Lines and Points in Three Views - An Integrated Approach

Richard I. Hartley

G.E. CRD, Schenectady, NY, 12301.
LIFIA Grenoble

Abstract

This note outlines a new method for carrying out a projective reconstruction of a scene from three views of a mixture of lines and points. The algorithm is non-iterative (in fact linear) consequently very rapid. A particular case of this algorithm gives a new linear method for computing a scene from seven points in three views.

1 Introduction

In a previous paper ([3, 4]) I gave a method for doing projective reconstruction from a set of 13 or more lines seen in three views. This method was demonstrated to be relatively stable in [4], provided sufficient care is taken with the implementation. Meanwhile, in [7] a method was given for computing a novel view of an image from two reference images. This method uses certain "trilinearity conditions" to transfer points into the new image. In that paper, a linear method was given for computing the coefficients of the trilinear equations from just seven points in two images. This result implied the possibility of a linear method for carrying out a projective reconstruction of a set of points seen in three views. How to extend Shashua's method to obtain a projective reconstruction was not entirely obvious. Shashua's derivation of his method uses somewhat unfamiliar concepts and terminology.

In this note, the results of Shashua are rederived using more standard means. As a result, a method for computing the projective structure using linear techniques is derived. Along the way, an unexpected connection appears between the 7-point algorithm of Shashua and the 13-line algorithm of [3, 4]. By merging the two techniques one obtains an algorithm that works for a mixed set of lines and points. This new algorithm integrates both lines and points into one linear framework. This contrasts with previously known algorithms that have been specific to either point sets (for example the 8-point algorithm of [5, 6]) or line sets ([3, 4]). Previously to deal with both points and lines at the same time, an iterative least-squares approach has been necessary.

The new algorithm has not to date been tried on mixed sets of points and lines. However, the results obtained in ([3, 4]) for lines and in [7] using points, both special cases of this algorithm show quite good results.

Thanks are due to Long Quan and Andrew Zisserman for discussions before and during the preparation of this note.

2 Notation

All vectors are written as lower case bold letters, such as $\mathbf{u}$ or $\boldsymbol{\lambda}$. Such a notation denotes a column vector. A row vector is written as a transposed column vector such as $\mathbf{u}^\top$. The inner product of two vectors is written as $\mathbf{a}^\top\mathbf{b}$, that is by viewing a vectors as a matrix with only one column, and carrying out a matrix multiplication. Similarly a product $\mathbf{a}\mathbf{b}^\top$ represents a matrix.

The symbol $\approx$ is used to denote equality up to multiplication by a non-zero scale factor.

3 Derivation of Shashua's Trilinearity Conditions

Suppose that a point $\mathbf{x}$ in space is seen in three images, and that the three cameras are given in the normalized form $P = (I \mid 0)$, $P' = (A \mid \mathbf{p}')$ and $P'' = (B \mid \mathbf{p}'')$. We may assume that this is the case, since the relationships that will be derived are independent of projective transformations of space, and hence we may apply an arbitrary projective transformation. This transformation may be chosen in such a way as to transform the first camera matrix to the desired normalized form $(I \mid 0)$.

We suppose that the point $\mathbf{x}$ is seen at positions $\mathbf{u}$, $\mathbf{u}'$ and $\mathbf{u}''$ in the three images, where $\mathbf{u}$ (and similarly $\mathbf{u}'$ and $\mathbf{u}''$) is a 3-vector $\mathbf{u} = (u, v, 1)$, the representation of the point in homogeneous coordinates. The coordinates (u, v) are the coordinates actually seen in the image. We wish to find a relationship between the coordinates of the points $\mathbf{u}$, $\mathbf{u}'$ and $\mathbf{u}''$.

Because of the form of the matrix $P = (I \mid 0)$, it is extremely simple to give a formula for the position of the point in space. In particular, since $(I \mid 0)\mathbf{x} \approx \mathbf{u}$, we may write $\mathbf{x} = \begin{pmatrix} \mathbf{u} \\ t \end{pmatrix}$ for some t, yet to be determined. It may be verified that t is the same as the "relative affine invariant", k, considered by Shashua ([7]). Now, projecting this point into the second image, we see that

$$\mathbf{u}' \approx P'\mathbf{x} = (A \mid \mathbf{p}')\begin{pmatrix} \mathbf{u} \\ t \end{pmatrix}$$

