Join Tensors: on 3D-to-3D alignment of Dynamic Sets

Lior Wolf, Amnon Shashua
School of Computer Science and Engineering,
The Hebrew University,
Jerusalem 91904, Israel
http://www.cs.huji.ac.il/∼{lwolf,shashua}

Yoni Wexler
Center for Automation Research
University of Maryland
College Park, MD 20742
http://www.cs.umd.edu/∼wexler

Abstract

This paper introduces a family of $4 \times 4 \times 4$ *tensors, referred to as "join tensors" or Jtensors for short, which perform "3D to 3D" alignment between coordinate systems of sets of* dynamic *3D points. 3D Configurations of points are obtained by a 3D measuring device (such as a structured light or laser range sensor, or a stereo rig) at times* t_1, t_2, t_3 *from different viewing positions in addition to the motion of the sensor the points are also allowed to move in space; each point can move along an arbitrary straight-line path — we refer to this situation as "dynamic". The problem is to recover the motion of the sensor given the 3D correspondences of the points over time.*

We introduce Jtensors to capture the problem described above. Three observations P, P', P'' *of a point measured at three time instants contribute a linear measurement to the Jtensor, regardless of whether the point has moved in space or has remained stationary while the sensor has changed position.*

1 Introduction

Consider the classic problem of "3D to 3D" alignment of point sets. We are given a set of 3D points $P_1, ..., P_n$ measured by some device such as a structured light range sensor [**?**] or a stereo rig of cameras. When the sensor changes its position in space while the 3D points remain stationary, the 3D positions of the measured points $P'_1, ..., P'_n$, have undergone a coordinate transformation. In a projective setting, five of these matching pairs in general position are sufficient to recover the 4×4 collineation A such that $AP_i \cong P'_i$, $i = 1, ..., n$. In a rigid motion setting the coordinate transformation consists of translation and rotation which can be recovered using four matching points; elegant techniques using SVD have been developed for this purpose [2].

In this paper we introduce a "dynamic" version of the 3D-to-3D alignment problem (see Fig. 1). We allow for the possibility that any number of the points may move along straight-line paths during the motion of the sensor. Points that remain in place are called *static* and points that move are called *dynamic*. There can be any number of dynamic points — including the possibility that *all* points are dynamic — and the system need not know in advance which points are static and which are dynamic (an unsegmented configuration). Under these conditions we wish to find the projective coordinate changes under *two* motions of the sensor.

We derive a $4 \times 4 \times 4$ family of tensors, referred to as *join tensors* or Jtensors in short, that capture the dynamic 3D-to-3D alignment problem. A triplet P, P', P'' of positions of some point measured at three time instants contributes a linear measurement to the Jtensor regardless of whether the physical point in space is dynamic or static while the sensor has changed positions. The linear constraints add up to a four-dimensional null space of Jtensors — that is, there exist four distinct Jtensors which are linearly recovered from matching points. We will show how the coordinate transformations are extracted from the Jtensors, how the mapping between coordinate systems is done directly using the Jtensors, and what can be said about combining static and dynamic observations in unsegmented and segmented situations.

2 Derivation of Jtensors

Let X be some point in 3D space with coordinate vector P. Let P' be the coordinate representation of the point X at some other time instant (the sensor has changed its viewing position) and let P'' be the coordinate representation of X at a third time instant. Let A, B be the collineations mapping the second and third coordinate representations back to the first representation, i.e., $P \cong AP'$ and $P'' \cong BP''$.

If the point X happens to *move* along some straight-line path during the change of coordinate systems, then P, AP', BP'' do not coincide but they form a rank-2 ma-

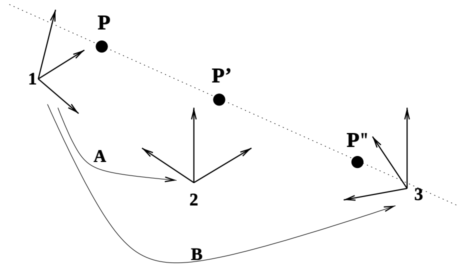


Figure 1. The points P,P' and P'' are measured at three time instants from different viewing positions of the sensor, i.e., each point is given in a different coordinate system. While the measuring device changes position, the physical point in space moves along a straight line path. In other words, the rank of the 4×3 matrix $[P, AP', BP'']$ is 2 for a moving point and 1 for a static point. The 4×4 matrices A, B are responsible for the change of coordinate system back to the starting position.

trix:

$$rank \begin{pmatrix} | & | & | \\ P & AP' & BP'' \\ | & | & | \end{pmatrix} = 2$$

