Robust Recovery of Camera Rotation from Three Frames *

B. Rousso **S. Avidan** **A. Shashua**† **S. Peleg**‡
Institute of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
e-mail : rousso@cs.huji.ac.il

Abstract

Computing camera rotation from image sequences can serve many computer vision applications. One direct application is image stabilization, and when the camera rotation is known the computation of camera translation and 3D scene structure are much simplified. A new approach for recovering camera rotation is presented in this paper, which proves to be much more robust than existing methods by avoiding the computation of the epipole. Another benefit of the new approach is that it does not assume any specific scene structure.

The rotation matrix of the camera is computed explicitly from three homography matrices, recovered using the trilinear tensor which describes the relations between the projections of a 3D point into three images. The entire computation is linear for small angles, and is therefore fast and stable. Iterating the linear computation can then be used to recover larger rotations as well.

1 Introduction

Recovering camera rotation is one of the basic steps in many image sequence applications, such as electronic image stabilization. Most existing methods take one of the following two approaches. One approach is to compute the camera rotation only after computing the camera translation (the epipole) [18, 4, 8, 12]. The second approach assumes a specific 3D scene structure, e.g. assuming the existence and the detection of a 3D plane in the scene [8, 17, 13, 10].

We propose a new method to recover rotations using three homography matrices, without using the epipoles and without assuming any specific 3D model. A homography is a transformation that maps the image of a 3D plane in one frame into its image in the second frame, and it can be represented as a 3×3 matrix (see Section 2). The homographies used do not correspond to any planes that have to be present in the image, and therefore there is no restriction on the scene structure. The benefits of our method are increased accuracy, as epipoles are believed to be a source for error [22], and more general applicability as no specific 3D model is assumed.

The theoretical background of this paper includes a recently discovered property of homography matrices between image pairs, namely that homography matrices between two images form a linear space of rank 4 (See [16]). We take this general result further and show that under small-motion assumptions (which is typically the case in video sequence processing) the rotational component of camera motion (which is the homography due to the plane at infinity) is spanned by only three homography matrices.

There are several possibilities for finding the three homography matrices which are needed to compute the camera rotation [7, 1, 2, 10, 13, 17]. However, most methods to compute homographies between two images assume that each homography corresponds to an actual planar surface in the scene. Even when several planar surfaces do exist in the scene, the accuracy of these methods reduces as more homographies are extracted.

The best method we found for computing the needed homographies is by using the trilinear tensor between three images [15, 4, 18, 5]. The trilinear tensor is relatively accurate since it is computed from three frames, rather than only two, and no 3D scene structure is assumed. We then use the recently discovered contraction property into three homography matrices [18] to obtain a closed-form solution for the rotational component of camera motion from the trilinear tensor. We thereby obtain linear expressions for the camera rotations that are clear and simple, and do not suffer from inaccuracies in epipole's recovery.

2 The Homography Matrix

In this section we will briefly define the homography matrices, and prove that all homography matrices between two images form a linear space of rank 4. For more detailed information on homography matrices in 3D-from-2D geometry see [3, 21, 14, 6, 17, 13, 9, 11, 8], and for more details on the rank-4 result see [16].

*This research was sponsored by ARPA through the U.S. Office of Naval Research under grant N00014-93-1-1202, R&T Project Code 4424341—01.

†A. Shashua is with the Dept. of Computer Science, Technion, Haifa, Israel.

‡S. Peleg is visiting David Sarnoff Research Center, Princeton, NJ 08540.

