A Geometric Derivation of the Trifocal Tensor and its Constraints

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

CUED/F - INFENG/TR. 331
October 1998

[1] Cavendish Laboratory, Madingley Road, Cambridge CB3 OHE, UK
+44 (1223) 337366
cperwass@mrao.cam.ac.uk
www.mrao.cam.ac.uk/˜cperwass

[2] C. U. Engineering Department, Trumpington Street, Cambridge CB2 1PZ, UK
+44 (1223) 332767
jl@eng.cam.ac.uk
www-sigproc.eng.cam.ac.uk/˜jl

Abstract

Reconstruction of 3D-objects from a number of images is a central subject of Computer Vision. In this paper we 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.

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

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. [1] and [2] 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 ([3], [4], [5]). 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 geometric algebra 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 [3] and [5]. However, here the derivation is done purely geometrically and not through the investigation of polynomials as in [3]. The effect of the newly found constraints on the reconstruction ability of the trifocal tensor will be investigated computationally.

2 Geometric Algebra

Since all the analysis in this paper is carried out in terms of GA, a short introduction will be given here. All the calculation rules and identities needed to follow this report are presented. However, proofs and derivations will be omitted. The interested reader may refer to [8] and [9] for a thorough treatment of GA. A shorter derivation of the most important results can be found in [6] and [7].

The easiest way to understand GA is to show how it extends the functionality of standard vector calculus (SVC), which we assume all readers are familiar with. In SVC the starting point is to define a frame. Here all calculations are performed in a cartesian frame, so we can start by defining an orthonormal basis of E^3, $\{e_1, e_2, e_3\}$ with signature $\{+++\}$. A vector $\boldsymbol{a}$ in this basis may then be defined as

$$\boldsymbol{a} = \alpha^i e_i$$

Here, as throughout the rest of the text, greek indices will be assumed to count from 1 to 4 and latin indices to count from 1 to 3. Also, a superscript index repeated as a subscript (or vice versa) implies a summation over the range of that index, unless specifically stated otherwise. Now, SVC defines a *scalar product* of two vectors which results in a scalar. For example, the scalar product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $s = \boldsymbol{a}\cdot\boldsymbol{b}$, where s is a scalar. The scalar s gives some information about the relative orientation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. That is, the scalar product is a *metric* operation, since it is only defined in relation to a frame.

GA extends the scalar product to an *inner product*. The inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is still written as $\boldsymbol{a}\cdot\boldsymbol{b}$ and it has the same metric meaning. However, the inner product can also be applied in a non-metric sense. In order to see this, we will first have to introduce the *outer product*.

The outer product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $\boldsymbol{a}\wedge\boldsymbol{b}$ and is called a *2-blade*. A 2-blade may be regarded as an *oriented area*. Analogously, the outer product of three vectors, a *3-blade*, $\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}$ can be interpreted as an *oriented volume*. However, in projective geometry, which will be treated later on, the geometric meaning of 2-blades and 3-blades is quite different. A more general interpretation of k-blades will be given at the end of this section.

The outer and inner product are also defined in the absence of a basis frame. This is where the power of GA lies. Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in E^n$, then the following rules apply to the inner product:

1. If $\boldsymbol{a}\cdot\boldsymbol{b} = 0$ then $\boldsymbol{a}$ and $\boldsymbol{b}$ are said to be *orthogonal*.
2. $\boldsymbol{a}\cdot\boldsymbol{b} = \boldsymbol{b}\cdot\boldsymbol{a} = s$ where s is a scalar. That is, the inner product is commutative.
3. $\boldsymbol{a}\cdot(\boldsymbol{b}+\boldsymbol{c}) = \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\cdot\boldsymbol{c}$. Distributive law.

For the outer product we have

1. If $\boldsymbol{a}\wedge\boldsymbol{b} = 0$ then $\boldsymbol{a}$ and $\boldsymbol{b}$ are said to be *parallel* or *linearly dependent*.
2. $\boldsymbol{a}\wedge\boldsymbol{b} = -\boldsymbol{b}\wedge\boldsymbol{a}$. That is, the outer product is anti-commutative.
3. $\boldsymbol{a}\wedge(\boldsymbol{b}+\boldsymbol{c}) = \boldsymbol{a}\wedge\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{c}$. Distributive law.
4. $\boldsymbol{a}\wedge(\boldsymbol{b}\wedge\boldsymbol{c}) = (\boldsymbol{a}\wedge\boldsymbol{b})\wedge\boldsymbol{c}$. Associative law.

From the first rule for the outer product it follows directly that the highest grade object in E^n is of grade n, simply because in E^n at most n mutually linearly independent vectors can be formed. The object of highest grade is called the *pseudoscalar* of that space. Obviously the pseudoscalars of some vector space can only differ by a scalar factor.

A *1-vector*, or simply vector, in GA is the same as a vector in SVC. In that sense it is also equivalent to a 1-blade. However, in GA we can have vectors of higher grade, as well. A *k-vector* is defined to be the sum of a number of k-blades. Note that a k-vector cannot necessarily be expressed as a k-blade, but every k-blade is also a k-vector. Some examples may help to clarify this idea. A 2-vector , or *bivector*, $\boldsymbol{w} \in E^3$ may be given by

$$\boldsymbol{w} = 2(e_1\wedge e_2) + 3(e_1\wedge e_3)$$

This particular bivector can also be written as a 2-blade;

$$\boldsymbol{w} = e_1 \wedge (2e_2 + 3e_3)$$

In fact, in E^3 any 2-vector can be expressed as a 2-blade. However, in higher dimensional spaces this is not necessarily the case. Consider the following bivector in E^4 with basis $\{e_1, e_2, e_3, e_4\}$.

$$\boldsymbol{w} = \alpha(e_1 \wedge e_2) + \beta(e_3 \wedge e_4)$$

where α and β are some scalar factors. This bivector *cannot* be written as a 2-blade.

Just as a k-vector is the sum of a number of k-blades, GA also defines a *multivector* which is the sum of a number of blades that are not necessarily of the same grade[1]. Working with multivectors is considerably more complicated than working with k-vectors. Since they are also not needed in this report multivectors will not be discussed here. We refer the interested reader to [8] and [9].

There is also a distributive law for the inner product with respect to the outer product. The following is an important general result. Let B_s be an s-blade and let the $\{\boldsymbol{a}_r\}$ form a set of unique vectors such that for no two $\boldsymbol{a}_i, \boldsymbol{a}_j \in \{\boldsymbol{a}_r\}$ $\boldsymbol{a}_i \wedge \boldsymbol{a}_j = 0$. Then,

$$B_s \cdot (\boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_r) = \sum_{\{j_i\}} \epsilon_{j_1 j_2 \cdots j_r} \Big[B_s \cdot (\boldsymbol{a}_{j_1} \wedge \boldsymbol{a}_{j_2} \wedge \ldots \wedge \boldsymbol{a}_{j_s}) \Big] \boldsymbol{a}_{j_{s+1}} \wedge \ldots \wedge \boldsymbol{a}_{j_r} \tag{1}$$

where $\epsilon_{j_1 j_2 \cdots j_r}$ is $+1$ if the $\{j_i\}$ form an even permutation of $\{1, 2, \ldots, r\}$, -1 if they form an odd permutation and 0 if any two indices are identical. Admittedly this equation looks rather confusing. A few examples, however, should clarify the situation.

$$\boldsymbol{a} \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) = (\boldsymbol{a} \cdot \boldsymbol{b}_1)\boldsymbol{b}_2 - (\boldsymbol{a} \cdot \boldsymbol{b}_2)\boldsymbol{b}_1 \tag{2-1}$$

$$\boldsymbol{a} \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2 \wedge \boldsymbol{b}_3) = (\boldsymbol{a} \cdot \boldsymbol{b}_1)(\boldsymbol{b}_2 \wedge \boldsymbol{b}_3) - (\boldsymbol{a} \cdot \boldsymbol{b}_2)(\boldsymbol{b}_1 \wedge \boldsymbol{b}_3) + (\boldsymbol{a} \cdot \boldsymbol{b}_3)(\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) \tag{2-2}$$