And for every column vector V we have

$$\det \begin{pmatrix} | & | & | & | \\ P & AP' & BP'' & V \\ | & | & | & | \end{pmatrix} = 0 \qquad (1)$$

Note that because V is spanned by a basis of size four, we can obtain at most four linearly independent constraints on some object consisting of A, B from a triplet of matching points P, P', P''. Note also that the null vector of a 4×3 matrix can be represented by the 3×3 determinant expansion. For example, let X, Y, Z be three column vectors in a 4×3 matrix, then the vector W representing the plane defined by the points X, Y, Z is

$$w_1 = \det \begin{pmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{pmatrix} \quad w2 = -\det \begin{pmatrix} x_1 & y_1 & z_1 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{pmatrix}$$

$$w_3 = \det \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_4 & y_4 & z_4 \end{pmatrix} \quad w_4 = -\det \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}$$

We can write the relationship between W and X, Y, Z as a tensor operation as follows:

$$w_i = \epsilon_{ijkl} x^j y^k z^l$$

where the entries of ϵ consist of $+1, -1, 0$ in the appropriate places. We will refer to ϵ as the "cross-product" tensor. Note that the determinant of a 4×4 matrix whose columns consist of $[X, Y, Z, T]$ can be compactly written as

$$t^i x^j y^k z^l \epsilon_{ijkl}.$$

Using the cross-product tensor we can write the constraint (1) as follows:

$$\begin{aligned} 0 &= \det \begin{pmatrix} | & | & | & | \\ P & AP' & BP'' & V \\ | & | & | & | \end{pmatrix} \\ &= P^i(\epsilon_{ilmu}(A^l_j P'^j)(B^m_k P''^k)V^u) \\ &= P^i P'^j P''^k(\epsilon_{ilmu} A^l_j B^m_k V^u) \end{aligned}$$

Note that the tensor form allows us to separate the measurements P, P', P'' from the unknowns A, B, and we denote the expression in parentheses

$$\mathcal{J}_{ijk} = \epsilon_{ilmu} A'^l_j B'^m_k V^u \qquad (2)$$

as the "join"[1] tensor, or Jtensor for short. Note that for every choice of the vector V we get a Jtensor. As previously mentioned, since V is spanned by a basis of dimension four there are four such tensors; each tensor is defined by the constraints

$$P^i P'^j P''^k \mathcal{J}_{ijk} = 0.$$

These are linear constraints on the 64 elements of the Jtensor. Because there are four Jtensors the linear system of equations for solving for $\mathcal{J}_{ijk}$ from the matching triplets P, P', P'' has a four-dimensional null space. The vectors of the null space are spanned by the Jtensors. In practical terms, given $N \geq 60$ matching triplets P, P', P'', each triplet contributes one linear equation $P^i P'^j P''^k \mathcal{J}_{ijk} = 0$ for the 64 entries of $\mathcal{J}_{ijk}$. The eigenvectors associated with the four smallest eigenvalues of the estimation matrix are *the Jtensors of the dynamic 3D-to-3D alignment problem.* We summarize this in the following theorem:

Theorem 1 (Jtensors) *Each matching triplet P, P', P'' arising from a dynamic or static point contributes one linear equation $P^i P'^j P''^k \mathcal{J}_{ijk} = 0$ to a $4 \times 4 \times 4$ tensor $\mathcal{J}_{ijk}$. Any $N \geq 60$ matching triplets in general position provide an estimation matrix for $\mathcal{J}_{ijk}$ with a four-dimensional null space. Therefore, the eigenvectors associated with the smallest four eigenvalues of the estimation matrix define four tensors that agree with the measurements.*

We see that at least 60 point measurements are needed for a solution to the Jtensors. If all of the measurements arise from dynamic points, these points should be distributed along at least ten lines, five of which can hold up

[1] The join operator is the exterior product of the Grassmann-Cayley algebra. A join of three 3D points is a plane which contains the three points.

