Unifying Two-View and Three-View Geometry

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

Abstract

The core of multiple-view geometry is governed by the fundamental matrix and the trilinear tensor. In this paper we unify both representations by first deriving the fundamental matrix as a rank-2 trivalent tensor, and secondly by deriving a unified set of operators that are transparent to the number of views. As a result, we show that the basic building block of the geometry of multiple views is a trivalent tensor that specializes to the fundamental matrix in the case of two views, and is the trilinear tensor (rank-4 trivalent tensor) in case of three views. The properties of the tensor (geometric interpretation, contraction properties, etc.) are independent of the number of views (two or three). As a byproduct, every two-view algorithm can be considered as a degenerate three-view algorithm and three-view algorithms can work with either two or three images, all using one standard set of tensor operations. To highlight the usefulness of this paradigm we provide two practical applications. First we present a novel view synthesis algorithm that starts with the rank-2 tensor and seamlessly move to the general rank-4 trilinear tensor, all using one set of tensor operations. The second application is a camera stabilization algorithm, originally introduced for three views, now working with two views without any modification.

1 Introduction

The geometry of multiple views is governed by certain multi-linear constraints, bilinear for pairs of views and trilinear for triplets of views — all other multilinear constraints (four views and beyond) are spanned by the bilinear and trilinear constraints. The bilinear constraint determines the "fundamental matrix" and the trilinear constraints determine the "trilinear tensor". The fundamental matrix is a rank-2 3×3 matrix and the trilinear tensor is a rank-4 trivalent tensor. There are known properties of the fundamental matrix, there are known properties of the trilinear tensor, and there are known connections between the two — for instance how to extract the fundamental matrix from the trilinear tensor. There are algorithms (for reconstruction, view synthesis, camera stabilization) that are defined for concatenation of pairs of views, and there are algorithms that are defined for concatenation of triplets of views. What is needed, therefore, is a canonical representation, a single object with a standard set of operators, that applies uniformly to pairs or triplets of views. In other words, the unification efforts that have appeared so far in the literature focus on the transformation groups (projective, affine and Euclidean) represented by the camera matrix, leading to a canonical framework [3, 15, 9] for the geometry of two views. Given the recent progress on multi-linear tensorial constraints across more than two views, there is a need to make a similar unification attempt but now across the temporal axis (number of views), rather than on the spatial axis (transformation groups).

The paper has two main results. First, we establish a set of operators that are used to synthesize tensors from one another. Second, we derive the geometry of two views using those operators and show that the familiar fundamental matrix is embedded in a rank-2 trivalent tensor (of 27 coefficients). We show that the properties of the rank-2 tensor are identical with the known properties of the rank-4 trilinear tensor (of three distinct views), and the set of operators apply uniformly to both tensors. As a result, the geometry of multiple views is governed by a single tensorial structure with a standard set of operators and is uniform with respect to the number of views — the only change that occurs when the number of views is two is that the rank of the tensor becomes 2 instead of 4, but this does not have an effect on the manner in which the tensor is used for applications.

Apart from the theoretical result, we show practical benefits of this unification step. First is the "cross-platform" capability of algorithms to work both in the case of two and three views, as the latter is simply a generalization of the former. This results in the ability to handle freely and seemlessly the geometry of two and three images in a single framework. Instead of existing two-views algorithms one can use three-view based algorithms, taking advantage of the third view, in case it is present, but working with two images as well without modification, all due to the introduction of the rank-2 tensor. To demonstrate these properties we present two applications — a novel view synthesis algorithm that highlights the simple handling of the geometry of two or three images and a video stabilization algorithm that works, as is, with two or three images.