Let P be a point in 3D space projecting onto images Ψ, Ψ'. Let $p \in \Psi$ and $p' \in \Psi'$ be the matching points in the two image planes described by:

$$p \cong M[I;0]P$$
$$p' \cong M'[R;T]P$$

where $\cong$ denotes equality up to scale, and M denotes the transformation from the observed coordinates of the image plane to the camera coordinate system, of the first camera, and M' of the second camera. In the simplest configuration, M is of the form:

$$M = \begin{bmatrix} f & 0 & x_o \\ 0 & f & y_o \\ 0 & 0 & 1 \end{bmatrix}$$

where f denotes the focal length of the camera and x_o, y_o is the origin of the image plane (known as the "principle point"). The rigid camera motion is represented by the rotation matrix R and translation vector T. Taken together, $P = (x, y, 1, 1/z)^\top$, where x, y are the image coordinates of the first view, z is the depth of the point, i.e., $zM^{-1}p$ are the Euclidean coordinates of the point P in the first camera frames, and:

$$p' \cong M'RM^{-1}p + \frac{1}{z}M'T.$$

When the points P live on a plane π, then $n^\top(zM^{-1}p) = d_\pi$ where n (normal vector) and d_π (scalar) are the parameters of π in the first camera coordinate system. We obtain,

$$\begin{aligned} p' &\cong M'RM^{-1}p + \frac{1}{z}\frac{1}{d_\pi}M'T(zn^\top M^{-1}p) \\ &= M'(R + \frac{1}{d_\pi}Tn^\top)M^{-1}p \\ &= H_\pi p. \end{aligned}$$

In other words, the homography matrix associated with π is

$$H_\pi \cong M'(R + \frac{Tn^\top}{d_\pi})M^{-1}. \tag{1}$$

Therefore, A homography H_π is a transformation associated with the two images Ψ and Ψ', and with the 3D plane π. For any point P in π, the homography H_π maps p to p' (see Fig. 1).

For a fixed pair of cameras (M, M', T and R are constant), given a homography matrix H_π of some 3D plane π, all other homography matrices can be described by

$$\lambda H_\pi + Tn^\top \tag{2}$$

for some scale factor λ and a normal to some plane n, since the homographies differ only in scale and in the plane parameters.

Consider homography matrices $H_1, H_2, ..., H_k$ each as a column vector in a $9 \times k$ matrix. Let $H_i = \lambda_i H_\pi + Tn_i{}^\top$. The following can be verified by inspection:

$$\begin{aligned} &\begin{bmatrix} & \\ & \end{bmatrix}_{9\times k} = [\lambda_1 H_\pi \cdots \lambda_k H_\pi]_{9\times k} + \\ &\begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}_{9\times 3} [n_1 \cdots n_k]_{3\times k} = \\ &= \begin{bmatrix} & T & 0 & 0 \\ H_\pi & 0 & T & 0 \\ & 0 & 0 & T \end{bmatrix}_{9\times 4} \begin{bmatrix} \lambda_1 \cdots \lambda_k \\ n_1 \cdots n_k \end{bmatrix}_{4\times k} \end{aligned} \tag{3}$$

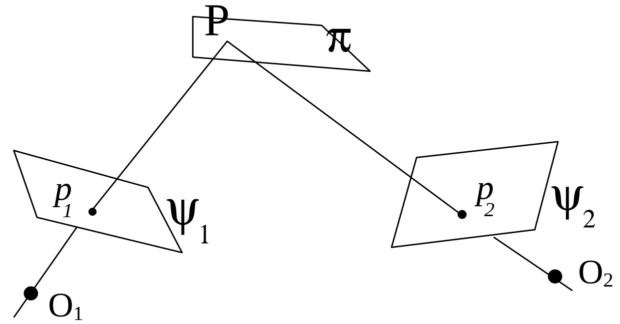

Figure 1: The homography induced by the plane π maps p to p'. p and p' are the perspective projections of any point P in the 3D plane π on the image planes Ψ and Ψ'.

We have thus proven that the space of all homography matrices between two fixed views is embedded in a 4 dimensional linear subspace of R^9.