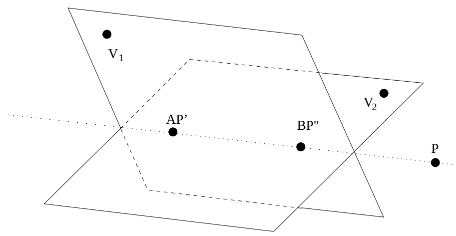


Figure 2. The points AP',BP'' and V define a plane. AP',BP'' and V' define another plane. The line of intersection of these planes contains P.

to eight dynamic points, and the remaining five up to four dynamic points. We will not prove this statement here. We will state the following results (for proof see [7]):

Theorem 2 *The constraints* $P^iP'^jP''^k\mathcal{J}_{ijk} = 0$ *made solely from static points span at most a 20-dimensional space.*

Theorem 3 *Out of the ten linearly independent constraints arising from a labeled static point, four lie in the rank-20 subspace spanned by unlabeled static points and six lie in the subspace spanned only by dynamic points.*

In other words, the first five labeled static points contribute 50 constraints (4 + 6 from each point), whereas each additional labeled static point contributes only six additional constraints (because the remaining four are already spanned by the 20-constraints spanning the 20-dimensional space generated by the static points). Consequently, one needs at least seven labeled static points for a unique solution for the Jtensors. In a mixed segmented and unsegmented situation, each labeled static point reduces the number of required dynamic points by 6. Thus, for example, if only one point is labeled as static, we need at least 50 additional (unlabeled) matching triplets out of which at most 16 can be static.

We will next investigate the tensor slices and the extraction of the constituent collineations A, B from the four Jtensors.

3 Tensor Slices and the Extraction of A, B

The tensor $\mathcal{J}_{ijk}$ is symmetric with respect to the position of the points P, P', P'' (this is true for every purely covariant or contravariant tensor, unlike the mixed covariant-contravariant trifocal tensor). It is therefore sufficient to investigate $P'^jP''^k\mathcal{J}_{ijk}$ as one of the tensor double-contractions; the others, $P^iP''^k\mathcal{J}_{ijk}$ and $P^iP'^j\mathcal{J}_{ijk}$, follow by symmetry.

Consider any Jtensor with its associated vector V. We will refer to V as the *principal point* of the tensor; as described next. Clearly, $\pi_i = P'^jP''^k\mathcal{J}_{ijk}$ is a plane (because of the index position). The plane π is defined by the three points V, AP' and BP'' because

$$P'^jP''^k\mathcal{J}_{ijk} = \epsilon_{ilmu}(A^l_jP'^j)(B^m_kP''^k)V^u,$$

which by definition of the cross-product tensor provides the plane associated with the three points acted upon by ϵ. Therefore by varying P' and P'' we obtain a star of planes all coincident with the point V. Therefore the principal point of the tensor can be recovered by taking three double slices of the tensor and finding their intersection.

To recover the line in space passing through AP' and BP'' it is necessary to take two Jtensors. The intersection of the planes $P'^jP''^k\mathcal{J}^1_{ijk}$ and $P'^jP''^k\mathcal{J}^2_{ijk}$ is the line passing through AP' and BP'' (see Fig. 2).

A single contraction $P''^k\mathcal{J}_{ijk}$ is a 4×4 matrix H_{ij} that maps points to planes. P'^jH_{ij} is the plane passing through V, AP', BP''; thus by varying P' one obtains a pencil of planes coincident with the line through V and BP''. Hence the rank of the matrix H must be 2.

Because HP' is the plane through V, AP', BP'', we have $P'^\top A^\top HP' = 0$ for every choice of P'. Therefore $A^\top H$ is a skew-symmetric matrix and thus provides ten linear constraints for A. By varying P'' and thus obtaining other H-matrices $P''^k\mathcal{J}_{ijk}$ we can obtain more constraints on A but this is not sufficient to obtain a unique solution for A. A unique solution requires the H-matrix of at least another Jtensor because the principal point must vary as well. Likewise, one can recover B from the contractions $P'^j\mathcal{J}_{ijk}$ by varying P' and taking at least two Jtensors.

4 Applications

Consider the problem of 3D reconstruction of an object which extends beyond the field of view of the sensor. For this purpose we can use a stereo rig, that contains a texture pattern projector for obtaining matching points on textureless areas of the object. Because the field of view of the cameras does not cover the entire object, the stereo rig must acquire images from multiple viewing positions. Each image provides a 3D patch of the object and the goal is to “stitch” these patches together by aligning their coordinate systems. In other words, we must recover the relative 3D motion of the rig. In this context, the dynamic points are the points arising from the projected texture and the static points arise from texture markings on the object’s surface. Hence, if the rig moves in a piecewise straight-line path and the object is polyhedral, Jtensor theory is an appropriate tool for aligning the coordinate systems of the 3D patches.

In the experiment, the sensor is a stereo rig with two cameras and a random texture projector. The stereo rig acquires three pairs of textured images while moving approximately on a straight line. We use the Lucas-Kanade optical flow algorithm [3] for tracking, both between and across the pairs. In each pair we compute the projective structure [5, 1] of the points separately. The Jtensor is computed from the reconstructed projective structure using Least Median of Squares [4]. The motion of the rig (matrices A and B) is then recovered and is used for filtering out the static feature points — a task whose degree of success provides an indication to the usefulness of the algorithm because once the static features have been identified one can then resort to classic 3D-to-3D alignment methods (cf. [2]). Fig. 3 shows the results.

