A Geometric Analysis of the Trifocal Tensor

C.B.U. Perwass [1], J. Lasenby [2]

[1] Cavendish Laboratory, Madingley Road, Cambridge CB3 OHE, cperwass@mrao.cam.ac.uk

[2] C. U. Engineering Department, Trumpington Street, Cambridge CB2 1PZ, jl@eng.cam.ac.uk

Abstract: Reconstruction of 3D-objects from a number of images is a central subject of Computer Vision. In this paper we will investigate the geometrical structure of the trifocal tensor using Geometric Algebra. Furthermore, we will give a novel expression for the trifocal tensor, derive constraints on its geometrical structure and investigate its reconstruction ability computationally. We will show that the reconstruction quality is not directly related to the self-consistency of the trifocal tensor.

Keywords: Trifocal Tensor, Geometric Algebra, Grassmann-Cayley Algebra, Reciprocal Frames, Reconstruction

[Published in proceedings *Image and Vision Computing New Zealand*. The University of Auckland, 1998.]

1 Introduction

Recently there has been much interest in deriving and characterising the trifocal tensor. The trifocal tensor is used to obtain a projective reconstruction from three images, taken with uncalibrated cameras from unknown positions of the same 3D-scene. It can also be used to transfer lines or points from one image to another. [3] and [11] give a discussion of the structure of the trifocal tensor and present examples of its use.

In effect the trifocal tensor encodes the relative positions and orientations of the cameras. It can be calculated if at least 7 point matches over the three images are available. Once the trifocal tensor has been calculated, the epipoles, camera matrices and fundamental matrices can be extracted from it. The quality of the initial point matches is crucial for obtaining good estimates of these values, however. Therefore, a lot of research has gone into obtaining a good estimate of the trifocal tensor from not so good point matches. The main problem being how to decide what estimate of a trifocal tensor is "good" if only point matches and nothing else are known.

The trifocal tensor has also been studied in terms of *Grassmann-Cayley* (GC) algebra ([1], [8], [2]). A derivation and analysis in terms of *Geometric Algebra* (GA) can be found in [6] and [7].

In this paper the derivation and analysis of the trifocal tensor in terms of *Geometric Algebra* will be extended. Although GA is similar to GC algebra, it will be shown that GA has some distinct advantages due to its use of the *inner product*. This is especially apparent in a novel interpretation of camera matrices and the trifocal tensor. In particular, a concise expression for the trifocal tensor is given, which allows a better insight into its geometrical meaning. Also, a set of constraints on the internal structure of the trifocal tensor will be derived. These constraints form a superset of constraints previously derived in [1] and [2]. However, here the derivation is done purely geometrically and not through the investigation of polynomials as in [1]. The effect of the newly found constraints on the reconstruction ability of the trifocal tensor will be investigated computationally.

2 Geometric Algebra

This section will give a very brief introduction to GA. We refer the reader to [4] and [5] for a thorough treatment of GA. A shorter derivation of the most important results can be found in [6] and [7].

The central operation in GA is the *geometric product*. The geometric product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $\boldsymbol{ab}$ and is defined by,

$$\boldsymbol{ab} = \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{b}$$

$\boldsymbol{a}\cdot\boldsymbol{b}$ is the *inner product* and $\boldsymbol{a}\wedge\boldsymbol{b}$ is the *outer product* of $\boldsymbol{a}$ and $\boldsymbol{b}$. The outer product is anti-commutative, whereas the inner product is commutative. The outer product of two vectors is called a *bivector*, or a *multivector of grade 2*. In an N-dimensional space exactly N mutually orthogonal vectors can be formed. Therefore, the highest grade multivector, or *pseudoscalar* of that space is of grade N. Obviously the pseudoscalars of a space can only differ by a scalar factor.

The standard results concerning inner and outer products of multivectors which will be needed in following sections can be found in [9]. We stress here that the role of the inner product in GA is much more than that of the inner product in standard vector calculus. For example, intersections and the concept of duality can be expressed via the inner product. The GC algebra lacks such an explicit inner product.