Denoting the i-th row of A by ${\mathbf{a}_i}^\top$, we may write this equation as three separate equations

$$\begin{array}{rcl} u'w' & = & {\mathbf{a}_1}^\top\mathbf{u} + p'_1 t \\ v'w' & = & {\mathbf{a}_2}^\top\mathbf{u} + p'_2 t \\ w' & = & {\mathbf{a}_3}^\top\mathbf{u} + p'_3 t \end{array} \tag{1}$$

where w' is an unknown scale factor. We may eliminate w' to obtain two separate equations

$$\begin{array}{rcl} u'({\mathbf{a}_3}^\top\mathbf{u} + p'_3 t) & = & {\mathbf{a}_1}^\top\mathbf{u} + p'_1 t \\ v'({\mathbf{a}_3}^\top\mathbf{u} + p'_3 t) & = & {\mathbf{a}_2}^\top\mathbf{u} + p'_2 t \end{array} \tag{2}$$

From each of these equations independently, one may compute the value of t. We obtain

$$\begin{aligned} t &= \frac{\mathbf{a}_1{}^\top \mathbf{u} - u'\mathbf{a}_3{}^\top \mathbf{u}}{u'p'_3 - p'_1} \\ &= \frac{\mathbf{a}_2{}^\top \mathbf{u} - v'\mathbf{a}_3{}^\top \mathbf{u}}{v'p'_3 - p'_2} \end{aligned} \tag{3}$$

Considering only the first of these expressions for t, we see that the point $\mathbf{x}$ may be written as

$$\begin{aligned} \mathbf{x} &= \begin{pmatrix} \mathbf{u} \\ (\mathbf{a}_1{}^\top \mathbf{u} - u'\mathbf{a}_3{}^\top \mathbf{u})/(u'p'_3 - p'_1) \end{pmatrix} \\ &\approx \begin{pmatrix} (u'p'_3 - p'_1)\mathbf{u} \\ \mathbf{a}_1{}^\top \mathbf{u} - u'\mathbf{a}_3{}^\top \mathbf{u} \end{pmatrix} \end{aligned}$$

Now, projecting this point via the third camera, we find that

$$\begin{aligned} \mathbf{u}'' &\approx (B \mid \mathbf{p}'')\mathbf{x} \\ &\approx (u'p'_3 - p'_1)B\mathbf{u} + \mathbf{p}''(\mathbf{a}_1{}^\top \mathbf{u} - u'\mathbf{a}_3{}^\top \mathbf{u}) \\ &\approx \left((u'p'_3 - p'_1)B + \mathbf{p}''(\mathbf{a}_1{}^\top - u'\mathbf{a}_3{}^\top)\right)\mathbf{u} \end{aligned}$$

Writing this in terms of the rows of B (denoted by $\mathbf{b}_i{}^\top$), we have an expression

$$\mathbf{u}'' \approx \begin{pmatrix} (u'p'_3 - p'_1)\mathbf{b}_1{}^\top + p''_1(\mathbf{a}_1{}^\top - u'\mathbf{a}_3{}^\top) \\ (u'p'_3 - p'_1)\mathbf{b}_2{}^\top + p''_2(\mathbf{a}_1{}^\top - u'\mathbf{a}_3{}^\top) \\ (u'p'_3 - p'_1)\mathbf{b}_3{}^\top + p''_3(\mathbf{a}_1{}^\top - u'\mathbf{a}_3{}^\top) \end{pmatrix}\mathbf{u} \ .$$

We may rearrange terms to obtain

$$\mathbf{u}'' \approx \begin{pmatrix} u'(p'_3\mathbf{b}_1{}^\top - p''_1\mathbf{a}_3{}^\top) - (p'_1\mathbf{b}_1{}^\top - p''_1\mathbf{a}_1{}^\top) \\ u'(p'_3\mathbf{b}_2{}^\top - p''_2\mathbf{a}_3{}^\top) - (p'_1\mathbf{b}_2{}^\top - p''_2\mathbf{a}_1{}^\top) \\ u'(p'_3\mathbf{b}_3{}^\top - p''_3\mathbf{a}_3{}^\top) - (p'_1\mathbf{b}_3{}^\top - p''_3\mathbf{a}_1{}^\top) \end{pmatrix}\mathbf{u} \tag{4}$$

In a similar manner, starting with the other estimate of t in (3), we have a second formula for $\mathbf{u}''$, namely

$$\mathbf{u}'' \approx \begin{pmatrix} v'(p'_3\mathbf{b}_1{}^\top - p''_1\mathbf{a}_3{}^\top) - (p'_2\mathbf{b}_1{}^\top - p''_1\mathbf{a}_2{}^\top) \\ v'(p'_3\mathbf{b}_2{}^\top - p''_2\mathbf{a}_3{}^\top) - (p'_2\mathbf{b}_2{}^\top - p''_2\mathbf{a}_2{}^\top) \\ v'(p'_3\mathbf{b}_3{}^\top - p''_3\mathbf{a}_3{}^\top) - (p'_2\mathbf{b}_3{}^\top - p''_3\mathbf{a}_2{}^\top) \end{pmatrix}\mathbf{u} \tag{5}$$