5 Summary

We have presented a tool for relating dynamic sets of points in 3D across three reference frames. The dynamic points may move along straight lines and may be measured at three arbitrary coordinate frames. The resulting tensors (Jtensors) are the 3D extension of the recently discovered "homography" tensors [6] applicable for 2D scenes viewed by moving camera.

We have demonstrated that a possible application of the Jtensor theory is dynamic alignment of 3D point sets resulting from a moving stereo rig with an attached random texture projector. The Jtensor alleviates the need to mark and segment static points on the object. This leads to an automatic method of 3D reconstruction from multiple stereo rigs without the need for a fixed source of texture projection.

[1] 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.

[2] G. Golub and C. V. Loan. *Matrix computations*. John Hopkins University Press, 1989.

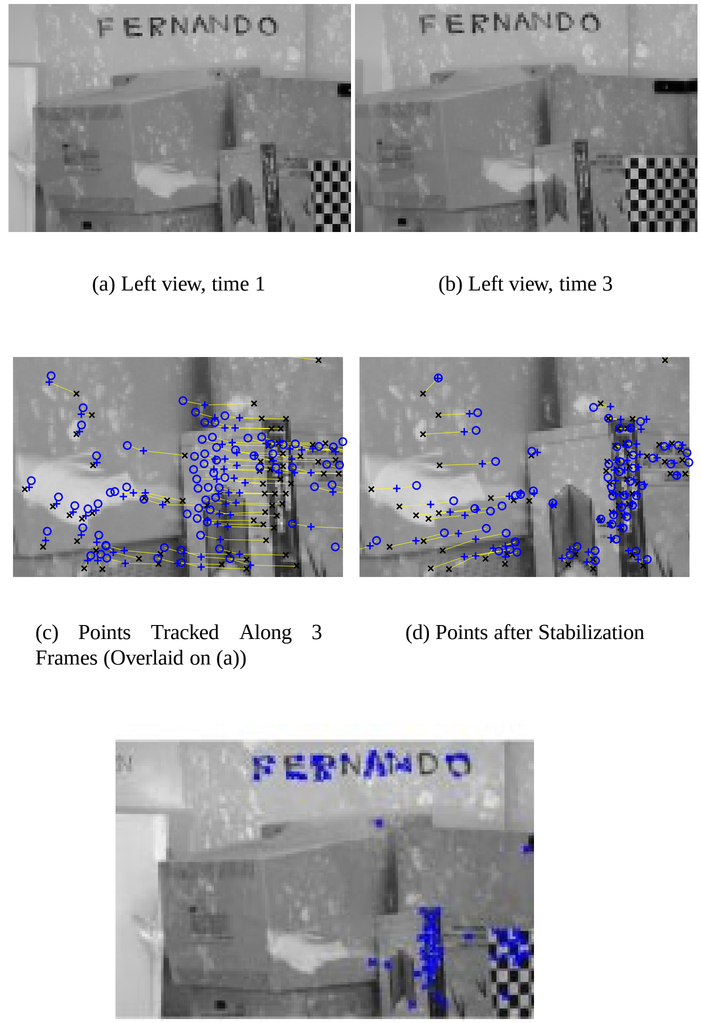
[3] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings IJCAI*, pages 674–679, Vancouver, Canada, 1981.

[4] P. Meer, D. Mintz, D. Kim, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

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

[6] A. Shashua and L. Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Dublin, Ireland, June 2000.

[7] L. Wolf, A. Shashua, and Y. Wexler. Join tensors: on 3d-to-3d alignment of dynamic sets. Technical Report CSE00-3, School of CSE, Hebrew University of Jerusalem, January 2000.



(a) Left view, time 1

(b) Left view, time 3

(c) Points Tracked Along 3 Frames (Overlaid on (a))

(d) Points after Stabilization

(e) Segmentation of moving/static points

Figure 3. Application of the Jtensor to 3D reconstruction. Row 1 displays the left image of the stereo pair at times 1 and 3. Note that the texture is partly projected (dynamic) and partly surface markings (static). Row 2 displays the tracked points before and after stabilization (motion of the rig canceled out). Note that the static points were stabilized, indicating that the Jtensor captured the correct 3D motion. *(e)* shows the static points which were identified by the Jtensor (see text).