3 The Rotation Matrix

Given a sequence of images taken by a camera moving in a static scene, we would like to recover the rotation parameters of the camera. The rotation matrix R is an orthonormal matrix (up to scale). The orthonormality is the only constraint on R, which can generate five non-linear constraints on the elements of R. Note also that $M'RM^{-1}$ is the homography of the plane at infinity (letting d_π go to infinity in eqn. 1). Since solving non-linear equations is in general less stable and harder to compute than linear equations, we will first examine the case of small rotations. In addition we will assume that $M = M' = I$, i.e., that the internal parameters of the cameras are known. We will show that in addition to working with linear constraints, the case of small rotations places a strong constraint on the family of admissible homographies: the only skew-symmetric homography matrix corresponds to the plane at infinity.

When small rotations are involved, the rotation matrix R can be approximated by a matrix having the following form (up to scale):

$$R \approx \hat{R} = \begin{bmatrix} 1 & \Omega_Z & -\Omega_Y \\ -\Omega_Z & 1 & \Omega_X \\ \Omega_Y & -\Omega_X & 1 \end{bmatrix}. \tag{4}$$

In this representation, the vector $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^\top$ is the rotation axis, and the magnitude of the vector is the magnitude of the rotation around this axis. Likewise, the family of all approximate homography matrices $\hat{H}_\pi$ is defined by:

$$H_\pi \approx \hat{H}_\pi \equiv \hat{R} + \frac{1}{d_\pi}Tn_\pi^\top, \tag{5}$$

where d_π is the distance from the origin to the plane π, $n_\pi^\top$ is the unit vector perpendicular to the plane toward the origin, and $T = (T_X, T_Y, T_Z)^\top$ is the camera's translation vector. Note that when $d_\pi \to \infty$, we have $\hat{H}_\pi = \hat{R}$. Therefore, the approximate rotation matrix $\hat{R}$ is also an approximate homography matrix of the plane at infinity.

There are two main advantages of this framework. First, as we shall see next, the non-linear constraints associated with orthonormal matrices is replaced with a skew-symmetric condition on the family of approximate homographies — which, moreover, leads to a unique choice for the appropriate rotation (unlike the general discrete case in which the space of homography matrices is represented by a 4 parameter family). Second, shown later, is that one can obtain a direct link between the trilinear tensor and the rotation matrix, thereby avoiding the computation of the translational component of camera motion (the epipolar geometry).

We show next that the asymmetric form of $\hat{R}$ is unique.

Lemma 1 *The only approximate homography matrix that has an asymmetric form*

$$\begin{bmatrix} r & u & -s \\ -u & r & t \\ s & -t & r \end{bmatrix} . \tag{6}$$

is the one associated with the plane at infinity.

Proof: if $\hat{H}_\pi$ has the asymmetric form (6), then $Tn_\pi^\top = d_\pi(\hat{H}_\pi - \hat{R})$ (using Equation 5) also has the asymmetric form (6), because $\hat{R}$ has this asymmetric form (Equation 4). However, the asymmetric form (6) has rank 3 (or rank 2 if $r = 0$), while the matrix $Tn_\pi^\top$ has rank 1, and we got a contradiction. $\square$

This Lemma together with the rank 4 result implies that the matrix $\hat{R}$ we are looking for is *the* asymmetric matrix spanned by four approximate homography matrices. In the following section we will take this result a step further and show that, first, only three homography matrices are required, and second, that $\hat{R}$ has a closed-form representation in terms of the coefficients of the trilinear tensor.

4 From Homographies to Rotations

We have shown (Section 2) that the space of all homography matrices between two images is a linear space of rank 4 (See also [16]). This holds also for the space of all approximate homographies (Equation 5), using the same proof. Given the rank-4 property we should find 4 linearly independent (approximate) homography matrices $\hat{E}_i, \quad i = 1..4$, between the two given images, and solve the equation:

$$\begin{bmatrix} 1 & \Omega_Z & -\Omega_Y \\ -\Omega_Z & 1 & \Omega_X \\ \Omega_Y & -\Omega_X & 1 \end{bmatrix} = \hat{R} = c_1\hat{E}_1 + c_2\hat{E}_2 + c_3\hat{E}_3 + c_4\hat{E}_4 \tag{7}$$

where $c_i, \quad i = 1..4$, are scalars. We should actually compute the $c_i, \quad i = 1..4$, that satisfy the asymmetric form constraints (six constraints).