3 Projective Geometry

This section will outline the GA framework for projective geometry. We define a set of 4 basis vectors $\{e_1, e_2, e_3, e_4\}$ with metric $\{- - - +\}$. The pseudoscalar of this space is defined as $I = e_1\wedge e_2\wedge e_3\wedge e_4$. A vector in this 4D-space (P^3), which will be called a *homogeneous* vector, can then be regarded as a projective line which describes a point in the corresponding 3D-space (E^3). Also, a line in E^3 is represented in P^3 by the outer product of two homogeneous vectors, and a plane in E^3 is given by the outer product of three homogeneous vectors in P^3. In the following, homogeneous vectors in P^3 will be written as capital letters, and their corresponding 3D-vectors in E^3 as lower case letters in bold face.

The projection of a 4D vector A into E^3 is given by,

$$\boldsymbol{a} = \frac{A\wedge e_4}{A\cdot e_4}$$

This is called the *projective split*. Note that a homogeneous vector with no e_4 component will be projected onto the plane at infinity.

A set $\{A_\mu\}$ of four homogeneous vectors forms a basis or *frame* of P^3 if and only if $(A_1\wedge A_2\wedge A_3\wedge A_4) \neq 0$. The *characteristic pseudoscalar* of this frame for 4 such vectors is defined as $I_a = A_1\wedge A_2\wedge A_3\wedge A_4$. Note that $I_a = \rho_a I$, where ρ_a is a scalar. This and results relating the inner products of multivectors with the pseudoscalars of the space are given in [9].

Another concept which is very important in the analysis to be presented is that of the dual of a multivector X. This is written as X^* and is defined as $X^* = XI^{-1}$. It will be extremely useful to introduce the *dual bracket* and the *inverse dual bracket*. They are related to the bracket notation as used in GC algebra and GA, [6]. The bracket of a pseudoscalar P is a scalar, defined as the dual of P in GA. That is, $[P] = PI^{-1}$. The dual and inverse dual brackets are defined as

$$\begin{aligned} [\![A_{\mu_1}\cdots A_{\mu_n}]\!]_a &\equiv (A_{\mu_1}\wedge\ldots\wedge A_{\mu_n})I_a^{-1} \quad\text{and}\quad [\![A_{\mu_1}\cdots A_{\mu_n}]\!] \equiv (A_{\mu_1}\wedge\ldots\wedge A_{\mu_n})I^{-1} \\ \langle\!\langle A_{\mu_1}\cdots A_{\mu_n}\rangle\!\rangle_a &\equiv (A_{\mu_1}\wedge\ldots\wedge A_{\mu_n})I_a \quad\text{and}\quad \langle\!\langle A_{\mu_1}\cdots A_{\mu_n}\rangle\!\rangle \equiv (A_{\mu_1}\wedge\ldots\wedge A_{\mu_n})I \end{aligned} \tag{1}$$

with $n \in \{0, 1, 2, 3, 4\}$. The range given here for n means that in P^3 none, one, two, three or four homogeneous vectors can be bracketed with a dual or inverse dual bracket. For example, if $P = A_1\wedge A_2\wedge A_3\wedge A_4$, then $[\![A_1A_2A_3A_4]\!] = [\![P]\!] = [P] = \rho_a$.

Using this bracket notation the *normalized reciprocal A-frame*, written $\{A_a^\mu\}$, is defined as $A_a^{\mu_1} = [\![A_{\mu_2}A_{\mu_3}A_{\mu_4}]\!]_a$. It is also useful to define a *standard reciprocal A-frame*: $A^{\mu_1} = [\![A_{\mu_2}A_{\mu_3}A_{\mu_4}]\!]$. Then, $A_\mu\cdot A_a^\nu = \delta_\mu^\nu$ and $A_\mu\cdot A^\nu = \rho_a\delta_\mu^\nu$, where δ_μ^ν is the Kronecker delta. That is, a reciprocal frame vector is nothing else but the dual of a plane. In the GC algebra these reciprocal vectors would be defined as elements of a *dual space*, which is indeed what is done in [1]. However, because GC algebra does not have an explicit inner product, elements of this dual space cannot operate on elements of the "normal" space. Hence, the concept of reciprocal frames cannot be defined in the GC algebra.