2 Background and Notations

We assume that the physical 3D world is represented by the 3D projective space $\mathcal{P}^3$ (object space) and its projections onto the 2D projective space $\mathcal{P}^2$ defines the image space. If $\boldsymbol{x} \in \mathcal{P}^3$ varies over the object space, represented by a tetrad of homogeneous coordinates, and $p \in \mathcal{P}^2$ is its projection (represented by a triplet of coordinates), then there exists a 3×4 matrix A satisfying the relation $p \cong A\boldsymbol{x}$, where $\cong$ represents equality up to scale and A is called the camera matrix. Since only relative camera positioning can be recovered from image measurements, the first camera matrix can be represented by $[I;0]$. In a pair of views, $p = [I;0]\boldsymbol{x}$ and $p' \cong A\boldsymbol{x}$, the left 3×3 minor of A stands for a 2D projective transformation of the chosen plane at infinity and the fourth column of A stands for the epipole (the projection of the first camera center on the image plane of the second camera). In particular, in a calibrated setting the 2D projective transformation is the rotational component of camera motion and the epipole is the translational component of camera motion.

We will occasionally use tensorial notations, which are briefly 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. For example, if p is a point incident to a line s in $\mathcal{P}^2$, then $p^i s_i = 0$. Vectors are also called 1-valence tensors. 2-valence tensors (matrices) have two indices and the transformation they represent depends on the covariant-contravariant positioning of the indices. For example, a_i^j is a mapping from points to points, and hyper-planes to hyper-planes, because $a_i^j p^i = q^j$ and $a_i^j s_j = r_i$ (in matrix form: $Ap = q$ and $A^\top s = r$); a_{ij} maps points to hyper-planes; and a^{ij} maps hyper-planes to points. When viewed as a matrix the row and column positions are determined accordingly: in a_i^j and a_{ji} the index i runs over the columns and j runs over the rows, thus $b_j^k a_i^j = c_i^k$ is $BA = C$ in matrix form. An outer-product of two 1-valence tensors (vectors), $a_i b^j$, is a 2-valence tensor c_i^j whose i, j entries are $a_i b^j$ — note that in matrix form $C = ba^\top$. An n-valence tensor described as an outer-product of n vectors is a rank-1 tensor. Any n-valence tensor can be described as a sum of rank-1 n-valence tensors. The rank of an n-valence tensor is the *smallest* number of rank-1 n-valence tensors with sum equal to the tensor. For example, a rank-1 trivalent tensor is $a_i b_j c_k$ where a_i,b_j and c_k are three vectors. The rank of a trivalent tensor α_{ijk} is the smallest r such that,

$$\alpha_{ijk} = \sum_{s=1}^{r} a_{is} b_{js} c_{ks}. \tag{1}$$

The tensor of vector products is denoted by ϵ_{ijk} (indices range 1-3) operates on two contravariant vectors of the 2D projective plane and produces a covariant vector in the dual space (a line): $\epsilon_{ijk} p^i q^j = s_k$, which in vector form is $s = p \times q$, i.e., s is the vector product of the points p and q.

Two views $p = [I;0]\boldsymbol{x}$ and $p' \cong A\boldsymbol{x}$ are known to produce a bilinear matching constraint whose coefficients are arranged in a 3×3 matrix F known as the "Essential matrix" of [8] described originally in an Euclidean setting, or the "Fundamental matrix" of [2] described in the setting of projective geometry (uncalibrated cameras):

$$f_{ij} = \epsilon_{ikl} v'^k a_j^l \tag{2}$$

where $A = [a; v']$ (a_j^l is the left 3×3 minor of A, and v' is the fourth column, the epipole, of A). The bilinear constraint is $f_{ij} p^i p'^j = 0$.

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

$$\alpha_i^{jk} = v'^j b_i^k - v''^k a_i^j \tag{3}$$

where $B = [b, v'']$. The four trilinear constraints are:

$$p^i s_j^\mu r_k^\rho \alpha_i^{jk} = 0 \tag{4}$$

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'' (see Fig. 1). Since the free indecis are μ, ρ each in the range 1,2, we have 4 trilinear equations (which are unique up to linear combinations). By changing the order of the views one can obtain at most 12 trilinear constraints arranged in three such tensors. These constraints first became prominent in [11] and the underlying theory has been studied intensively since in [16, 6, 12, 4, 17, 7, 13].