We can reduce the need for 4 matrices to 3 by choosing the fourth matrix to be

$$\hat{E}_4 = \begin{bmatrix} T_x & 0 & 0 \\ T_y & 0 & 0 \\ T_z & 0 & 0 \end{bmatrix} , \tag{8}$$

where $T = (T_x, T_y, T_z)^\top$ is the translation vector from Ψ to Ψ', which is unknown yet. Furthermore, we can set $c_4 = 1$ because the translation vector is defined up to scale, and so does $\hat{E}_4$.

Lemma 2 *The matrix $\hat{E}_4$ is a valid homography matrix from Ψ to Ψ'.*

Proof: Given the rotation matrix $\hat{R}$ and a translation vector T, the homography matrix from Ψ to Ψ' can be defined by $\hat{H}_\pi = d_\pi(\hat{R} + \frac{1}{d_\pi}Tn_\pi^\top)$ for any value of the plane parameters d_π and n_π (Equation 5). When using this equation for the plane having $n^\top = (1, 0, 0)$ at the limit of $d_\pi \rightarrow 0$ we get

$$\lim_{d_\pi \rightarrow 0} \hat{H}_\pi = Tn_\pi^\top = \begin{bmatrix} T_x & 0 & 0 \\ T_y & 0 & 0 \\ T_z & 0 & 0 \end{bmatrix} , \tag{9}$$

which is the matrix $\hat{E}_4$. $\square$

To solve Equation (7), with $\hat{E}_4$ defined in Equation (8), we need to solve for the six unknowns: $c_1, c_2, c_3, T_x, T_y, T_z$, using the six constraints on the asymmetric form of $\hat{R}$:

$$\begin{bmatrix} 1 & \Omega_Z & -\Omega_Y \\ -\Omega_Z & 1 & \Omega_X \\ \Omega_Y & -\Omega_X & 1 \end{bmatrix} = \hat{R} =$$

$$c_1\hat{E}_1 + c_2\hat{E}_2 + c_3\hat{E}_3 + \begin{bmatrix} T_x & 0 & 0 \\ T_y & 0 & 0 \\ T_z & 0 & 0 \end{bmatrix} . \tag{10}$$

Any three homography matrices recovered in any known method can be used to recover the camera rotation (assuming small angle rotations). Our method of choice, however, is the trilinear tensor computed for three images from point correspondences [15, 18, 5]. There are two advantages for this choice. First, the tensor provides directly three homography matrices [18], hence as a technique for producing homography matrices one is exploiting three instead of two views (thereby gaining numerical redundancy) and also no assumption on scene structure is being made (i.e., that the scene contains dominant and distinct physical planes). Second, we will show that there is a closed-form solution for $\hat{R}$ as a function of the tensor coefficients, thereby obtaining a linear method that uses all the matching points across three views (for computing the tensor), does not require solving for the translational component of camera motion as an intermediate stage, and is very simple.

The trilinear tensor is described in Appendix A. The tensor α_i^{jk} contains 27 entries (coefficients) as $i, j, k = 1, 2, 3$. The coefficients can be recovered linearly from at least 7 matching points across three views. The connection between the tensor and homography matrices comes from contraction properties as follows: for any vector $s_j = (s_1, s_2, s_3)$, the matrix $s_j\alpha_i^{jk}$ is a homography matrix from view 1 to view 2, where s describes the orientation of the associated plane (similarly, $s_k\alpha_i^{jk}$ is a homography matrix from view 1 to view 3). In particular, when $s = (1, 0, 0), (0, 1, 0)$ and $(0, 0, 1)$ we get our three independent homography matrices [18].