A reciprocal frame can be used to transform a vector from one frame into another. That is, $X = (X\cdot A_a^\mu)A_\mu = (X\cdot A_\nu)A_a^\nu$. It will be important later not only to consider vector frames but also line frames. The A-line frame $\{L_a^i\}$ is defined as $L_a^{i_1} = A_{i_2}\wedge A_{i_3}$. The $\{i_1, i_2, i_3\}$ are assumed to be an even permutation of $\{1,2,3\}$. The *normalised reciprocal A-line frame* $\{\bar{L}_i^a\}$ and the *standard reciprocal A-line frame* $\{L_i^a\}$ are given by $\bar{L}_i^a = [\![A_i A_4]\!]_a$ and $L_i^a = [\![A_i A_4]\!]$, respectively. Hence, $L_a^i \cdot \bar{L}_j^a = \delta_j^i$ and $L_a^i \cdot L_j^a = \rho_a \delta_j^i$. Again, this shows the universality of the inner product: bivectors can be treated in the same fashion as vectors.

The *meet* and *join* are the two operations needed to calculate intersections between two lines, two planes or a line and a plane – these are discussed in more detail in [9],[5] and [6]; here we will give just the most relevant expression for the meet. If A and B represent two planes or a plane and a line in P^3 their meet may be written as

$$A \vee B = \langle\!\langle [\![A]\!][\![B]\!] \rangle\!\rangle = [\![A]\!] \cdot B \tag{2}$$

A pinhole camera can be defined by 4 homogeneous vectors in P^3: one vector gives the optical centre and the other three define the image plane [6], [7]. Thus, the vectors needed to define a pinhole camera also define a frame for P^3. Conventionally the fourth vector of a frame, eg. A_4, defines the optical centre, and the outer product of the other three defines the image plane.

Suppose that X is given in some frame $\{Z_\mu\}$ as $X = \zeta^\mu Z_\mu$, it can be shown [9] that the projection of some point X onto the A-image plane can be written as

$$X_a = (\,X \cdot A^i\,) A_i = (\zeta^\mu\, Z_\mu \cdot A^i\,) A_i = \zeta^\mu K_{i \atop \mu} A_i\,; \quad K_{i \atop \mu} \equiv Z_\mu \cdot A^i \tag{3}$$

The matrix $K_{i \atop \mu}$ is the *camera matrix* of camera A, for projecting points given in the Z-frame onto the A-image plane[1].

In [1] the derivations begin with the camera matrices by noting that the row vectors *refer* to planes. As was shown here, the row vectors of a camera matrix are the reciprocal frame vectors $\{A^i\}$, whose dual *is* a plane.

With the same method as before, lines can be projected onto an image plane. For example, let L be some line in P^3, then its projection onto the A-image plane is: $(L \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) = (\,L \cdot L_i^a\,) L_a^i$.

4 The Trifocal Tensor

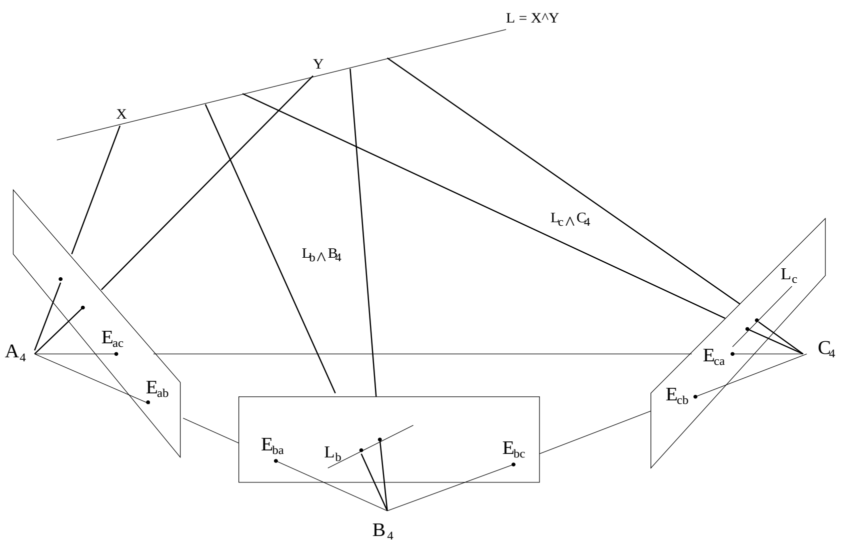

Figure 1: Line projected onto three image planes.

[1]Note that the indices of K are not given as super- and subscripts of K but are raised (or lowered) relative to each other. This notation was adopted since it leaves the superscript position of K free for other usages.