Furthermore,

$$\begin{aligned} (\boldsymbol{a}_1 \wedge \boldsymbol{a}_2) \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) &= \boldsymbol{a}_1 \cdot \Big(\boldsymbol{a}_2 \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) \Big) \\ &= \boldsymbol{a}_2 \cdot \boldsymbol{b}_1 \ \boldsymbol{a}_1 \cdot \boldsymbol{b}_2 \ - \ \boldsymbol{a}_2 \cdot \boldsymbol{b}_2 \ \boldsymbol{a}_1 \cdot \boldsymbol{b}_1 \end{aligned} \tag{3-1}$$

$$\begin{aligned} (\boldsymbol{a}_1 \wedge \boldsymbol{a}_2) \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2 \wedge \boldsymbol{b}_3) &= \Big[(\boldsymbol{a}_1 \wedge \boldsymbol{a}_2) \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) \Big] \boldsymbol{b}_3 \\ &\quad - \Big[(\boldsymbol{a}_1 \wedge \boldsymbol{a}_2) \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_3) \Big] \boldsymbol{b}_2 \\ &\quad + \Big[(\boldsymbol{a}_1 \wedge \boldsymbol{a}_2) \cdot (\boldsymbol{b}_2 \wedge \boldsymbol{b}_3) \Big] \boldsymbol{b}_1 \end{aligned} \tag{3-2}$$

Equations (2) and (3) clearly show the *non-metric* side of the inner product. For example, in equation (2-1) the inner product of a vector with a bivector results in a vector. In equation (2-2) the inner product of a vector with a *trivector*[2] gives a 2-vector. Similarly for equations (3).

[1] In more general texts on GA a k-vector as defined here is called a homogeneous multivector of grade k. We have chosen not to follow this naming convention since in projective geometry the term "homogeneous vector" is already used to describe something quite different.

[2] A "trivector" is a 3-vector. Note that for vectors higher than grade 3 there are no special names.

That is, *the inner product reduces the grade of a k-vector whereas the outer product increases it.*

Following this interpretation of inner and outer product consequently leads to the notion that a scalar is a 0-vector, because the inner product of two vectors results in a scalar. However, then we must also assert that the inner product of a scalar with a vector is identically zero.

In Section 3 it will be shown that intersections as well as the dual operation can be expressed in terms of the inner product. GC algebra lacks such a universal operator and therefore has to resort to defining a number of different inner-product-like structures.

Now we are in a position to see what the algebraic meaning of a bivector is. Let a vector $\boldsymbol{x} \in E^3$ be defined as

$$\boldsymbol{x} = \boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)$$

We can get some information about the orientation of $\boldsymbol{x}$ by calculating

$$\begin{aligned} \boldsymbol{a}\cdot\boldsymbol{x} &= \boldsymbol{a}\cdot\Big[\boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)\Big] \\ &= (\boldsymbol{a}\wedge\boldsymbol{a})\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) \quad \text{from equation (3-1)} \\ &= 0 \end{aligned} \tag{4}$$

This shows that $\boldsymbol{x}$ and $\boldsymbol{a}$ are orthogonal. Furthermore, we have

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) \\ &= (\boldsymbol{a}\cdot\boldsymbol{b}_1)\boldsymbol{b}_2 - (\boldsymbol{a}\cdot\boldsymbol{b}_2)\boldsymbol{b}_1 \end{aligned} \tag{5}$$

and hence $\boldsymbol{x}$ lies in the plane given by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Therefore, we can interpret the bivector $\boldsymbol{b}_1\wedge\boldsymbol{b}_2$ as the combination of the linear dependencies given by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Taking the inner product of $\boldsymbol{a}$ with this bivector then "takes out" the linear dependence represented by $\boldsymbol{a}$. What we are left with therefore has to be orthogonal to $\boldsymbol{a}$.

By definition the inner product is commutative and the outer product anti-commutative. GA defines another product which combines these two properties and is accordingly called the *geometric product*. In fact, it is the most fundamental operation in GA[3]. The geometric product of two vectors is written as $\boldsymbol{ab}$ and defined by

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

3 Projective Geometry

3.1 Fundamentals

Projective geometry can be expressed in terms of GA by defining a set of 4 basis vectors $\{e_1, e_2, e_3, e_4\}$ with signature $\{- - -+\}$, ie. $e_\mu\cdot e_\nu = 2\delta_{\mu 4}\delta_{\nu 4} - \delta_{\mu\nu}$. The pseudoscalar of this space is defined as,

$$I = e_1\wedge e_2\wedge e_3\wedge e_4.$$

[3]Had an axiomatic approach been followed here, the geometric product would have been the first product to be defined. The inner and outer product can then be derived from that. However, here we present a more "intuitive" introduction to GA.

The inverse pseudoscalar I^{-1} is defined such that $II^{-1} = 1$. From the metric given above it follows that $II = I^{-1}I^{-1} = -1$. Furthermore,

$$I = -I^{-1} \tag{6}$$

A vector in this 4D-space (P^3), which will be called a *homogeneous* vector[4], 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.

Let A be a homogeneous vector, i.e. $A = \alpha^\mu e_\mu$, where the $\{\alpha^\mu\}$ are some scalar components. The projection of 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. Also, an overall scalar factor of A cancels when A is projected down to 3D-space via the projective split. Therefore, if two homogeneous vectors of any grade are equal up to a scalar factor, they are identical when projected down to 3D-space. Since we are ultimately only interested in 3D-space vectors, equality up to a scalar factor is often sufficient. For that purpose we use the symbol $\simeq$. For example, $A \simeq \rho A$, where ρ is a scalar constant.

The following gives an example of the projective split. Let $A = \rho\alpha^\mu e_\mu$, where the $\{\alpha^\mu\}$ and ρ are some scalar values. Then

$$\begin{aligned} \boldsymbol{a} &= \frac{A \wedge e_4}{A \cdot e_4} \\ &= \frac{\rho\alpha^1\, e_1 \wedge e_4 \;+ \rho\alpha^2\, e_2 \wedge e_4 \;+ \rho\alpha^3\, e_3 \wedge e_4}{\rho\alpha^4} \end{aligned} \tag{7}$$

If we define a new basis $\{g_i\}$ as

$$g_i \equiv e_i \wedge e_4$$

then the vector $\boldsymbol{a}$ may be written as

$$\boldsymbol{a} = \tfrac{\alpha^i}{\alpha^4} g_i$$

The basis $\{g_i\}$ has signature $\{+++\}$, as required. This may be shown quite easily;

$$\begin{aligned} g_i \cdot g_i &= (e_i \wedge e_4) \cdot (e_i \wedge e_4) \\ &= -\, e_i \cdot e_i \; e_4 \cdot e_4 \quad \text{from equation (3-1)} \\ &= +1 \quad \text{from previously defined metric} \end{aligned} \tag{8}$$

Similarly it may be shown that $g_i \cdot g_j = 0$ if $i \neq j$. Now it is clear why the signature of the basis $\{e_\mu\}$ had to be defined as $\{---+\}$.

[4]This definition of homogeneous differs from its conventional use in GA but is here chosen to tie in with the Computer Vision convention.