In the case of small rotations we can enforce the constraints on the rotation $\hat{R}$ on the homography matrices recovered from the tensor. Using Maple for symbolically solving for the system of equations we found that the translational component T_x, T_y, T_z have been eliminated from the result, and we are left with a very simple, closed-form, expression relating the tensor α_i^{jk} and $\Omega_X, \Omega_Y, \Omega_Z$:

$$
\begin{aligned}
\Omega_X &= det \begin{pmatrix} \alpha_2^{j3} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
\Omega_Y &= det \begin{pmatrix} -\alpha_1^{j3} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
\Omega_Z &= det \begin{pmatrix} \alpha_1^{j2} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix} / K \\
K &= det \begin{pmatrix} \alpha_2^{j2} \\ \alpha_2^{j3} + \alpha_3^{j2} \\ \alpha_3^{j3} - \alpha_2^{j2} \end{pmatrix}
\end{aligned}
\tag{11}
$$

where α_2^{j2} stands for (α_2^{12}, α_2^{22}, α_2^{32}), etc. This expression recovers directly and simply small rotations from the trilinear tensor.

5 Video Stabilization

In this section we present the algorithm for video stabilization which contains two steps. The first step is to compute the trilinear tensors, and the second step is to compute the camera rotations and perform derotation on the frames.

5.1 Trilinear Tensor Computation

Following are the steps we performed to compute the trilinear tensor from a set of three images. For computation stability, all coordinates are normalized to the range of (-1, 1).

- **Selection of corresponding points**
  Optical flow is computed between all three possible pairs of the three images. As corresponding points we selects only those points having a high gradient, and for which all the pairwise optical flows are consistent. This process results in the selection of several hundred points as corresponding in all three frames ("corresponding triplets").
- **Robust Estimation**
  From all corresponding triplets computed in the previous step, several hundred subsets of ten triplets are randomly selected [20]. For each subset the trilinear tensor is computed using Eq. 14. Each computed tensor is then applied to all matching triplets, and the single tensor for which the maximal number of triplets satisfy Eq. 14 is selected.
- **Least Square Step**
  As a final step we use all the points which satisfied the selected tensor in the previous step to solve the tensor again from Eq. 14 using a least squares method. From this tensor the homography matrices will be computed.

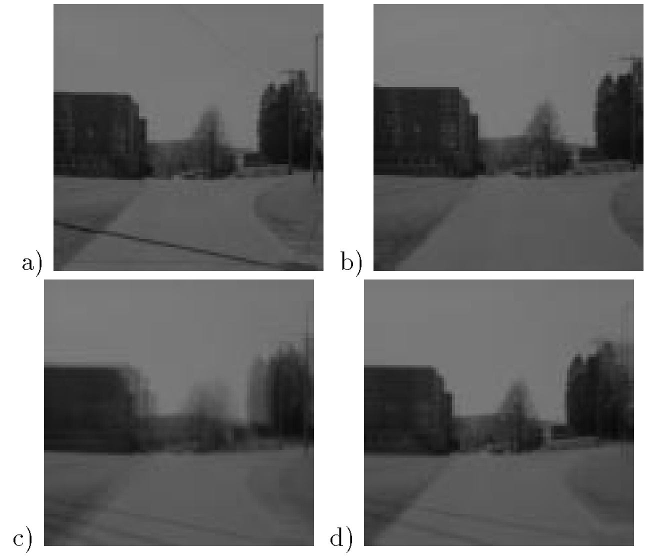

Figure 2: First sequence - outdoor scene.
a) First original frame.
b) Last (fourth) original frame. The camera was rotating while moving forward.
c) average of the four original images.
d) average of the four images after rotation cancellation. The remaining motion is only due to the original translation. The sequence looks as if it was taken using a stabilized camera.

5.2 Derotation of Video Frames

The image sequence is stabilized with regard to the first image. This is done by selecting an arbitrary image to serve as the third image and sequentially going through the images. For each such triplet of images the trilinear tensor is computed as described above. From the compute tensor we compute $\Omega_X, \Omega_Y, \Omega_Z$ using Eq. 11. The image is then warped back to cancel the rotation, thus getting a stable sequence.