By eliminating in the usual way the constant factor implied by the $\approx$ sign in equations (4) and (5), we obtain the four independent trilinear forms defined by Shashua ([7]). Following Shashua ([7]) we now write

$$\boldsymbol{\alpha}_{ij}{}^\top = p'_i\mathbf{b}_j{}^\top - p''_j\mathbf{a}_i{}^\top \tag{6}$$

Then, the equations (4) and (5) may be written as

$$\mathbf{u}'' \approx \begin{pmatrix} u'\boldsymbol{\alpha}_{31}{}^\top - \boldsymbol{\alpha}_{11}{}^\top \\ u'\boldsymbol{\alpha}_{32}{}^\top - \boldsymbol{\alpha}_{12}{}^\top \\ u'\boldsymbol{\alpha}_{33}{}^\top - \boldsymbol{\alpha}_{13}{}^\top \end{pmatrix}\mathbf{u} \approx \begin{pmatrix} v'\boldsymbol{\alpha}_{31}{}^\top - \boldsymbol{\alpha}_{21}{}^\top \\ v'\boldsymbol{\alpha}_{32}{}^\top - \boldsymbol{\alpha}_{22}{}^\top \\ v'\boldsymbol{\alpha}_{33}{}^\top - \boldsymbol{\alpha}_{23}{}^\top \end{pmatrix}\mathbf{u} \ . \tag{7}$$

Eliminating the unknown scale factors in the usual way, we obtain for each correspondence $\mathbf{u} \leftrightarrow \mathbf{u}' \leftrightarrow \mathbf{u}''$ a set of 4 equations in the unknown entries of the vectors α_{ij}. Given 7 point correspondences, we have sufficiently many equations to solve for the vectors α_{ij}. Thus, we may obtain the transfer equations (7) linearly from a set of 7 point matches in all three images.

There are other ways of finding the particular coefficients α_{ij} in the transfer equations. For instance, given sufficiently many correspondences $\mathbf{u} \leftrightarrow \mathbf{u}'$ between the first two images, we may do a complete projective reconstruction of the scene ([1, 2]) to obtain the position of points in a projective reconstruction of the scene. One also obtains the camera matrices $(I \mid 0)$ and $(A \mid \mathbf{p}')$ of the first two cameras from this reconstruction ([2]). Subsequently, given sufficiently many points in the third image that correspond with matched points in the first two images, we may compute the camera matrix $P'' = (B \mid \mathbf{p}'')$ directly. Specifically, given matches $\mathbf{u} \leftrightarrow \mathbf{u}' \leftrightarrow \mathbf{u}''$, we effectively have a match $\mathbf{x} \leftrightarrow \mathbf{u}''$, where $\mathbf{x}$ is the reconstructed point in the scene. From these scene – image matches one can compute the camera matrix P''. This is the classical camera resectioning problem. From six or more point matches (actually $5\frac{1}{2}$ matches suffice) one can compute the camera matrix $P'' = (B \mid \mathbf{p}'')$ using the Direct Linear Transformation (DLT) method ([9]). For more precise results, the matrix P'' can be computed using iterative techniques ([8]). Once the three camera matrices are known, the coefficients of the transfer equations (7) may be computed directly.

4 Connection with Line Transfer

In [3, 4] a triple of matrices T_i were introduced for the purpose of transferring lines from a pair of images into a third image. This method was used to do projective reconstruction from line matches. We summarize some of the main results of that paper here, using the notation of this note.

Theorem 4.1. *Consider three images of a set of lines taken by cameras with camera matrices $P = (I \mid 0)$, $P' = (A \mid \mathbf{p}')$ and $P' = (B \mid \mathbf{p}'')$. Let $\boldsymbol{\lambda} \leftrightarrow \boldsymbol{\lambda}' \leftrightarrow \boldsymbol{\lambda}''$ be a set of corresponding lines in the three images. Then there exist 3×3 matrices T_1, T_2 and T_3 such that*

$$\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}'^\top T_1 \boldsymbol{\lambda}'' \\ \boldsymbol{\lambda}'^\top T_2 \boldsymbol{\lambda}'' \\ \boldsymbol{\lambda}'^\top T_3 \boldsymbol{\lambda}'' \end{pmatrix} \ . \tag{8}$$

Each matrix T_i has the form

$$T_i = \hat{\mathbf{a}}_i \mathbf{p}''^\top - \mathbf{p}' \hat{\mathbf{b}}_i{}^\top \tag{9}$$

where $\hat{\mathbf{a}}_i$ is the i-th column *of the matrix A, and $\hat{\mathbf{b}}_i$ is the i-th column of B.*

Note the difference between $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{a}}_i$ and the vectors $\mathbf{b}_i$ and $\mathbf{a}_i$ that appear in (6). Vector ${\mathbf{b}_i}^\top$ is the i-th *row* of the matrix B, and ${\mathbf{a}_i}^\top$ is similarly defined for A. Vectors $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$ on the other hand are the *columns* of A and B.