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

$$I_a = A_1 \wedge A_2 \wedge A_3 \wedge A_4$$

Since I_a and I are both pseudoscalars of the same space, they can only differ by a scalar factor. That is,

$$I_a = \rho_a I \tag{9}$$

where ρ_a is the *scale* of the A-frame, given by

$$\rho_a = (A_1 \wedge A_2 \wedge A_3 \wedge A_4) I^{-1}$$

The inverses of these two pseudoscalars are related by

$$I_a^{-1} = \rho_a^{-1} I^{-1} \tag{10}$$

From equations (6), (9) and (10) it follows that

$$I_a^{-1} = -\rho_a^{-2} I_a \tag{11}$$

The outer product of a vector with a pseudoscalar is always zero. Hence, the geometric product of a vector with a pseudoscalar reduces to the inner product of the two. From this fact and with help of equation (1) the following important result follows;

$$\begin{aligned} A_\mu I_a &= A_\mu \cdot (A_1 \wedge A_2 \wedge A_3 \wedge A_4) \\ &= \sum_{\nu_1 = 1}^{4} (A_\mu \cdot A_{\nu_1})(A_{\nu_2} \wedge A_{\nu_3} \wedge A_{\nu_4}) \end{aligned} \tag{12}$$

Here, and throughout the rest of the text the $\{\nu_1, \nu_2, \nu_3, \nu_4\}$ are assumed to be an even permutation of $\{1, 2, 3, 4\}$, unless otherwise stated. Since the inner product of two vectors is a scalar, the result of this calculation is a multivector of grade 3. Similarly, the geometric product of a bivector with a pseudoscalar gives a bivector and the geometric product of a trivector with a pseudoscalar gives a vector. This introduces the concept of the *dual*.

The dual of a multivector X, written X^*, is defined as

$$X^* = X I^{-1}$$

Therefore, if X is of grade $r \leq 4$ then X^* is of grade $4 - r$. It will be extremely useful to introduce the *dual bracket* and the *inverse dual bracket*. To a certain extent they are related to the bracket notation as used in GC algebra and GA[5]. There the bracket of a pseudoscalar P, say, is a scalar, defined as the dual of P in GA. That is, $[P] = PI^{-1}$; here however the dual bracket concept can produce something other than a scalar.

The dual bracket is defined as

$$[\![A_{\mu_1} A_{\mu_2} \cdots A_{\mu_n}]\!]_a \equiv (A_{\mu_1} \wedge A_{\mu_2} \wedge \ldots \wedge A_{\mu_n}) I_a^{-1} \tag{13-1}$$

$$[\![A_{\mu_1} A_{\mu_2} \cdots A_{\mu_n}]\!] \equiv (A_{\mu_1} \wedge A_{\mu_2} \wedge \ldots \wedge A_{\mu_n}) I^{-1} \tag{13-2}$$

[5]See, for example [6]

The inverse dual bracket is defined as

$$\langle\!\langle A_{\mu_1} A_{\mu_2} \cdots A_{\mu_n} \rangle\!\rangle_a \equiv (A_{\mu_1} \wedge A_{\mu_2} \wedge \ldots \wedge A_{\mu_n}) I_a \tag{14-1}$$

$$\langle\!\langle A_{\mu_1} A_{\mu_2} \cdots A_{\mu_n} \rangle\!\rangle \equiv (A_{\mu_1} \wedge A_{\mu_2} \wedge \ldots \wedge A_{\mu_n}) I \tag{14-2}$$

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 $[\![A_1 A_2 A_3 A_4]\!] = [\![P]\!] = [P] = \rho_a$. Furthermore, the following identities hold:

$$\langle\!\langle X \rangle\!\rangle = -[\![X]\!] \tag{15-1}$$

$$[\![X]\!] = \rho_a [\![X]\!]_a \tag{15-2}$$

$$\langle\!\langle X \rangle\!\rangle = \rho_a^{-1} \langle\!\langle X \rangle\!\rangle_a \tag{15-3}$$

$$[\![X]\!]_a = -\rho_a^{-2} \langle\!\langle X \rangle\!\rangle_a \tag{15-4}$$

$$\langle\!\langle [\![X]\!] \rangle\!\rangle = [\![\langle\!\langle X \rangle\!\rangle]\!] = X \tag{15-5}$$

$$[\![[\![X]\!]]\!] = \langle\!\langle \langle\!\langle X \rangle\!\rangle \rangle\!\rangle = -X \tag{15-6}$$

There is another useful identity;

$$\begin{aligned} [\![A_{\mu_1} A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot I^{-1} \\ &= A_{\mu_1} \cdot \big((A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot I^{-1} \big) \\ &= A_{\mu_1} \cdot [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] \end{aligned} \tag{16}$$

Similarly it may be shown that

$$\begin{aligned} [\![A_{\mu_1} A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] &= (A_{\mu_1} \wedge A_{\mu_2}) \cdot [\![A_{\mu_3} A_{\mu_4}]\!] \\ &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3}) \cdot [\![A_{\mu_4}]\!] \\ &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot [\![1]\!] \end{aligned} \tag{17}$$

Note that $[\![1]\!] = I^{-1}$. The same identities also apply for the $[\![\cdots]\!]_a$ type brackets. Put simply, vectors may be "pulled" out of a dual bracket (or inverse dual bracket) by taking the inner product of them with the remainder of the bracket.

3.2 Reciprocal Vector Frames

It is now straightforward to define *reciprocal frames*. From equation (16) it follows that

$$\begin{aligned}
& [\![\langle\!\langle 1 \rangle\!\rangle_a]\!]_a = 1 \\
\Rightarrow\quad & [\![A_1 A_2 A_3 A_4]\!]_a = 1 \\
\Longleftrightarrow\quad & A_{\mu_1} \cdot [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!]_a = 1 \\
\Longleftrightarrow\quad & A_{\mu_1} \cdot A_a^{\mu_1} = 1
\end{aligned} \tag{18}$$

with no implicit summation over the range of μ_1. This defines the *normalized reciprocal A-frame*, written $\{A_a^{\mu}\}$, 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*.

$$A^{\mu_1} = [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!]$$

The relation between $A_a^{\mu_1}$ and A^{μ_1} is

$$A_a^{\mu_1} = \rho_a^{-1} A^{\mu_1} \tag{19}$$

From these definitions of reciprocal frame vectors it follows that

$$A_\mu \cdot A_a^\nu \quad = \quad \delta_\mu^\nu \tag{20-1}$$

$$A_\mu \cdot A^\nu \quad = \quad \rho_a \delta_\mu^\nu \tag{20-2}$$

where δ_μ^ν is the Kronecker delta. That is, a reciprocal frame vector is nothing else but the dual of a plane. It may therefore also be regarded as the *normal* of the plane that is its dual.

In GC algebra these reciprocal vectors would be defined as elements of a *dual space*, which is indeed what is done in [3]. However, because GC algebra does not define an inner product explicitly as in GA, elements of this dual space cannot operate on elements of the "normal" space. Hence, the concept of reciprocal frames cannot be defined in GC algebra.

A reciprocal frame can be used to transform a vector from one frame into another. That is,

$$X = (X \cdot A_a^\mu) A_\mu = (X \cdot A_\nu) A_a^\nu \tag{21}$$

To show the first part of this equation we can say that since the $\{A_\mu\}$ form a basis of P^3, X can be given in terms of that frame as $X = \alpha^\mu A_\mu$, where the $\{\alpha^\mu\}$ are some scalars. Then, with use of equation (20-1)

$$X \cdot A_a^\nu = \alpha^\mu (A_\mu \cdot A_a^\nu) = \alpha^\mu \delta_\mu^\nu = \alpha^\nu$$

from which the first part of equation (21) follows.

Note that the $\{A_a^\mu\}$ also form a basis of P^3 since $A_a^1 \wedge A_a^2 \wedge A_a^3 \wedge A_a^4 \neq 0$. Therefore, X can also be given as $X = \alpha_\mu A_a^\mu$, where the $\{\alpha_\mu\}$ are a set of scalars different to the $\{\alpha^\mu\}$. Then, using again equation (20-1)

$$X \cdot A_\nu = \alpha_\mu (A_a^\mu \cdot A_\nu) = \alpha_\mu \delta_\nu^\mu = \alpha_\nu$$

and hence the second part of equation (21).