6 Experimental Results: Stabilization

Given a sequence of images, a trilinear tensor is recovered from the first frame to all other frames, using an arbitrary frame as the third frame. The rotation from the first frame to all other images is then recovered by using the the tensor values in Eq. 11. By warping back every image using the calculated rotation, we obtain a new sequence of images having no rotation compared to the first image. The remaining motion in the new sequence is only due to the original translation, thus the new sequence is smooth and clear, as can be observed in the average images.

The method was tested on outdoor scene (Fig. 2), indoor scene (Fig. 4), and on a scene with objects that are very close to the camera (Fig. 3). The method proved to be robust and efficient.

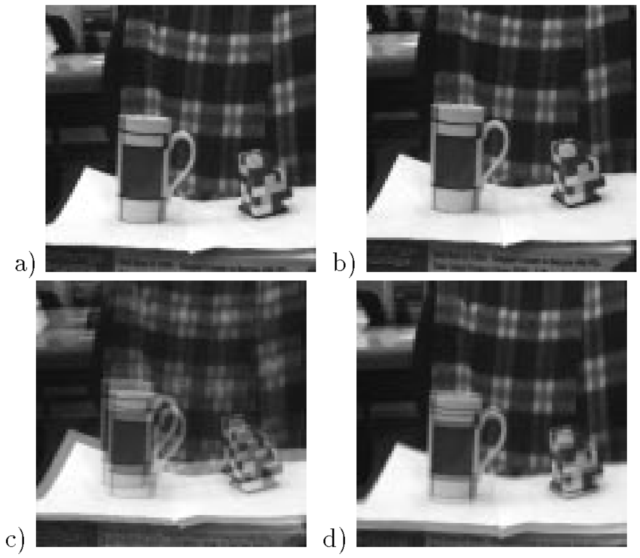

Figure 3: Second sequence - close objects.
a) First original frame.
b) Second original frame. The camera was moving and rotating around the objects.
c) average of the two original images.
d) average of the two images after rotation cancelation. The remaining motion is only due to the original translation.

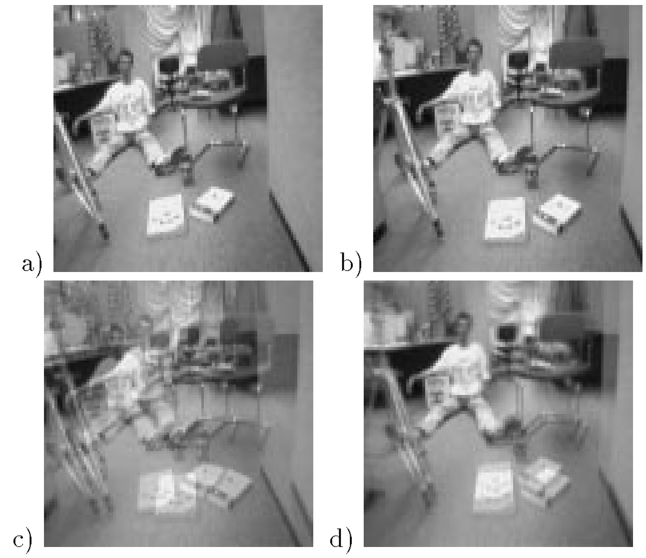

Figure 4: Third sequence - indoor scene.
a) First original frame.
b) Second original frame. The unstable camera was moving towards the man.
c) average of the two original images. The rotations make the average image unclear.
d) average of the two images after rotation cancellation. The remaining motion is only due to the original translation. The sequence looks as if it was taken using a stabilized camera.

7 Concluding Remarks

A new robust method to recover the rotation of the camera was described. The main contribution to the robustness is the fact that we do not have to recover the epipoles, and the rotation is computed *directly* from three homography matrices assuming small rotations. The homography matrices are obtained from three images using the trilinear tensor parameters, and the recovery process does not assume any 3D model. The method can be extended to handle also the case of general rotations by using iterations.