Let the frames $\{A_\mu\}$, $\{B_\mu\}$ and $\{C_\mu\}$ define three distinct cameras. Also, let $L = X \wedge Y$ be some line in P^3. The plane $L \wedge B_4$ is then the same as the plane $\lambda_i^b L_b^i \wedge B_4$, up to a scalar factor, where $\lambda_i^b = L \cdot L_i^b$. But $L_b^{i_1} \wedge B_4 = B_{i_2} \wedge B_{i_3} \wedge B_4 = \langle\!\langle B^{i_1} \rangle\!\rangle$. Intersecting planes $L \wedge B_4$ and $L \wedge C_4$ has to give L. If two lines intersect, their outer product is zero. Thus, the outer product of $A_4 \wedge X$ (or $A_4 \wedge Y$) with L has to be zero. In [9] it is shown how we can use these facts to obtain a particularly concise and geometrically meaningful expression for the trifocal tensor $T_i{}^{jk}$;

$$T_i{}^{jk} = \left[\!\!\left[(A_4 \wedge A_i) \langle\!\langle B^j C^k \rangle\!\rangle \right]\!\!\right] \tag{4}$$

which has to satisfy $\alpha^i \lambda_j^b \lambda_k^c T_i{}^{jk} = 0$, where $\alpha^i = X \cdot A^i$. This expression for the trifocal tensor can be expanded in two different, but equivalent ways. The first way yields,

$$T_i{}^{jk} = K^b_{j_i} K^c_{k_4} - K^c_{k_i} K^b_{j_4} \tag{5}$$

where K^b and K^c are the camera matrices for cameras B and C relative to camera A, respectively. This is the expression for the trifocal tensor given by Hartley in [3]. On the other hand, equation (4) can also be expanded to

$$T_i{}^{jk} = L_i^a \cdot \langle\!\langle B^j C^k \rangle\!\rangle \tag{6}$$

This expression for the trifocal tensor is somewhat more instructive than the previous one. It can be shown [9] that the projection of the line $T^{jk} = \langle\!\langle B^j C^k \rangle\!\rangle$ onto image plane A, denoted by T_a^{jk}, is $T_a^{jk} = T_i{}^{jk} L_a^i$, thus giving another geometric interpretation of the trifocal tensor coefficients.

The trifocal tensor may be minimally factorised using the epipoles. Nonetheless, it may be better to constrain the trifocal tensor without using any values other than the componenets of the trifocal tensor itself. Such constraints can be derived in the GA framework [9], following the approach given in [1]. However, not only has this approach been generalized but the arguments used are also purely geometrical in origin. In particular, the derivation given does not involve working with any polynomials.

The underlying idea is to find relations between the lines T^{jk} which also hold for their projections T_a^{jk}. Relations between the T_a^{jk} can in turn be directly related to the components of the trifocal tensor.

In the following, the $\{i_1, i_2, i_3\}$, etc. are *no longer* assumed to be any particular kind of permutation. By considering the intersections of lines and the fact that various intersection points lie on particular lines, the following constraints on the elements of T can be derived [9];

$$\begin{aligned}
0 &= |T_a^{j_1k_1} T_a^{j_2k_1} T_a^{j_1k_2}||T_a^{j_2k_2} T_a^{j_1k_3} T_a^{j_2k_3}| - |T_a^{j_1k_1} T_a^{j_2k_1} T_a^{j_2k_2}||T_a^{j_1k_2} T_a^{j_1k_3} T_a^{j_2k_3}| \\
0 &= |T_a^{j_1k_1} T_a^{j_1k_2} T_a^{j_2k_1}||T_a^{j_2k_2} T_a^{j_3k_1} T_a^{j_3k_2}| - |T_a^{j_1k_1} T_a^{j_1k_2} T_a^{j_2k_2}||T_a^{j_2k_1} T_a^{j_3k_1} T_a^{j_3k_2}| \\
0 &= |T_a^{i_1j_1} T_a^{i_2j_1} T_a^{i_1j_2}|\,|T_a^{i_2j_2} T_a^{i_1j_2} T_a^{i_2j_3}| - |T_a^{i_1j_1} T_a^{i_2j_1} T_a^{i_2j_2}|\,|T_a^{i_1j_2} T_a^{i_1j_3} T_a^{i_2j_2}|
\end{aligned} \tag{7}$$

where, for example, $|T_a^{j_1k_1} T_a^{j_2k_1} T_a^{j_1k_2}|$ is defined as the determinant of a matrix with rows given by the components of lines $T_a^{j_1k_1}$, $T_a^{j_2k_1}$ and $T_a^{j_1k_2}$ in exactly that order from top to bottom.

