{\top} v' = 0$.

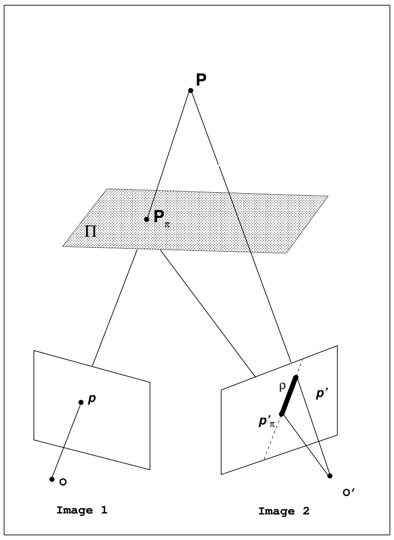


Figure 8: The Relative Affine Structure ρ (the bold line) measures how much does the point p deviates from the plane π. ρ is invariant to the position of the second camera. Stabilizing the *same* plane along a sequence of images is therefore analogous to recovering the *same* 3D structure from all the images.

A.2 Plane + Parallax Representation

The claim that recovering a consistent camera trajectory is equivalent to recovering camera matrices that are all due to the same reference plane relies on the *Relative Affine Structure* representation.

The collinearity of v', Ap and p', where A is the homography matrix due to *some* reference plane π, can be used to describe p' as follows:

$$p' \cong Ap + \rho v' \tag{8}$$

The coefficient ρ depends on the point p and the position of the plane π, is *invariant* to the choice of the second camera position (see Figure 8). Thus, by fixing the same plane along an image sequence we obtain the same relative affine structure - ρ for all the images. This is analogous to recovering the *same* 3D structure from all the images. Further details can be found in [18].

A.3 The Trilinear Tensor of Three Views

Matching image points across three views will be denoted by p, p', p''; the homogeneous coordinates will be referred to as p^i, p'^j, p''^k, or alternatively as non-homogeneous image coordinates $(x,y), (x',y'), (x'',y'')$ — hence, $p^i = (x, y, 1)$, etc.

Three views, $p = [I; 0]\boldsymbol{x}, p' \cong \mathbf{A}\boldsymbol{x}$ and $p'' \cong \mathbf{B}\boldsymbol{x}$, are known to produce four trilinear forms whose coefficients are arranged in a tensor representing a bilinear function of the camera matrices $\mathbf{A}, \mathbf{B}$:

$$\mathcal{T}_i^{jk} = v'^j b_i^k - v''^k a_i^j \tag{9}$$

where $\mathbf{A} = [a_i^j, v'^j]$ (a_i^j is the 3×3 left minor and v' is the fourth column of $\mathbf{A}$) and $\mathbf{B} = [b_i^k, v''^k]$. The tensor acts on a triplet of matching points in the following way:

$$p^i s_j^\mu r_k^\rho \mathcal{T}_i^{jk} = 0 \tag{10}$$

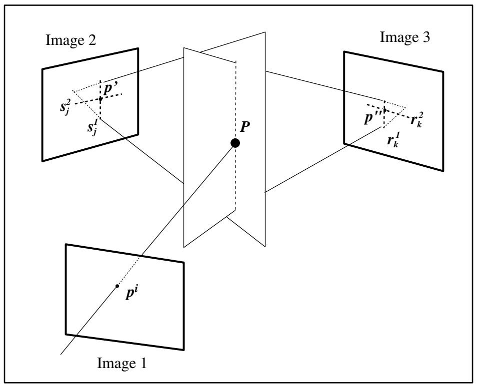


Figure 9: Each of the four trilinear equations describes a matching between a point p in the first view, some line s_j^μ passing through the matching point p' in the second view and some line r_k^ρ passing through the matching point p'' in the third view. In space, this constraint is a meeting between a ray and two planes.

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 4 trilinear equations (unique up to linear combinations). If we choose the *standard* form where s^μ (and r^ρ) represent vertical and horizontal scan lines, i.e.,

$$s_j^\mu = \begin{bmatrix} -1 & 0 & x' \\ 0 & -1 & y' \end{bmatrix}$$

then the four trilinear forms, referred to as *trilinearities*[14], have the following explicit form:

$$\begin{aligned}
x''\mathcal{T}_i^{13}p^i - x''x'\mathcal{T}_i^{33}p^i + x'\mathcal{T}_i^{31}p^i - \mathcal{T}_i^{11}p^i &= 0,\\
y''\mathcal{T}_i^{13}p^i - y''x'\mathcal{T}_i^{33}p^i + x'\mathcal{T}_i^{32}p^i - \mathcal{T}_i^{12}p^i &= 0,\\
x''\mathcal{T}_i^{23}p^i - x''y'\mathcal{T}_i^{33}p^i + y'\mathcal{T}_i^{31}p^i - \mathcal{T}_i^{21}p^i &= 0,\\
y''\mathcal{T}_i^{23}p^i - y''y'\mathcal{T}_i^{33}p^i + y'\mathcal{T}_i^{32}p^i - \mathcal{T}_i^{22}p^i &= 0.
\end{aligned}$$

These constraints were first derived in [14]; the tensorial derivation leading to Eqns. 9 and 10 was first derived in [16]. The Trilinear tensor has been well known in disguise in the context of Euclidean line correspondences and was not identified at the time as a tensor but as a collection of three matrices (a particular contraction of the tensor known as correlation contractions) [20, 21, 26]. The link between the two and the generalization to projective space was identified later in [7, 9]. Additional work in this area can be found in [19, 6, 24, 10, 17, 1, 3, 22].