A Appendix: The Trilinear tensor

In this section we briefly present the trilinear tensor and give an example to measure its quality. The trilinear tensor is an extension of the fundamental matrix to the case of three images, and it describes the spatial relation of three cameras. The quality of the tensor can be evaluated by reprojecting the third image from the first two images using the trilinear tensor. More details about the trilinear tensor can be found in [15, 18].

Let P be a point in 3D projective space projecting onto p, p', p'' three views Ψ, Ψ', Ψ'' represented by the two dimensional projective space. The relationship between the 3D and the 2D spaces is represented by the 3×4 matrices, $[I, 0]$, $[A, v']$ and $[B, v'']$, i.e.,

$$p = [I, 0]P$$
$$p' \cong [A, v']P$$
$$p'' \cong [B, v'']P$$

We may adopt the convention that $p = (x, y, 1)^\top$, $p' = (x', y', 1)^\top$ and $p'' = (x'', y'', 1)^\top$, and therefore $P = [x, y, 1, \rho]$. The coordinates $(x, y), (x'y'), (x'', y'')$ are matching points (with respect to some arbitrary image origin — say the geometric center of each image plane). The matrices A and B homography matrices from Ψ to Ψ' and Ψ'', respectively, induced by *some* plane in space (the plane $\rho = 0$). The vectors v' and v'' are known as epipolar points (the projection of O, the center of projection of the first camera, onto views Ψ' and Ψ'', respectively).

The trilinear tensor is an array of 27 entries:

$$\alpha_i^{jk} = v'^k b_i^j - v''^j a_i^k. \qquad i, j, k = 1, 2, 3 \tag{12}$$

where superscripts denote contravariant indices (representing points in the 2D plane, like v') and subscripts denote covariant indices (representing lines in the 2D plane, like the rows of A). Thus, a_i^k is the element of the k'th row and i'th column of A, and v'^k is the k'th element of v'. The tensor α_i^{jk} forms the set of coefficients of certain trilinear forms that vanish on any corresponding triplet p, p', p'' (i.e., functions of views that are invariant to object structure). These functions have the following form: let s_k^l be the matrix,

$$s = \begin{bmatrix} 1 & 0 & -x' \\ 0 & 1 & -y' \end{bmatrix}$$

and, similarly, let r_j^m be the matrix,

$$r = \begin{bmatrix} 1 & 0 & -x'' \\ 0 & 1 & -y'' \end{bmatrix}$$

Then, the tensorial equations are:

$$s_k^l r_j^m p^i \alpha_i^{jk} = 0, \tag{13}$$

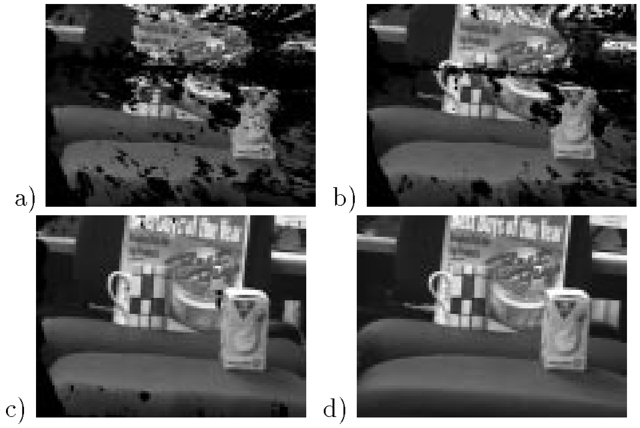

Figure 5: Reprojection in different methods.
a) Reprojection using epipolar line intersection. Fundamental Matrices computed with code distributed by INRIA.
b) Reprojection using epipolar line intersection. Fundamental Matrices computed from tensor.
c) Reprojection using the tensor equations.
d) Original third image. Presented for comparison.