The elements of α_i^{jk} satisfy certain properties. The algebraic relations among the elements are described in [4], and contraction properties in [16]. Among the contraction properties it will be useful for later to mention that $\delta_k \alpha_i^{jk}$ (for any vector δ) produces a 2D projective transformation (a homography) from image 1 to 2 via some plane of reference, and $\eta_j \alpha_i^{jk}$ produces a homography matrix from image 1 to 3 via some plane. The orientation of the plane of reference is determined by δ, and if we set $\delta = (1,0,0), (0,1,0)$ and $(0,0,1)$ we obtain three homography matrices associated with planes attached to the coordinate frame of the third camera (in a calibrated setting the planes are normal to the coordinate axes, see Fig. 2). If we denote these

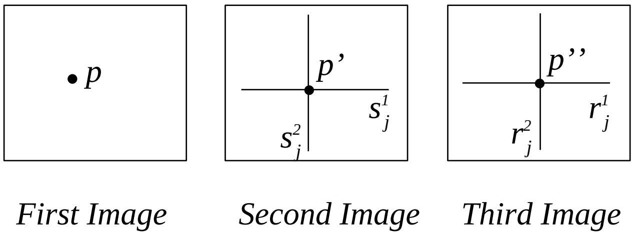

Figure 1: The trilinear tensor of three images relates a point in the first image to lines in the second and third images.

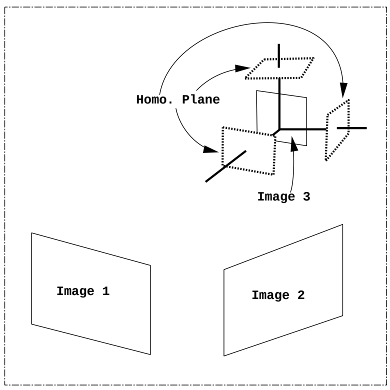

Figure 2: The Three homography planes (dashed), related to the coordinate system of the third image, defining homography matrices from the first image to the second.

homography matrices by E_1, E_2, E_3, then the fundamental matrix of images 1,2 is the solution to the over-determined linear system $E_j^\top F + F^\top E_j = 0$, and the elements of the epipolar point v'' are the repeated generalized eigenvalues of pairs E_i, E_j. Finally, it has been recently shown in [14] that the rank of α_i^{jk} is 4 (we will return to tensor ranks later).

3 The Basic Tensorial Operators

The basic operation described next describes the transformation a tensor of three views undergoes when one of the cameras changes its position. In other words, this operator can be used to create a chain of tensors, each created from its predecessor in the chain. Later, we will start from a Null tensor (all three views are repeated) and create a chain that along the way creates a representation of two views as a rank-2 trivalent tensor.

Consider the tensor α_i^{jk} of the views $< 1,2,3 >$ (in that order), and let the coordinate frame of camera 3 undergo a 2D transformation (say a rotation of the coordinate axis), i.e., p'' is replaced by $R^{-1}p''$. Clearly, the camera matrix B is replaced by RB and therefore the tensor of the cameras $[I;0]$, A and RB is $\beta_i^{jk} = r_l^k \alpha_i^{jl}$ since,

$$\beta_i^{jk} = r_l^k \alpha_i^{jl} = v'^j(r_l^k b_i^l) - (v''^l r_l^k) a_i^j, \qquad (5)$$

where $RB = [r_l^k b_i^l; v''^l r_l^k]$. Consider next translating the center of the third camera RB by a vector t (see Fig. 3). Clearly, the new camera matrix is now $C = [r_l^k b_i^l; v''^l r_l^k + t^k]$. Therefore, the tensor γ_i^{jk} of the cameras $[I;0], A, C$ is:

$$\gamma_i^{jk} = v'^j(r_l^k b_i^l) - (v''^l r_l^k + t^k) a_i^j = r_l^k \alpha_i^{jl} - t^k a_i^j. \quad (6)$$

We have thus proved the following theorem:

Theorem 1 *Given a tensor α_i^{jk} of camera positions $[I;0], A, B$, then the tensor γ_i^{jk} of camera positions $[I;0], A, C$, where C is obtained from B by an incremental change of coordinates R and translation t from the position of the third camera has the form:*

$$\boxed{\gamma_i^{jk} = r_l^k \alpha_i^{jl} - t^k a_i^j} \qquad (7)$$

Likewise, if we apply an incremental change to the position of the second camera, rather than the third, then the tensor γ_i^{jk} will have the form:

$$\boxed{\gamma_i^{jk} = r_l^j \alpha_i^{lk} + t^j b_i^k} \qquad (8)$$

4 The rank-2 Trivalent Tensor of Two Views

We will use the tensorial operators (eqns. 7 and 8) to create a chain starting from the (Null) tensor of views $< 1,1,1 >$ (all three views are repeated), to tensor of views $< 1,2,1 >$, to tensor $< 1,2,2 >$ and finally to tensor α_i^{jk} of views $< 1,2,3 >$. All the tensors of the chain are trivalent tensors, and of interest are the tensors that represent only two distinct views.

The tensor β_i^{jk} of views $< 1,2,1 >$ can be derived from the Null tensor using eqn. 8,

$$\begin{aligned} \beta_i^{jk} &= a_l^j \alpha_i^{lk} - v'^j b_i^k \\ &= a_l^j 0_i^{lk} + v'^j I_i^k \\ &= v'^j I_i^k, \end{aligned} \qquad (9)$$

by the incremental motion $A = [a, v']$ from views $< 1,1,1 >$ to views $< 1,2,1 >$. The elements of the tensor are either 0 or the epipole v'.

Next, we apply an incremental motion of the the third view going from tensor of views $< 1,2,1 >$ to tensor of views $< 1,2,2 >$. The incremental motion is again $A = [a, v']$ and we use the operator described in eqn 7 to obtain:

$$\begin{aligned} \gamma_i^{jk} &= a_l^k \beta_i^{jl} - v'^k a_i^j \\ &= a_l^k (v'^j I_i^l) - v'^k a_i^j \\ &= v'^j a_i^k - v'^k a_i^j \end{aligned} \qquad (10)$$

and γ_i^{jk} is the tensor of the image triplet $< 1, 2, 2 >$. It can be readily verified that the elements of γ_i^{jk} are composed of the fundamental matrix $f_{ij} = \epsilon_{ikl} v'^k a_j^l$, $-f_{ij}$, and the remaining (nine) elements vanish. In other words, we have derived a trivalent tensor representing the geometry of two views, and is composed of the elements of the fundamental matrix.

Theorem 2 *The rank of the trivalent tensor of two views,*

$$\gamma_i^{jk} = v'^j a_i^k - v'^k a_i^j$$

is 2.

Proof: The 2D homography a_i^j represents a family of projective transformations determined by 4 parameters (representing the orientation of the plane at infinity) — the entire family produces the same tensor γ_i^{jk}. Among the 4-parameter family of homography matrices, there is a subset 3-parameter family of rank-2 matrices (these correspond to the 2D transformation from image one to image two via a plane coplanar with the camera center of the second view) that has the form $[c]_x F$ where $[]_x$ denotes the skew-symmetric matrix of vector products and F is the fundamental matrix (see Corollary 1). Therefore, the minimal decomposition of a_i^j is into two rank-1 matrices. Choose a rank-2 homography matrix for which v' is one of the eigenvectors. This is possible because Fv' is a line coincident with v', thus choose c as another line coincident with v' thereby obtaining $[c]_x F v' \cong v'$. Thus, it is always possible to choose the following decomposition:

$$a_i^j = \lambda_1 v_i' v'^j + \lambda_2 a_{i3} a^{j4}, \tag{11}$$

i.e., by performing a Singular Value Decomposition and choosing the first element of the decomposition (the vector a_i) to be one of the eigenvectors (v') of the matrix a_i^j (see [5], pp. 71). Then, γ_i^{jk} is decomposed to:

$$\begin{aligned} \gamma_i^{jk} &= v'^j(\lambda_1 v_i' v'^k + \lambda_2 a_{i3} a^{k4}) - \\ & \quad v'^k(\lambda_1 v_i' v'^j + \lambda_2 a_{i3} a^{j4}) \\ &= \lambda_2 (v'^j a_{i3} a^{k4} - v'^k a_{i3} a^{j4}) \end{aligned} \tag{12}$$

because the tensors $v'^j v_i' v'^k$ and $v'^k v_i' v'^j$ have identical elements, thus their difference vanishes. Hence, γ_i^{jk} has a minimal decomposition of two rank-1 tensors. ∎