A.4 Contraction Properties of the Tensor

The lines s_j^μ coincident with p' and the lines r_k^ρ coincident with p'' for a basis for all lines coincident with p' and p'', thus we readily have the "point+line+line" property:

$$p^i s_j r_k \mathcal{T}_i^{jk} = 0 \tag{11}$$

where s_j is *some* line through p' and r_k is *some* line through p''. Similarly, since $s_j r_k \mathcal{T}_i^{jk}$ is a line (coincident with p), then a triplet of matching lines provides two constraints:

$$s_j r_k \mathcal{T}_i^{jk} \cong q_i \tag{12}$$

for all lines q, s, r coincident with the points p, p', p'' (in particular, matching lines). The third point-line property is the "reprojection" constraint:

$$p^i s_j \mathcal{T}_i^{jk} \cong p''^k \tag{13}$$

which provides a direct means for "transfer" of image measurements from views 1,2 onto view 3 (prediction of p'' from views 1,2).

The tensor has certain contraction properties and can be sliced in three principled ways into matrices with distinct geometric properties divided into two families: *Homography Contractions* and *Correlation Contractions*. We will briefly introduce the Homography contractions described originally in [19] — further details on that and on Correlation contractions can be found in [15].

Consider the matrix arising from the contraction,

$$\delta_k \mathcal{T}_i^{jk} \tag{14}$$

which is a 3×3 matrix, we denote by E, obtained by the linear combination $E = \delta_1 \mathcal{T}_i^{j1} + \delta_2 \mathcal{T}_i^{j2} + \delta_3 \mathcal{T}_i^{j3}$ (which is what is meant by a contraction), and δ_k is an *arbitrary* covariant vector. The matrix E has a general meaning introduced in [19]:

Proposition 1 (Homography Contractions) *The contraction $\delta_k \mathcal{T}_i^{jk}$ for some arbitrary δ_k is a homography matrix from image one onto image two determined by the plane containing the third camera center C'' and the line δ_k in the third image plane. Generally, the rank of E is 3. Likewise, the contraction $\delta_j \mathcal{T}_i^{jk}$ is a homography matrix from image one onto image three.*

For proof see [19]. Clearly, since δ is spanned by three vectors, we can generate up to at most three distinct homography matrices by contractions of the tensor. We define the *Standard Homography Slicing* as the homography contractions associated by selecting δ be $(1,0,0)$ or $(0,1,0)$ or $(0,0,1)$, thus the three standard homography slices between image one and two are $\mathcal{T}_i^{j1}, \mathcal{T}_i^{j2}$ and $\mathcal{T}_i^{j3}$, and we denote them by E_1, E_2, E_3 respectively, and likewise the three standard homography slices between image one and three are $\mathcal{T}_i^{1k}, \mathcal{T}_i^{2k}$ and $\mathcal{T}_i^{3k}$, and we denote them by W_1, W_2, W_3 respectively.

A.5 The 2-view Tensor

We return to Equation 9 and consider the case where the third image coincide with the second. The camera matrices for both images are $\mathbf{A} = [A; v']$ and this special tensor can be written as:

$$\mathcal{F}_i^{jk} = v'^j a_i^k - v'^k a_i^j \tag{15}$$

which is composed of the elements of the Fundamental matrix, as the following lemma shows.

Lemma 1 *The two-view-tensor $\mathcal{F}_i^{jk}$ is composed of the elements of the Fundamental matrix:*

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

where F_{li} is the Fundamental matrix and ϵ^{ljk} is the cross-product tensor.

Proof: We consider Equation 9 with $\mathcal{F}_i^{jk} = \epsilon^{ljk} F_{li}$ to derive the following equalities:

$$\begin{aligned} p^i s_j r_k \mathcal{F}_i^{jk} &= \\ p^i s_j r_k (\epsilon^{ljk} F_{li}) &= \\ p_i \underbrace{(s_j r_k \epsilon^{ljk})}_{p''^l} F_{li} &= 0 \end{aligned} \tag{16}$$

□

The two-view-tensor is an admissible tensor that embodies the Fundamental matrix in a three-image-framework. Algorithm that works with the Trilinear tensor of three views can work with this tensor as well. In particular, the point-line contractions and the Homography contractions hold, for example:

$$p^i s_j \mathcal{F}_i^{jk} \cong p'^k$$

which takes p and a line s coincident with p' and produces p'. The contraction $\delta_k \mathcal{F}_i^{jk}$ is a homography contraction, i.e., produces a homography matrix from view 1 onto view 2 given by the plane coincident with the center of projection of camera 2 and the line δ in view 2. Similarly to the 3-view tensor, the Standard Homography Slices correspond to setting δ to $(1,0,0)$ or $(0,1,0)$ or $(0,0,1)$. Further details can be found in [2].