3.3 Reciprocal Line Frames

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 $L_a^{i_1} = A_{i_2} \wedge A_{i_3}$. Once again, the $\{i_1, i_2, i_3\}$ are assumed to be an even permutation of $\{1, 2, 3\}$. A *reciprocal line frame* can then be defined as follows, again by using the identities in equation (17)

$$\begin{array}{rl} & [\![\langle\!\langle 1 \rangle\!\rangle_a]\!]_a = 1 \\ \Rightarrow & [\![A_1 A_2 A_3 A_4]\!]_a = 1 \\ \Longleftrightarrow & (A_{i_1} \wedge A_{i_2}) \cdot [\![A_{i_3} A_4]\!]_a = 1 \\ \Longleftrightarrow & L_a^{i_3} \cdot \bar{L}_{i_3}^a = 1 \end{array} \tag{22}$$

This[6] defines the *normalised reciprocal A-line frame* $\{\bar{L}_i^a\}$ and the *standard reciprocal A-line frame* $\{L_i^a\}$ as

$$\bar{L}_i^a = [\![A_i A_4]\!]_a \tag{23-1}$$

$$L_i^a = [\![A_i A_4]\!] \tag{23-2}$$

Hence,

$$L_a^i \cdot \bar{L}_j^a = \delta_j^i \tag{24-1}$$

$$L_a^i \cdot L_j^a = \rho_a \delta_j^i \tag{24-2}$$

Again, this shows the universality of the inner product: bivectors can be treated in the same fashion as vectors.

Note that L_a^i can also be expressed in the following way,

$$\begin{aligned} L_a^{i_1} &= A_{i_2} \wedge A_{i_3} \\ &= (A_{i_2} \wedge A_{i_3}) I_a^{-1} I_a \quad \text{since } I_a^{-1} I_a = 1 \\ &= \Big[(A_{i_2} \wedge A_{i_3}) \cdot (A_a^4 \wedge A_a^3 \wedge A_a^2 \wedge A_a^1)\Big] I_a \\ &= -(A_a^{i_1} \wedge A_a^4) I_a \\ &= -\langle\!\langle A_a^{i_1} A_a^4 \rangle\!\rangle_a \\ &\simeq \langle\!\langle A_a^{i_1} A_a^4 \rangle\!\rangle \end{aligned} \tag{25}$$

This particular form of L_a^i will become useful later on.

3.4 Meet and Join

The *meet* and *join* are the two operations needed to calculate intersections between two lines, two planes or a line and a plane. In general terms the join is the sum and the meet is the intersection of two spaces. In GA any blade can be treated as a pseudoscalar of a particular subspace.

[6]Note how similar this derivation is to that of reciprocal vector frames (equation (18)).

The join of two blades A and B, written as $A \triangle B$ can be defined in general as the pseudoscalar of the space given by the sum of the spaces spanned by A and B. For example, if $A = e_1 \wedge e_2$ and $B = e_2 \wedge e_3$ then $A \triangle B = e_1 \wedge e_2 \wedge e_3$.

The meet of A and B, written as $A \vee B$, is defined to give the space that A and B have in common. Using the definitions of A and B from the previous example $A \vee B \simeq e_2$. In general, the following expression for the meet can be given. Let A and B be two arbitrary multivectors, and let $J = A \triangle B$, then

$$A \vee B = \Big[(AJ^{-1}) \wedge (BJ^{-1})\Big] J \tag{26}$$

For the intersection of two planes or a plane and a line in P^3, the join will always be the pseudoscalar I, unless the line lies on the plane or the two planes are the same. In the following we will assume that this is not the case. Then, for intersections between two planes or a plane and a line equation (26) may be written as

$$\begin{aligned} A \vee B &= \langle\!\langle [\![A]\!][\![B]\!] \rangle\!\rangle \\ &= [\![A]\!] \cdot \langle\!\langle [\![B]\!] \rangle\!\rangle \quad \text{from equation (17)} \\ &= [\![A]\!] \cdot B \quad \text{from equation (15-5)} \end{aligned} \tag{27}$$

More details about meet and join may be found in [8] and [6].

There is a particularly nice feature of the meet operation which is worth mentioning here: *a vector is transformed into a particular frame by "meeting" it with the pseudoscalar of that frame.* The proof of this statement relies on the fact that the operation $X \vee I_a$ can be expanded in three different ways. First of all

$$\begin{aligned} X \vee I_a &= \Big((XI^{-1}) \wedge \underbrace{(I_a I^{-1})}_{\text{scalar}}\Big) I \\ &= XI^{-1} I_a I^{-1} I \\ &= \rho_a X \end{aligned} \tag{28}$$

It may be shown with an analysis similar to the one used in equations (2) that $XI = -IX$ and simlilarly $XI^{-1} = -I^{-1}X$. Also, $I_a I^{-1} = I^{-1} I_a$. Using these facts $X \vee I_a$ can also be expanded as

$$\begin{aligned} X \vee I_a &= XI^{-1} I_a \\ &= -I_a X I^{-1} \\ &= -(A_1 \wedge A_2 \wedge A_3 \wedge A_4) \cdot [\![X]\!] \\ &= -A_1 [\![A_2 A_3 A_4 X]\!] + A_2 [\![A_1 A_3 A_4 X]\!] \\ &\quad -A_3 [\![A_1 A_2 A_4 X]\!] + A_4 [\![A_1 A_2 A_3 X]\!] \end{aligned} \tag{29}$$

Writing this as the sum over an index gives

$$\begin{aligned} X \vee I_a &= \sum_{\mu_1} [\![X A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] A_{\mu_1} \\ &= \sum_{\mu_1} \Big[X \cdot [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!]\Big] A_{\mu_1} \\ &= X \cdot A^{\mu_1} A_{\mu_1} \end{aligned} \tag{30}$$

Similarly

$$
\begin{aligned}
X \vee I_a &= -I^{-1} X I_a \\
&= -I^{-1}(X \cdot I_a) \\
&= \sum_{\mu_1} -I^{-1}\, X \cdot A_{\mu_1}\, (A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \\
&= X \cdot A_{\mu_1}\, A^{\mu_1}
\end{aligned}
\tag{31}
$$

Equating equations (28), (30) and (31) gives

$$
X = (\, X \cdot A_a^{\mu}\,) A_{\mu} = (\, X \cdot A_{\nu}\,) A_a^{\nu}
$$

which is the same as equation (21).

3.5 Cameras and Projections

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.

Projection of some point X onto the image plane is done by intersecting the line connecting the optical centre with X, with the image plane. Intersections are calculated with the *meet* operation. As an example, consider a camera defined by the A-frame. The line connecting some point X with the optical centre is then given by $X \wedge A_4$, and the image plane of the camera is given by $(A_1 \wedge A_2 \wedge A_3)$. Therefore, the projection of X onto the image plane is given using equations (17) and (27) by

$$
\begin{aligned}
(X \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) &= [\![X A_4]\!] \cdot (A_1 \wedge A_2 \wedge A_3) \\
&= \sum_{i_3} [\![X A_{i_1} A_{i_2} A_4]\!] A_{i_3} \\
&= \sum_{i_3} \Big[X \cdot [\![A_{i_1} A_{i_2} A_4]\!]\Big] A_{i_3} \\
&= (\, X \cdot A^{i}\,) A_i
\end{aligned}
\tag{32}
$$

Suppose that X is given in some frame $\{Z_\mu\}$ as $X = \zeta^{\mu} Z_{\mu}$. Then the projection X_a of X onto the A-image plane can be written as

$$
\begin{aligned}
X_a &= (\, X \cdot A^{i}\,) A_i \\
&= (\zeta^{\mu}\, Z_{\mu} \cdot A^{i}\,) A_i \\
&= \zeta^{\mu} K_{i_{\mu}} A_i\,; \qquad K_{i_{\mu}} \equiv Z_{\mu} \cdot A^{i}
\end{aligned}
\tag{33}
$$

The matrix $K_{i_{\mu}}$ is the *camera matrix*[7] of camera A, for projecting points given in the Z-frame onto the A-image plane.