Theorem 3 *the tensor γ_i^{jk} shares the same properties as the general rank-4 tensor of three views.*

Proof: From [16] we need to show that the contraction properties hold. In other words, that $\delta_k \gamma_i^{jk}$ produces a homography matrix for any vector δ; second, the connection between the repeated eigenvalues of pairs of such homography matrices and the epipole v'.

Consider $\delta = (1, 0, 0), (0, 1, 0)$ and $(0, 0, 1)$. The E-matrices we obtain are:

$$E_{(1)} = (1, 0, 0)_k \gamma_i^{jk} = v'^j a_i^1 - v'^1 a_i^j$$

$$E_{(2)} = (0, 1, 0)_k \gamma_i^{jk} = v'^j a_i^2 - v'^2 a_i^j$$

$$E_{(3)} = (0, 0, 1)_k \gamma_i^{jk} = v'^j a_i^3 - v'^3 a_i^j$$

each of these matrices satisfies the general form of a homography matrix: $h_i^j = \lambda a_i^j + v'^j n_i$, where λ and n represent the orientation and position of the reference plane.

Next, we must show that the repeated generalized eigenvalue of the pair $E_{(k)}, E_{(l)}$ is an element of the epipole v'. Namely, that the following matrix is of rank-1:

$$\begin{aligned} E_{(k)} - \frac{v'^k}{v'^l} E_{(l)} &= (v'^j a_i^k - v'^k a_i^j) - \\ & \quad \frac{v'^k}{v'^l}(v'^j a_i^l - v'^l a_i^j) \\ &= v'^j (a_i^k - \frac{v'^k}{v'^l} a_i^l) - \\ & \quad a_i^j (v'^k - v'^l \frac{v'^k}{v'^l}) \\ &= v'^j (a_i^k - \frac{v'^k}{v'^l} a_i^l) \end{aligned} \tag{13}$$

∎

A byproduct of these properties is that we can characterize the family of rank-2 homography matrices:

Corollary 1 *$[c]_\times F$, where $c = (c_1, c_2, c_3)$ is a general 3-vector, defines a family of homography matrices from the first image to the second image due to a plane passing through the center of projection of the second camera.*

Proof: Simply note that $\sum_{i=1}^3 c_i E_{(i)} = [c]_\times F$ where $[]_x$ denotes the skew-symmetric matrix of vector products. The $E_{(i)}$ are rank-2 homography matrices, therefore their span is also a homography matrix, it is of rank-2 because their span is represented by a product with a rank-2 matrix $[c]_x$. ∎

The corollary extends the result of [9] that $[v']_\times F$ is a homography matrix to a family of homography matrices $[c]_\times F$ passing thru the center of projection of the second camera.

Finally, note that the bilinear constraint follows from γ_i^{jk} in the same manner as in the general rank-4 tensor: $p^i s_j r_k \gamma_i^{jk} = 0$ describes a contraction with the point p in the first view, some line s passing through p' and some other line r passing through p' as well. Thus, we get the same point-line-line interpretation we get with the general rank-4 tensor.

The last tensor in the chain is to go from tensor of views $< 1, 2, 2 >$ to the general tensor of views $< 1, 2, 3 >$. This can be readily done using the operator of eqn. 7.

To conclude, we have shown the basic "building block" of stereo vision to be the trilinear tensor of three cameras. Every other object, be it the epipole or the fundamental matrix, is merely a degenerate case of the general trilinear tensor. Since camera parameters can be recovered directly from the trilinear tensor, there is no need for the fundamental matrix, other than to serve as a tool for constructing the rank-2 trivalent tensor in case only two, rather than three, views are given. As a result algorithms developed under the three-view paradigm will apply to all camera configurations, be it two or three cameras.