Since each line match gives two linear equations in the entries of the matrices T_i, it is possible, gven at least 13 line matches, to compute the 27 unknown entries, up to an unknown (and insignificant) common scale factor.

It is further shown in [3, 4] how the camera matrices of the three cameras may be computed from the three matrices T_i. This is done using non-iterative linear techniques. In this way, the projective structure of a scene may be deduced from a set of 13 lines seen in 3 views.

The fact that we have 27 unknown values in the three matrices T_i as well as in the 9 vectors $\boldsymbol{\alpha}_{ij}$ suggests that there could be a connection between these techniques. This is indeed the case as will now be demonstrated. We denote the k-th entry of the vector $\boldsymbol{\alpha}_{ij}$ by α_{ijk}. Similarly, we denote the (jk)-th entry of the matrix T_i by T_{ijk}. The entries of matrices A and B will be denoted by a_{ij} and b_{ij}.

Now, from (6) we have

$$\alpha_{ijk} = p'_i b_{jk} - p''_j a_{ik} \ .$$

Further, from (9) we have

$$T_{ijk} = a_{ji} p''_k - p'_j b_{ki}$$

One immediately sees that

$$\alpha_{ijk} = -T_{kij} \tag{10}$$

This equation has a number of significant implications.

1. Given a set of 7 point matches in 3 images, one has a linear method of computing the projective structure of the scene, including the three camera matrices and the fundamental matrices. One uses the 7-point linear method to compute the values α_{ijk} and from this one reassembles the matrices T_i. Finally, the methods of [3, 4] may be used to compute the camera matrices.

2. We may compute (linearly) the projective structure of a set of lines and points seen in three views, provided that $2\#lines + 4\#points \geq 26$. In particular, for each point, equation (7) gives 4 equations in the values α_{ijk}, and for each line, equation (8) gives two equations in the entries T_{ijk}. However, the unknowns T_{ijk} and α_{ijk} are the same (modulo a change of sign and permutation of the indices). Thus, provided we have at least 26 equations, we may compute the values of T_{ijk} linearly and then reconstruct the scene as in [3, 4].

There is an alternative method of computing the matrices T_i, and hence projective structure from seven points. For each pair of points in space, one considers the line that passes through these two points. The corresponding line in an image is the line that passes through the images of the two points. In this way counting all pairs of points, one has a total of 21 lines. Applying the line algorithm to this set of lines one may solve for T_i. Because of obvious redundancies between these lines it is not clear that the 21 lines will contain enough information to ensure that the set of equations derived from (8) will have sufficient rank to ensure a unique solution. It has been empirically observed that they do.

On the other hand with only 6 points one has 15 lines, more than the minimum number of 13 lines necessary for the line-algorithm. However in this case the equation matrix does not have sufficient rank and a solution can not be found. This is not surprising, since otherwise we would have a linear algorithm for solving for six points in three views. This is not likely, since according to ([6]) this configuration leads to a cubic equation with three possible solutions.

It is not clear whether using 7 points to derive 21 line equations will give the same (or as good) results as using them directly to derive 28 point equations using (7). Since information is apparently lost in using the points only to define lines, it would appear that this is not in general a good approach, and that it is preferable to use (7) to derive four equations for each point directly.

References

[1] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer-Verlag*, pages 563 – 578, 1992.

[2] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[3] R. I. Hartley. Camera calibration using line correspondences. In *Proc. DARPA Image Understanding Workshop*, pages 361–366, 1993.

[4] Richard I. Hartley. Projective reconstruction from line correspondences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 903–907, 1994.

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

[6] L. Quan. Invariants of 6 Points from 3 Uncalibrated Images. Rapport Technique RT 101 IMAG 19 LIFIA, LIFIA-IMAG, Grenoble, October 1993. To appear in ECCV94.

[7] Amnon Shashua. Algebraic functions for recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):779–789, August 1995.

[8] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, fourth edition, 1980.

[9] I.E. Sutherland. Three dimensional data input by tablet. *Proceedings of IEEE*, Vol. 62, No. 4:453–461, April 1974.