with the standard summation convention that an index that appears as a subscript and superscript is summed over (known as a contraction). For details on the derivation of this equation see Appendix A. Hence, we have four trilinear equations (note that $l, m = 1, 2$). In more explicit form, these functions (referred to as "trilinearities") are:

$$\begin{aligned}
x''\alpha_i^{13}p^i - x''x'\alpha_i^{33}p^i + x'\alpha_i^{31}p^i - \alpha_i^{11}p^i &= 0, \\
y''\alpha_i^{13}p^i - y''x'\alpha_i^{33}p^i + x'\alpha_i^{32}p^i - \alpha_i^{12}p^i &= 0, \\
x''\alpha_i^{23}p^i - x''y'\alpha_i^{33}p^i + y'\alpha_i^{31}p^i - \alpha_i^{21}p^i &= 0, \\
y''\alpha_i^{23}p^i - y''y'\alpha_i^{33}p^i + y'\alpha_i^{32}p^i - \alpha_i^{22}p^i &= 0.
\end{aligned}$$

Since every corresponding triplet p, p', p'' contributes four linearly independent equations, then seven corresponding points across the three views uniquely determine (up to scale) the tensor α_i^{jk}. More details and applications can be found in [15]. Also worth noting is that these trilinear equations are an extension of the three equations derived by [19] under the context of unifying line and point geometry.

The connection between the tensor and homography matrices comes from contraction properties described in Section 4, and from the homography matrices one can obtain the "fundamental" matrix F (the tensor produces 18 linear equations of rank 8 for F, for details see [18]). Fig. 5 shows an example of image reprojection (transfer) using the trilinearities, compared to using the epipolar geometry (recovered using INRIA code or using F recovered from the tensor). One can see that the best results are obtained from the trilinearities directly.

References

[1] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, Santa Margarita Ligure, May 1992.

[2] O.D. Faugeras. *Three-Dimensional Computer Vision, A Geometric Viewpoint*. MIT Press, 1993.

[3] O.D. Faugeras and F. Lustman. Let us suppose that the world is piecewise planar. In O. D. Faugeras and Georges Giralt, editors, *International Symposium on Robotics Research*, pages 33–40. MIT Press, Cambridge, MA, 1986.

[4] O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *International Conference on Computer Vision*, pages 951–956, Cambridge, MA, June 1995.

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

[6] R. Hartley and R. Gupta. Computing matched-epipolar projections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 549–555, New York, NY, 1993.

[7] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, Santa Margarita Ligure, May 1992.

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

[9] R. Kumar and P. Anandan. Direct recovery of shape from multiple views: A parallax based approach. In *International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994.

[10] J.M. Lawn and R. Cipolla. Robust egomotion estimation from affine motion parallax. In *European Conference on Computer Vision*, pages 205–210, May 1994.

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

[12] S. Negahdaripour and S. Lee. Motion recovery from image sequences using first-order optical flow information. In *IEEE Workshop on Visual Motion*, pages 132–139, Princeton, NJ, October 1991.

[13] Harpreet Sawhney. 3d geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[14] A. Shashua. Projective structure from uncalibrated images: structure from motion and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(8):778–790, 1994.

[15] A. Shashua. Algebraic functions for recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:779–789, 1995.

[16] A. Shashua and S. Avidan. The rank4 constraint in multiple view geometry. In *European Conference on Computer Vision*, Cambridge, UK, April 1996. Also in CIS report #9520, November 1995, Technion.

[17] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Washington, 1994.

[18] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *International Conference on Computer Vision*, pages 920–925, Cambridge, MA, 1995.

[19] M.E. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *ARPA IU Workshop*, 1990.

[20] P.H.S Torr, A. Zisserman, and D. Murray. Motion clustering using the trilinear constraint over three views. In *Workshop on Geometrical Modeling and Invariants for Computer Vision*. Xidian University Press., 1995.

[21] R. Tsai and T.S. Huang. Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition. *IEEE Trans. on Acoustic, Speech and Signal Processing*, 30, 1982.

[22] A. Zisserman and S. Maybank. A case againt epipolar geometry. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, pages 35–50, Ponta Delgada, Azores, 1994. Springer.