5 Computations

It is interesting to see what effect the determinant constraints have on the "quality" of a trifocal tensor. That is, a trifocal tensor calculated only from point matches has to be compared with a trifocal tensor calculated from point matches while enforcing the determinant constraints. For the calculation of the former, a simple linear algorithm is used that employs the trilinearity relationships, as for example given by Hartley in [3]. In the following this algorithm will be called the "7pt" algorithm.

To enforce all the determinant constraints, first an estimate of the trifocal tensor is found using the 7pt algorithm. From this tensor the epipoles are extracted and the image points are transformed into the epipolar frame. With these transformed point matches the trifocal tensor can then be found in the epipolar basis (it can be shown [7] that the trifocal tensor in the epipolar basis has only 7 non-zero components). The trifocal tensor in the "normal" basis is then recovered by transforming the trifocal tensor in the epipolar basis back with the initial estimates of the epipoles. The trifocal tensor found in this way has to be fully self-consistent since it was calculated from the minimal number of parameters. That also means that the determinant constraints have to be fully satisfied. This algorithm will be called the "MinFact" algorithm.

In [9] the relative merits of the linear and "MinFact" algorithms are discussed.

The question is, of course, *how* to measure the quality of the trifocal tensor. Here the quality is measured by how good a reconstruction can be achieved with the trifocal tensor in a geometric sense. Having generated simulated data (with varying amounts of noise), we find the trifocal tensor with each of the two algorithms described above. The epipoles and camera matrices are extracted from T and points are reconstructed using a version of what is called "Method 3" in [10] adapted for three views. However, this reconstruction still contains an unknown projective transformation. Therefore, a projective transformation matrix that best transforms the reconstructed points into the true points (which are known for synthetic data) can be calculated. Then the reconstruction can be compared with the original 3D-object geometrically. The final measure of "quality" is arrived at by calculating the mean distance in 3D-space between the reconstructed and the true points. These quality values are evaluated for a number of different noise magnitudes. For each particular noise magnitude the above procedure is performed 100 times. The final quality value for a particular noise magnitude is then taken as the average of the 100 trials.

Figure 2 shows the mean distance between the original points and the reconstructed points in 3D-space in some arbitrary units, as a function of the noise magnitude. The camera resolution was 600 by 600 pixels.

This figure shows that for a noise magnitude of up to approximately 10 pixels both trifocal tensors seem to produce equally good reconstructions. Note that for zero added noise the reconstruction quality is not perfect. This is due to the quantisation noise of the cameras. The small increase in quality for low added noise compared to zero added noise is probably due to the cancellation of the quantisation and the added noise.

Apart from looking at the reconstruction quality it is also interesting to see how close the components of the calculated trifocal tensors are to those of the true trifocal tensor. Figure 3 shows the mean of the percentage differences between the components of the true trifocal tensor and the trifocal tensor calculated with the 7pt and MinFact algorithms, as a function of added noise in pixels. This shows that the trifocal tensor calculated with the MinFact algorithm is indeed very different to the true trifocal tensor, much more so than the trifocal tensor calculated with the 7pt algorithm.

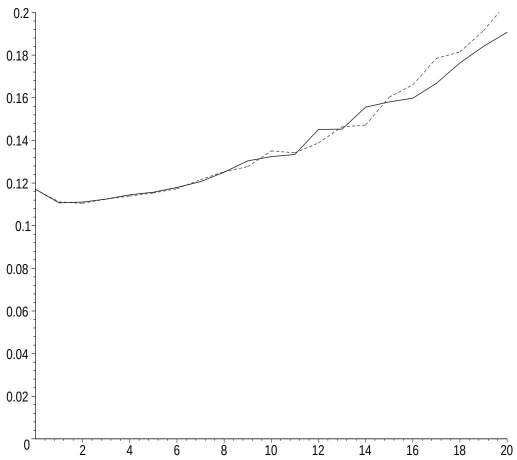

Figure 2: Reconstruction quality as a function of noise, for MinFact algorithm (solid line) and 7pt algorithm (dashed line).