[7]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.

In [3] 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, 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

$$(L \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) = (\, L \cdot L_i^a \,) L_a^i \tag{34}$$

4 The Trifocal Tensor

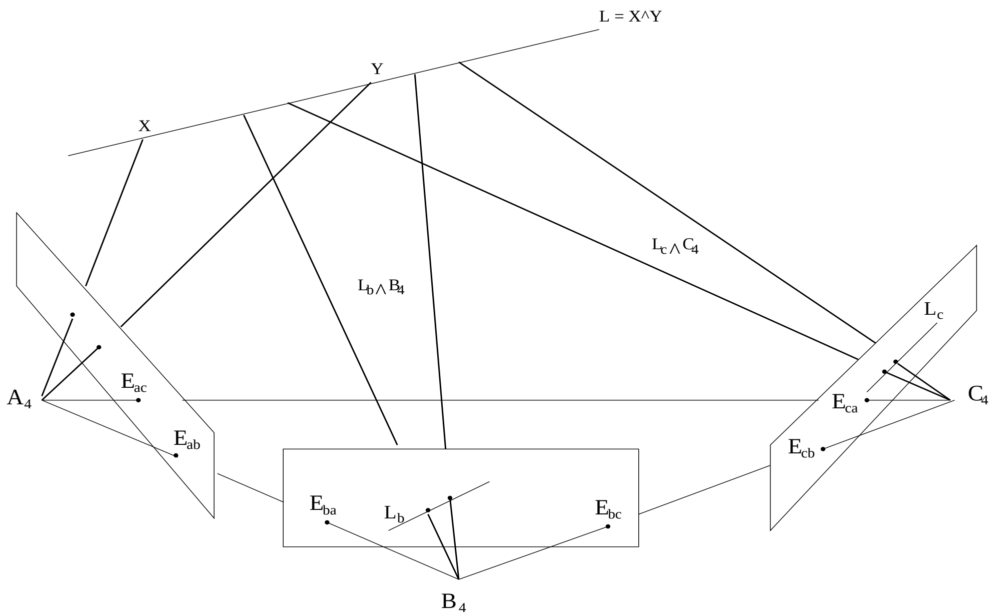

Figure 1: Line projected onto three image planes. Note that although the figure is drawn in E^3, lines and points are denoted by their corresponding vectors in P^3.

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. Therefore, $(\lambda_i^b \langle\!\langle B^i \rangle\!\rangle) \vee (\lambda_j^c \langle\!\langle C^j \rangle\!\rangle)$ has to give L up to a scalar factor. Now, if two lines intersect, their outer product is zero. Thus, the outer product of lines $X \wedge A_4$ (or $Y \wedge A_4$) and L has to be zero. Note that $X \wedge A_4$ defines the same line as $(\alpha^i A_i) \wedge A_4$, up to a scalar factor, where $\alpha^i = X \cdot A^i$. Figure 1 shows this construction. Combining all these expressions gives

$$\begin{aligned} 0 &= (X \wedge A_4 \wedge L) I^{-1} \\ &= \alpha^i \lambda_j^b \lambda_k^c \left[\!\left[(A_i \wedge A_4)(\langle\!\langle B^j \rangle\!\rangle \vee \langle\!\langle C^k \rangle\!\rangle) \right]\!\right] \\ &= \alpha^i \lambda_j^b \lambda_k^c \left[\!\left[(A_i \wedge A_4) \langle\!\langle B^j C^k \rangle\!\rangle \right]\!\right] \end{aligned} \tag{35}$$

where the following identity was used which can be derived with the help of equation (27);

$$\begin{aligned} \langle\!\langle B^j \rangle\!\rangle \vee \langle\!\langle C^k \rangle\!\rangle &= \left\langle\!\!\left\langle [\![\langle\!\langle B^j \rangle\!\rangle]\!] [\![\langle\!\langle C^k \rangle\!\rangle]\!] \right\rangle\!\!\right\rangle \\ &= \langle\!\langle B^j C^k \rangle\!\rangle \end{aligned} \tag{36}$$

If the trifocal tensor $T_{i}{}^{jk}$ is defined as

$$T_{i}{}^{jk} = \left[\!\!\left[(A_i \wedge A_4) \langle\!\langle B^j C^k \rangle\!\rangle \right]\!\!\right] \tag{37}$$

then, from equation (35) it follows that it has to satisfy $\alpha^i \lambda_j^b \lambda_k^c T_{i}{}^{jk} = 0$. This expression for the trifocal tensor can be expanded in two different, but equivalent ways. The first way yields,

$$\begin{aligned} T_{i}{}^{jk} &= (A_i \wedge A_4) \cdot [\![\langle\!\langle B^j C^k \rangle\!\rangle]\!] \\ &= (A_i \wedge A_4) \cdot (B^j \wedge C^k) \\ &= (A_4 \cdot B^j)(A_i \cdot C^k) - (A_4 \cdot C^k)(A_i \cdot B^j) \\ &= K^b_{j_4} K^c_{k_i} - K^c_{k_4} K^b_{j_i} \end{aligned} \tag{38}$$

where $K^b_{j_i} \equiv A_i \cdot B^j$ and $K^c_{k_i} \equiv A_i \cdot C^k$ are the camera matrices for cameras B and C, respectively, relative to camera A. This is the expression for the trifocal tensor given by Hartley in [1]. Note that the camera matrix for the A-plane would be written as $K^a_{j_\mu} \equiv A_\mu \cdot A^j \simeq \delta_i^j$. That is, $K^a = [I|0]$ in standard matrix notation. In many other derivations of the trifocal tensor (eg. [1]) this form of the camera matrices is assumed at the beginning. Here, however, the trifocal tensor is defined first geometrically and we then find that it implies this particular form for the camera matrices.

On the other hand, equation (37) can also be expanded to

$$\begin{aligned} T_{i}{}^{jk} &= [\![A_i A_4]\!] \cdot \langle\!\langle B^j C^k \rangle\!\rangle \\ &= L_i^a \cdot \langle\!\langle B^j C^k \rangle\!\rangle \quad \text{from equation (23-2)} \end{aligned} \tag{39}$$

This expression for the trifocal tensor is somewhat more instructive than the previous one. Recall that $\lambda_j^b \lambda_k^c \langle\!\langle B^j C^k \rangle\!\rangle$ gives line L up to a scalar factor. From equation (34) it then becomes clear that $\lambda_j^b \lambda_k^c T_{i}{}^{jk}$ gives the components of the projection of line L onto image plane A, up to a scalar factor. Alternatively, let $T^{jk} = \langle\!\langle B^j C^k \rangle\!\rangle$. Then the projection of line T^{jk} onto image plane A (from equation (34)), denoted by T_a^{jk} is

$$T_a^{jk} \equiv T_{i}{}^{jk} L_a^i \tag{40}$$

Since epipoles are not essential in this report only a short definition will be given here. More details may be found in [6].

An epipole is the projection of the optical centre of one camera onto the image plane of another. For example, the epipole E_{ba} is the projection of the optical centre of camera A (A_4) onto the image plane of camera B ($B_1 \wedge B_2 \wedge B_3$). That is, from equation (32)

$$\begin{aligned} E_{ba} &= (A_4 \wedge B_4) \vee (B_1 \wedge B_2 \wedge B_3) \\ &= (\, A_4 \cdot B^i \,) B_i \end{aligned} \tag{41}$$

From the definition of the camera matrices as given in equation (33) and equation (38) it then follows that

$$E_{ba} = K^{b}_{i_4} B_i$$

In other words, the fourth column of the camera matrix gives the coordinates of an epipole.