5 Applications

This section presents two applications to highlight two of the ideas advocated in this paper. The first example highlights the simple and uniform way to treat tensors (both rank-4 and rank-2) in order to obtain new ones. There is no need to distinguish between the geometry of two and three views. Specifically we present an image-based rendering algorithm that starts with a pair of images, related by a rank-2 tensor, and generate a novel view by seamlessly moving from the rank-2 tensor to the rank-4 tensor. The second application demonstrates the generality of algorithms developed in tensor context - they act the same both for the case of two views and three views. We show this on a stabilization algorithm originally developed in the three-view framework.

5.1 Novel View Synthesis

Novel view synthesis, also referred to as image-based rendering, aims at synthesizing novel views of a scene from a given pair of images, without first reconstructing the 3D model. This method can be faster and more accurate to compute than building the 3D model first. The trilinear tensor is an ideal candidate for image-based rendering system, as it is numerically stable and has no degenerate configurations. We use the basic tensor operators described earlier and the rank-2 trivalent tensor of two views to build new tensors. Once a tensor is built we use equation 4 to re-project the novel image. The algorithm is composed of two stages - a preprocessing stage that is done once and the actual re-projection scheme that is calculated for every new image.

1. **Preprocessing**
   (a) Compute dense correspondence between the two model images.
   (b) Recover the fundamental matrix of the two model images.
   (c) Construct the rank-2 trivalent tensor $< 1,2,2 >$ from the fundamental matrix elements.
   (d) Recover the rotation angles between the first two images.
2. **View Synthesis**
   (a) Accept camera motion parameters (rotation/translation) from the second camera to its new position.

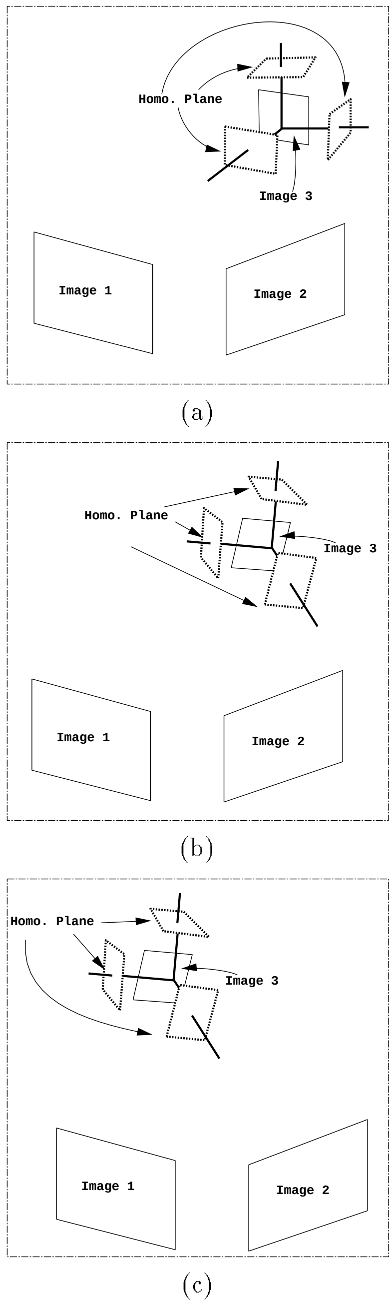

Figure 3: Original tensor α_i^{jk} (a). Original tensor after rotation $r_l^k \alpha_i^{jl}$ (b), The coordinate system of the third camera has rotated. Original tensor after rotation and translation $r_l^k \alpha_i^{jl} - t^k a_i^j$ (c) The coordinate system of the third camera has now translated along the x-axis.

(b) Apply equation 7 to compute the tensor of the two model images and the novel image. Use the rotation angles as recovered in the preprocessing stage.

(c) Render the novel view, using the point correspondence between the two model images and the tensor computed in the previous step.