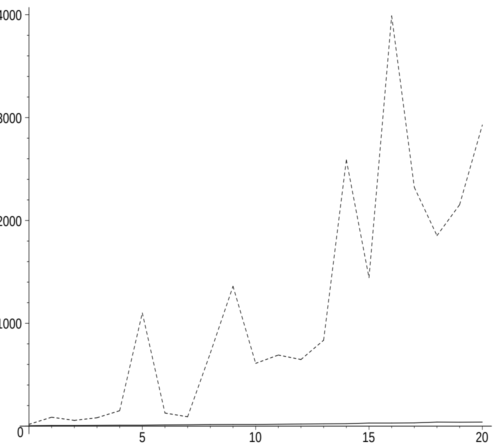

Figure 3: Mean difference between elements of calculated and true tensors in percent, for MinFact algorithm (solid line) and 7pt algorithm (dashed line).

6 Conclusions

We conclude that the GA approach to the trifocal tensor problem leads to a clearer geometrical understanding. In particular, constraints on the internal structure of the trifocal tensor could be derived through mainly geometrical arguments. The use of *reciprocal frames* and especially their extension to line frames clearly shows the advantage of the GA approach over a GC algebra approach, due to GA's *inner product*.

The data presented in section 5 seems to indicate that a tensor that obeys the determinant constraints, i.e. is self-consistent, but does not satisfies the trilinearity relationships particularly well (MinFact algorithm) is equally good, in terms of reconstruction ability, as an inconsistent trifocal tensor that satisfies the trilinearity relationships quite well (7pt algorithm). In particular the fact that the trifocal tensor calculated with the MinFact algorithm is so very much different to the true trifocal tensor (see figure 3) does not seem to have a big impact on the final recomputation quality.

One possible explanation for this is that all the differences between the reconstructions are evened out when the final projective transformation is applied. That would mean that to strive for a very good estimate of the trifocal tensor is not actually necessary since any reconstruction will always include a projective transformation that can be chosen arbitrarily. However, it remains to be seen if an algorithm which minimises a cost function while imposing the constraints via Lagrange multipliers gives better results [2].

References

[1] Olivier Faugeras and Bernard Mourrain. On the Geometry and Algebra of the Point and Line Correspondences between N Images. Technical Report 2665, INRIA, Sophia Antipolis, 1995.

[2] Olivier Faugeras and Theodore Papadopoulo. Grassmann-Cayley Algebra for Modelling Systems of Cameras and the Algebraic Equations of the Manifold of Trifocal Tensors. *Phil. Trans. R. Soc. Lond. A*, 356(1740):1123–1152, 1998.

[3] Richard I. Hartley. Lines and Points in Three Views and the Trifocal Tensor. *International Journal of Computer Vision*, pages 125–140, 1997.

[4] D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Dordrecht, 1984.

[5] David Hestenes and Renatus Ziegler. Projective Geometry with Clifford Algebra. *Acta Applicandae Mathematicae*, 23:25–63, 1991.

[6] Joan Lasenby and E. Bayro-Corrochano. Computing Invariants in Computer Vision using Geometric Algebra. Technical Report CUED/F - INFENG/TR. 224, Cambridge University Engineering Department, 1997.

[7] Joan Lasenby and Anthony N. Lasenby. Estimating Tensors for Matching over Multiple Views. *Phil. Trans. R. Soc. Lond. A*, 356(1740):1267–1282, 1998.

[8] Theodore Papadopoulo and Olivier Faugeras. A New Characterization of the Trifocal Tensor. On INRIA Sophia Antipolis Web-Site.

[9] C.B.U. Perwass and J. Lasenby. A Geometric Derivation of the Trifocal Tensor and its Constraints. Technical Report CUED/F - INFENG/TR. 331, Cambridge University Engineering Department, 1998.

[10] Charlie Rothwell, Olivier Faugeras, and Gabriella Csurka. A Comparison of Projective Reconstruction Methods for Pairs of Views. *Computer Vision and Image Understanding*, 68-1:37–58, 1997.

[11] A. Shashua. Trilinear Tensor: the Fundamental Construct of Multiple-View Geometry and its Applications. In G. Sommer and J.J. Koenderink, editors, *Algebraic frames for the Perception-Action Cycle*, number 1315 in Lecture Notes in Computer Science, 1997.