5 Constraints on the Trifocal Tensor

By transforming the trifocal tensor into an epipolar basis, it can be shown quite easily (see [7]) that the trifocal tensor only has 18 degrees of freedom (DOF). This also yields a minimal parameterisation of the trifocal tensor in term of its epipoles. Nevertheless, this approach has two big problems. Firstly, the epipoles are only known once the trifocal tensor has been calculated. Secondly, preliminary attempts have shown that this parameterisation is very non-linear. That is, a tiny change in the value of one epipole appears to result in a large change in the components of the full trifocal tensor. Therefore, an iterative minimisation routine that tries to find the correct epipolar values, would have to search over a very non-linear surface in 18 dimensions. Nonetheless, the epipolar parameterisation is an easy way to prove that the trifocal tensor has indeed only 18 DOF.

A potentially better approach for calculating the trifocal tensor is to use all 27 components as free variables, but to restrain the whole system through some additional constraints. These constraints have to define the structure of the trifocal tensor without depending on any values other than its components.

Such constraints are derived here following the approach given in [3]. However, not only has this approach been generalized but the arguments used are also of purely geometrical origin. In particular, the derivation given here 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^{jk}_a. Relations between the T^{jk}_a can in turn be directly related to the components of the trifocal tensor. There are two types of constraints.

5.1 Constraint Type 1

In the following, the $\{i_1, i_2, i_3\}$, etc. are no longer assumed to be any particular kind of permutation.

The constraints we are looking for somehow have to relate the lines $\{T^{ij}\}$. Finding relations between the intersection points of these lines seems to be a promising idea. However, there is no guarantee that any two lines of the set $\{T^{ij}\}$ do intersect, i.e. are co-planar. Therefore, it is better to find the intersection between a plane $A_4 \wedge T^{i_1 j_1}$ and a line $T^{i_2 j_2}$ which is always well defined, as long as A_4 does not lie on the line $T^{i_1 j_1}$. In the following we will assume that $A_4 \wedge T^{i_1 j_1} \neq 0$.

To simplify the notation, the intersection between $A_4 \wedge T^{i_1 j_1}$ and $T^{i_2 j_2}$ is written as $p(i_1 j_1, i_2 j_2)$ and given by

$$
\begin{aligned}
p(i_1 j_1, i_2 j_2) &\equiv (A_4 \wedge \langle\!\langle B^{i_1} C^{j_1} \rangle\!\rangle) \vee \langle\!\langle B^{i_2} C^{j_2} \rangle\!\rangle \\
&= \left\langle\!\!\left\langle \left[\!\left[A_4 \langle\!\langle B^{i_1} C^{j_1} \rangle\!\rangle \right]\!\right] \left[\!\left[\langle\!\langle B^{i_2} C^{j_2} \rangle\!\rangle \right]\!\right] \right\rangle\!\!\right\rangle \\
&= \left\langle\!\!\left\langle \Big(A_4 \cdot \left[\!\left[\langle\!\langle B^{i_1} C^{j_1} \rangle\!\rangle \right]\!\right] \Big) B^{i_2} C^{j_2} \right\rangle\!\!\right\rangle \\
&= \left\langle\!\!\left\langle \Big(A_4 \cdot (B^{i_1} \wedge C^{j_1}) \Big) B^{i_2} C^{j_2} \right\rangle\!\!\right\rangle \\
&= \left\langle\!\!\left\langle (A_4 \cdot B^{i_1}) C^{j_1} B^{i_2} C^{j_2} - (A_4 \cdot C^{j_1}) B^{i_1} B^{i_2} C^{j_2} \right\rangle\!\!\right\rangle \\
&= \varepsilon_{ba}^{i_1} \langle\!\langle C^{j_1} B^{i_2} C^{j_2} \rangle\!\rangle + \varepsilon_{ca}^{j_1} \langle\!\langle B^{i_1} C^{j_2} B^{i_2} \rangle\!\rangle
\end{aligned}
\tag{42}
$$

where $\varepsilon_{ba}^{i} \equiv A_4 \cdot B^i$ and $\varepsilon_{ca}^{i} \equiv A_4 \cdot C^i$ are the image point coordinates for epipoles E_{ba} and E_{ca}, respectively.

Consider the following types of intersection points.

$$
p(i_1 j, i_2 j) = \varepsilon_{ca}^{j} \langle\!\langle B^{i_1} C^{j} B^{i_2} \rangle\!\rangle \tag{43-1}
$$

$$
p(i j_1, i j_2) = \varepsilon_{ba}^{i} \langle\!\langle C^{j_1} B^{i} C^{j_2} \rangle\!\rangle \tag{43-2}
$$

Using just this type of intersection point a very simple constraint can be found. First of all consider

$$
\begin{aligned}
p(i_1 j_1, i_2 j_1) \wedge p(i_1 j_2, i_2 j_2) &= \varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \underbrace{\langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle}_{\text{grade 1 vector}} \wedge \underbrace{\langle\!\langle B^{i_1} C^{j_2} B^{i_2} \rangle\!\rangle}_{\text{grade 1 vector}} \\
&= \varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \Big(\langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \wedge \langle\!\langle B^{i_1} C^{j_2} B^{i_2} \rangle\!\rangle \Big) I^{-1} I \\
&= \varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \Big(\underbrace{\langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle}_{\text{grade 1 vector}} \cdot \underbrace{(B^{i_1} \wedge C^{j_2} \wedge B^{i_2})}_{\text{grade 3 vector}} \Big) I
\end{aligned}
\tag{44}
$$

Using equation (16) we get

$$
\begin{aligned}
p(i_1 j_1, i_2 j_1) \wedge p(i_1 j_2, i_2 j_2) &= \varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \Big(\quad B^{i_1} \cdot \langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \, (C^{j_2} \wedge B^{i_2}) \\
&\qquad - \quad C^{j_2} \cdot \langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \, (B^{i_1} \wedge B^{i_2}) \\
&\qquad + \quad B^{i_2} \cdot \langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \, (B^{i_1} \wedge C^{j_2}) \Big) I \\
&= \varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \Big(\quad \underbrace{\langle\!\langle B^{i_1} B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle}_{=0} \langle\!\langle C^{j_2} B^{i_2} \rangle\!\rangle \\
&\qquad - \quad \langle\!\langle C^{j_2} B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \\
&\qquad + \quad \underbrace{\langle\!\langle B^{i_2} B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle}_{=0} \langle\!\langle B^{i_1} C^{j_2} \rangle\!\rangle \Big) \\
&= -\varepsilon_{ca}^{j_1} \varepsilon_{ca}^{j_2} \underbrace{\langle\!\langle B^{i_1} B^{i_2} C^{j_1} C^{j_2} \rangle\!\rangle}_{\text{scalar}} \langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle
\end{aligned}
\tag{45}
$$

Note that only the term $\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle$ is not a scalar. Following a similar analysis it can be shown that

$$
\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \wedge p(i_1 j_3, i_2 j_3) = \varepsilon_{ca}^{j_3} \langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \wedge \langle\!\langle B^{i_1} C^{j_3} B^{i_2} \rangle\!\rangle = 0 \tag{46}
$$

Therefore,

$$\boxed{p(i_1j_1, i_2j_1) \wedge p(i_1j_2, i_2j_2) \wedge p(i_1j_3, i_2j_3) = 0} \tag{47}$$

and similarly

$$\boxed{p(i_1j_1, i_1j_2) \wedge p(i_2j_1, i_2j_2) \wedge p(i_3j_1, i_3j_2) = 0} \tag{48}$$

These two constraints simply express the fact that all three intersection points (all the p's) lie on the same line. It is fairly simple to see which line that is. Just as $\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle$ is the intersection between planes $\langle\!\langle B^{i_1} \rangle\!\rangle$ and $\langle\!\langle B^{i_2} \rangle\!\rangle$, $\langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle$ is the intersection between the three planes $\langle\!\langle B^{i_1} \rangle\!\rangle$, $\langle\!\langle B^{i_2} \rangle\!\rangle$ and $\langle\!\langle C^{j_1} \rangle\!\rangle$. Therefore, equation (47) can also be written as

$$\Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_1} \rangle\!\rangle\Big) \wedge \Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_2} \rangle\!\rangle\Big) \wedge \Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_3} \rangle\!\rangle\Big) = 0 \tag{49}$$

That is, we take the outer product of the intersection points of line $\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle$ with the planes $\langle\!\langle C^{j_1} \rangle\!\rangle$, $\langle\!\langle C^{j_2} \rangle\!\rangle$ and $\langle\!\langle C^{j_3} \rangle\!\rangle$. Obviously all three intersection points have to lie on line $\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle$, hence their outer product is zero. This construction is shown in figure 2.