5.2 Implementation

In the following example two images of the head were taken, one after the other, using a Canon VC-C1 digital camera. We than computed the optical flow between the two images (to obtain the dense correspondence) and recovered the rank-2 tensor, as explained earlier. Next the user specifies the new camera position in terms of rotation and translation from the second image position. We generate the new tensor of images $1, 2$ and the new image using equation 7. See figure 4 for results. The full movie can be seen at the attached web-page at http:www-seq-1.html .

5.3 Video Stabilization

This application illustrates the "cross-platform" capability of three-view algorithms. As an example we show how to convert a three-view stabilization algorithm, originally presented in [10], to work with two images only. The purpose of the stabilization algorithm was defined to cancel rotation between successive frames. The original paper makes use of the fact that the tensor is composed of three homography matrices to establish a linear relation between the elements of the trilinear tensor to those of the rotation matrix, in case the cameras are calibrated and the angles are small. Formally, a very simple, closed-form, expression relating the tensor α_i^{jk} and $\Omega_X, \Omega_Y, \Omega_Z$ (The angles of the rotation matrix from the first image to the second) was derived

$$
\begin{aligned}
\Omega_X &= det \begin{pmatrix} \alpha_2^{j3} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
\Omega_Y &= det \begin{pmatrix} -\alpha_1^{j3} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
\Omega_Z &= det \begin{pmatrix} \alpha_1^{j2} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
K &= det \begin{pmatrix} \alpha_2^{j2} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} \qquad (14)
\end{aligned}
$$

where α_2^{j2} stands for $(\alpha_2^{12},\ \alpha_2^{22},\ \alpha_2^{32})$, etc. This expression recovers directly and simply small rotations from the trilinear tensor. The stabilization algorithm proceeds as follows:

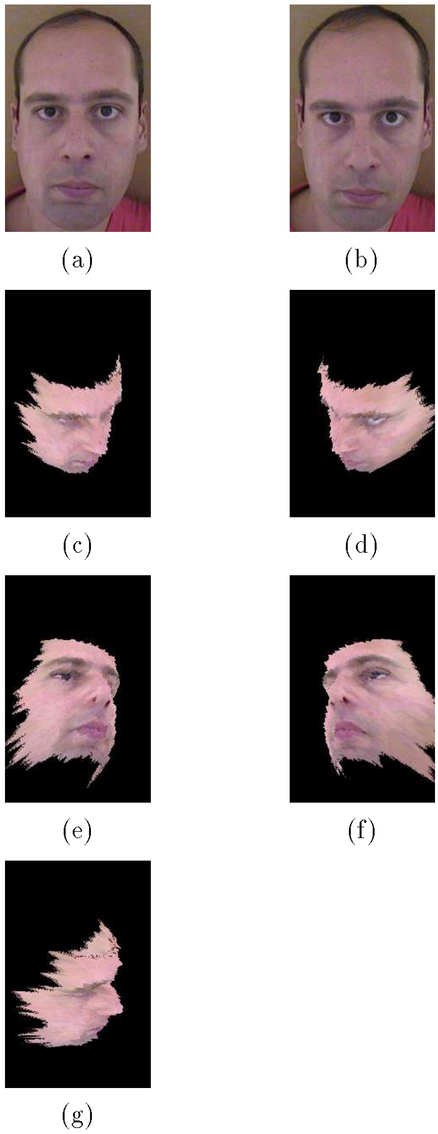

Figure 4: The original two images ((a),(b)). Novel views of the face ((c)-(g)).

1. Given three successive frames $n, n+1, n+2$, compute the trilinear tensor.

2. Recover the small-angles rotation matrix between frames n and $n+1$ as expressed in equation 14.

3. Derotate frame $n+1$ using the inverse of the rotation matrix recovered in the previous step.

5.3.1 The Two-View Stabilization Algorithm

In the two-view case the tensor used to recover the rotation is a rank-2 trivalent tensor of two views, rather than the general rank-4 trilinear tensor. Since the algorithm makes use of tensor properties (in this case the fact that the rank-4 trilinear tensor is composed of three homography matrices) and since the rank-2 tensor posses this property as well, the algorithm can go unchanged. For clarity we present the modified algorithm:

1. Given two successive frames $n, n+1$, compute the fundamental matrix.

2. Rearrange the fundamental matrix elements to obtain the rank-2 tensor of two views.

3. Recover the small-angles rotation matrix between frames n and $n+1$ as expressed in equation 14.

4. Derotate frame $n+1$ using the inverse of the rotation matrix recovered in the previous step.

5.4 Implementation

We tested our method on gray scale images with a resolution of 640×480 pixels. We computed the fundamental matrix between the two images and applied the algorithm described in the previous subsection. Figure 5 shows the two input images as well as an average image of the original images and an average image of the two images after rotation cancellation. For verification we compared our results with the three-view algorithm, by adding a third image. The recovered rotation angles differed by less than 0.01 radians. The visual result was indistinguishable.

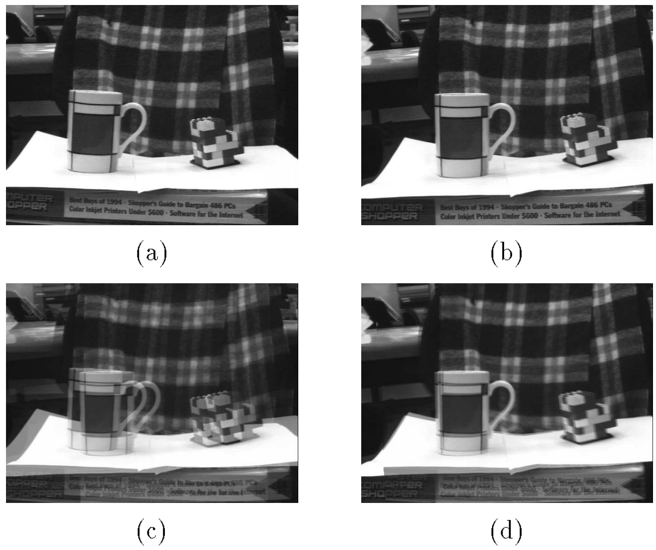

Figure 5: The original two images ((a),(b)). Average of the original two images (c). Average of the two images after rotation cancellation (d).

6 Conclusion

We unified two-view, three-view and, as a result, multi-view geometry with the trilinear tensor as the connecting thread. This was done by developing a basic tensorial operator that describes the change in the tensor elements as a result of camera motion and using it to create a chain of tensors that include the epipole, the fundamental matrix - as a rank-2 trivalent tensor, and the rank-4 trilinear tensor in a single framework. The rank-2 tensor of two views and the rank-4 tensor of three views share the same properties and are governed by a single set of basic tensorial operators. As a result algorithms developed under the three-view paradigm will apply to all camera configurations, be it two or three cameras. Apart from the theoretical result, we showed two practical examples that make use of this theory. An image-based rendering application that uses the basic tensorial operators to seemlessly move from rank-2 tensor (representing the geometry of two views) to rank-4 tensors (representing the geometry of three views), and an image-stabilization algorithm that works unchanged for two or three images.

References

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. CS report, CIS-9602, Technion, January 1996.

[2] 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.

[3] O.D. Faugeras. Stratification of three-dimensional vision: projective, affine and metric representations. *Journal of the Optical Society of America*, 12(3):465–484, 1995.

[4] 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.

[5] G.H. Golub and C.F. Van Loan. *Matrix computations*. John Hopkins University Press, 1989.

[6] 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.

[7] 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.

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

[9] Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. In *Proceedings of the European Conference on Computer Vision*, pages 589–599, Stockholm, Sweden, May 1994. Springer Verlag, LNCS 800.

[10] B. Rousso, S. Avidan, A. Shashua and S. Peleg. Robust Recovery of Camera Rotation from Three Frames. In *IEEE Conference on Computer Vision and Pattern Recognition*, San-Francisco, CA., June 1996.

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

[12] 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.

[13] 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.

[14] A. Shashua and S.J. Maybank. Degenerate n point configurations of three views: Do critical surfaces exist? Technical report, Hebrew University of Jerusalem, November 1996.

[15] 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.

[16] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *Proceedings of the International Conference on Computer Vision*, June 1995.

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