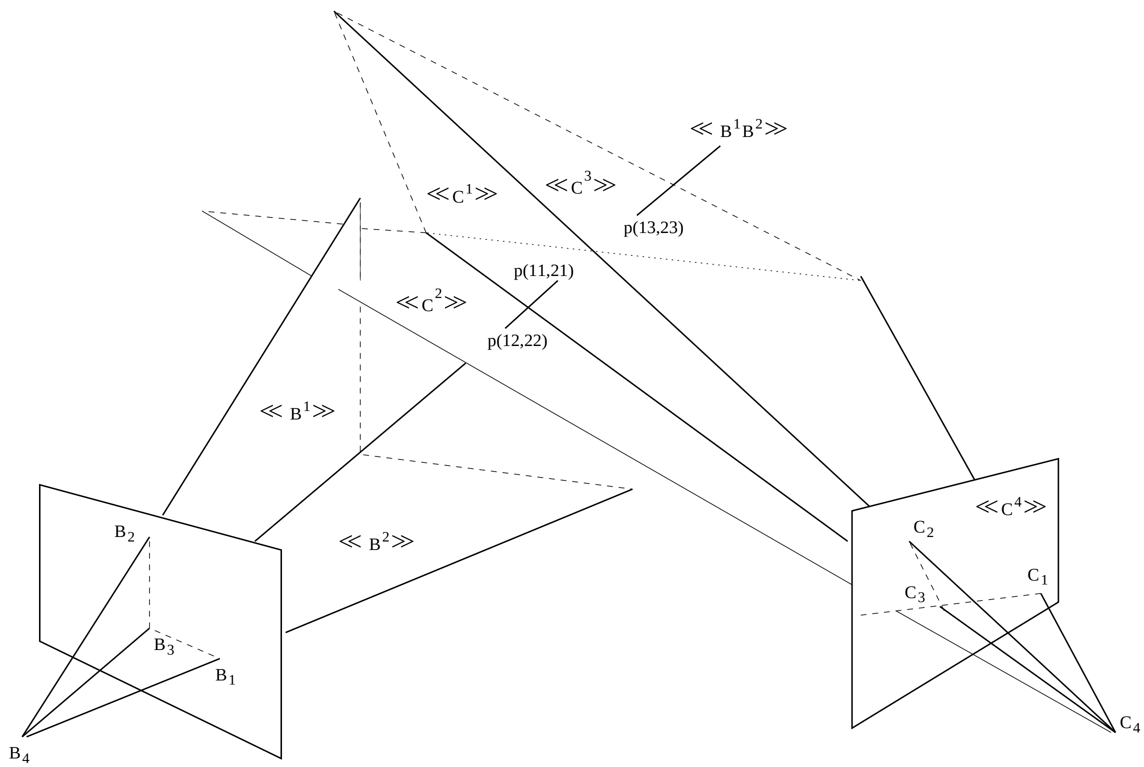

Figure 2: This demonstrates the constraint type 1 for $i_1 = 1$, $i_2 = 2$ and $j_1 = 1$, $j_2 = 2$, $j_3 = 3$. For example, $p(12, 22) \simeq \langle\!\langle B^1 C^2 B^2 \rangle\!\rangle \simeq \langle\!\langle B^1 B^2 \rangle\!\rangle \vee \langle\!\langle C^2 \rangle\!\rangle$.

These constraints also have to hold for the intersection points of the projected lines T_a^{jk}. Let the intersection between lines $T_a^{j_1k_1}$ and $T_a^{j_2k_2}$ be written as $p_a(j_1k_1, j_2k_2)$. This intersection point lies on the A-image plane by definition, and can therefore be given in the A-line basis (equation (40)). With the help of equation (25) we get

$$
\begin{aligned}
p_a(j_1k_1, j_2k_2) &\equiv T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} (A_4 \wedge L_a^{i_1}) \vee L_a^{i_2} \\
&= T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} (A_4 \wedge \langle\!\langle A_a^{i_1} A_a^4 \rangle\!\rangle) \vee \langle\!\langle A_a^{i_2} A_a^4 \rangle\!\rangle \\
&= T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} \left\langle\!\!\left\langle \left(A_4 \cdot (A_a^{i_1} \wedge A_a^4) \right) A_a^{i_2} A_a^4 \right\rangle\!\!\right\rangle \\
&\simeq T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} \langle\!\langle A_a^{i_1} A_a^{i_2} A_a^4 \rangle\!\rangle
\end{aligned} \tag{50}
$$

following a similar analysis as in equation (45) it is possible to show that

$$
\begin{aligned}
p_a(j_1k_1, j_2k_2) \wedge p_a(j_3k_3, j_4k_4) \simeq\ & T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} T_{i_3 j_3 k_3} \langle\!\langle A_a^{i_1} A_a^{i_2} A_a^{i_3} A_a^4 \rangle\!\rangle T_{i_4 j_4 k_4} \langle\!\langle A_a^{i_4} A_a^4 \rangle\!\rangle \\
& - T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} T_{i_4 j_4 k_4} \langle\!\langle A_a^{i_1} A_a^{i_2} A_a^{i_4} A_a^4 \rangle\!\rangle T_{i_3 j_3 k_3} \langle\!\langle A_a^{i_3} A_a^4 \rangle\!\rangle
\end{aligned} \tag{51}
$$

From the definition of the angle bracket it follows that for any scalar components $\{\alpha_i\}$, $\{\beta_j\}$ and $\{\eta_k\}$

$$
\begin{aligned}
\alpha_i \beta_j \eta_k \langle\!\langle A_a^i A_a^j A_a^k A_a^4 \rangle\!\rangle_a &= \alpha_i \beta_j \eta_k \epsilon_{ijk} \\
&= \det(\alpha_i, \beta_j, \eta_k)_{ijk}
\end{aligned} \tag{52}
$$

where ϵ_{ijk} is $+1$ if $\{ijk\}$ form an even permutation of $\{1,2,3\}$, -1 if they form an odd permutation, and 0 if any two indices are equal. $\det(\alpha_i, \beta_j, \eta_k)_{ijk}$ denotes the determinant of a matrix with rows given by $\{\alpha_i\}$, $\{\beta_j\}$ and $\{\eta_k\}$ in exactly that order from top to bottom. The subscript gives the indices that are used to form the matrix rows. If the $\{\alpha_i\}$, $\{\beta_j\}$ and $\{\eta_k\}$ are written as vectors $\boldsymbol{a} = \alpha_i e^i$, $\boldsymbol{b} = \beta_j e^j$ and $\boldsymbol{c} = \eta_k e^k$ then we define

$$
\begin{aligned}
\det(\alpha_i, \beta_j, \eta_k)_{ijk} &\equiv \det(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \\
&\equiv |\boldsymbol{abc}|
\end{aligned} \tag{53}
$$

Therefore,

$$
\begin{aligned}
T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} T_{i_3 j_3 k_3} \langle\!\langle A_a^{i_1} A_a^{i_2} A_a^{i_3} A_a^4 \rangle\!\rangle_a &= \det(T_{i_1 j_1 k_1} T_{i_2 j_2 k_2} T_{i_3 j_3 k_3})_{i_1 i_2 i_3} \\
&\equiv |T_a^{j_1 k_1} T_a^{j_2 k_2} T_a^{j_3 k_3}|
\end{aligned} \tag{54}
$$

Using this notation, equation (51) may be written more concisely as,

$$
\begin{aligned}
p_a(j_1k_1, j_2k_2) \wedge p_a(j_3k_3, j_4k_4) \simeq\ & |T_a^{j_1 k_1} T_a^{j_2 k_2} T_a^{j_3 k_3}| T_{i_4 j_4 k_4} L_a^{i_4} \\
& - |T_a^{j_1 k_1} T_a^{j_2 k_2} T_a^{j_4 k_4}| T_{i_3 j_3 k_3} L_a^{i_3}
\end{aligned} \tag{55}
$$

Therefore, expressing equation (47) in terms of the p_a gives,

$$
\begin{aligned}
0 =\ & p_a(j_1k_1, j_2k_1) \wedge p_a(j_1k_2, j_2k_2) \wedge p_a(j_1k_3, j_2k_3) \\
=\ & |T_a^{j_1 k_1} T_a^{j_2 k_1} T_a^{j_1 k_2}|\ |T_a^{j_2 k_2} T_a^{j_1 k_3} T_a^{j_2 k_3}| \\
-\ & |T_a^{j_1 k_1} T_a^{j_2 k_1} T_a^{j_2 k_2}|\ |T_a^{j_1 k_2} T_a^{j_1 k_3} T_a^{j_2 k_3}|
\end{aligned} \tag{56}
$$

and the constraint in equation (48) becomes,

$$
\begin{aligned}
0 &= p_a(j_1k_1, j_1k_2) \wedge p_a(j_2k_1, j_2k_2) \wedge p_a(j_3k_1, j_3k_2) \\
&= |T_a^{j_1k_1} T_a^{j_1k_2} T_a^{j_2k_1}|\ |T_a^{j_2k_2} T_a^{j_3k_1} T_a^{j_3k_2}| \\
&- |T_a^{j_1k_1} T_a^{j_1k_2} T_a^{j_2k_2}|\ |T_a^{j_2k_1} T_a^{j_3k_1} T_a^{j_3k_2}|
\end{aligned}
\tag{57}
$$

5.2 Constraint Type 2

The second type of constraint is slightly more complicated. Here, the following type of intersection point is needed

$$
\begin{aligned}
p(i_1j_1, i_2j_2) &= \varepsilon^{i_1}_{ba} \langle\!\langle C^{j_1} B^{i_2} C^{j_2} \rangle\!\rangle + \varepsilon^{j_1}_{ca} \langle\!\langle B^{i_1} C^{j_2} B^{i_2} \rangle\!\rangle \\
p(i_1j_2, i_2j_1) &= \varepsilon^{i_1}_{ba} \langle\!\langle C^{j_2} B^{i_2} C^{j_1} \rangle\!\rangle + \varepsilon^{j_2}_{ca} \langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle
\end{aligned}
$$

Therefore,

$$
p(i_1j_1, i_2j_2) + p(i_1j_2, i_2j_1) = \varepsilon^{j_1}_{ca} \langle\!\langle B^{i_1} C^{j_2} B^{i_2} \rangle\!\rangle + \varepsilon^{j_2}_{ca} \langle\!\langle B^{i_1} C^{j_1} B^{i_2} \rangle\!\rangle \tag{59}
$$

Comparing this with equation (45) it can be seen right away that as in equation (46) the following has to be true

$$
\boxed{p(i_1j_1, i_2j_1) \wedge p(i_1j_2, i_2j_2) \wedge \Big(p(i_1j_1, i_2j_2) + p(i_1j_2, i_2j_1)\Big) = 0} \tag{60}
$$

This constraint simply states that the point $(p(i_1j_1, i_2j_2) + p(i_1j_2, i_2j_1))$ lies on the line $p(i_1j_1, i_2j_1) \wedge p(i_1j_2, i_2j_2)$. Or, writing equation (60) in terms of intersections of lines and planes

$$
\Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_1} \rangle\!\rangle\Big) \wedge \Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_2} \rangle\!\rangle\Big) \wedge \Big(\langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_1} \rangle\!\rangle + \langle\!\langle B^{i_1} B^{i_2} \rangle\!\rangle \vee \langle\!\langle C^{j_2} \rangle\!\rangle\Big) = 0 \tag{61}
$$

which is even more trivial than equation (49).

Translating this into relations between the components of the trifocal tensor gives,

$$
\boxed{|T_a^{i_1j_1} T_a^{i_2j_1} T_a^{i_1j_2}|\ |T_a^{i_2j_2} T_a^{i_1j_2} T_a^{i_2j_3}| - |T_a^{i_1j_1} T_a^{i_2j_1} T_a^{i_2j_2}|\ |T_a^{i_1j_2} T_a^{i_1j_3} T_a^{i_2j_2}| = 0} \tag{62}
$$

The constraints found here were inspired by work done by O.Faugeras and B.Mourrain in [3]. However, the constraints given in [3] form a subset of those given here. Furthermore, here the constraints were derived through mainly geometrical arguments, rather than through the investigation of polynomials as in [3].

The constraint equations (56) and (57) are not given in determinant form[8] in [3]. The constraints given in [3] as equations (12) through (15) are a subset of equation (62) as given here.

[8] These constraints are basically the same as the relations between lines detailed on page 26 of [3].

6 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 form 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 [1]. In the following this algorithm will be called the "7pt algorithm".

To enforce all the determinant constraints, an estimate of the trifocal tensor is first found using the 7pt algorithm. From this tensor the epipoles are estimated. Using these epipoles 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[9]. Using the image point matches in the epipolar frame these 7 components can be found linearly. The trifocal tensor in the "normal" basis is then recovered by tranforming 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.

The main problem with the MinFact algorithm is that it depends crucially on the quality of the initial epipole estimates. If these are bad, the trifocal tensor will still be perfectly self-consistent but will not represent the true camera structure particularly well. This is reflected in the fact that typically a trifocal tensor calculated with the MinFact algorithm does not satisfy the trilinearity relationships as well as a trifocal tensor calculated with the 7pt algorithm, which is of course calculated to satisfy these relationships as well as possible.

Unfortunately, there does not seem to be a way to find the epipoles and the trifocal tensor in the epipolar basis simultaneously with a linear method. In fact, the trifocal tensor in a "normal" basis is a *non-linear* combination of the epipoles and the 7 non-zero components of the trifocal tensor in the epipolar basis.

Nevertheless, since the MinFact algorithm produces a fully self-consistent tensor, the camera matrices extracted from it also have to form a self-consistent set. Reconstruction using such a set of camera matrices may be expected to be better than reconstruction using an inconsistent set of camera matrices, as typically found from an inconsistent trifocal tensor. The fact that the trifocal tensor found with the MinFact algorithm may not resemble the true camera structure very closely, might not matter too much, since reconstruction is only exact up to a projective transformation.

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. This is done as follows:

[9] From this it follows directly that the trifocal tensor has 18 DOF: 12 epipolar components plus 7 non-zero components of the trifocal tensor in the epipolar basis minus 1 for an overall scale.

1. A 3D-object is projected onto the image planes of the three cameras, which subsequently introduce some Gaussian noise into the projected point coordinates. These coordinates are then quantised according to the simulated camera resolution. The magnitude of the applied noise is measured in terms of the mean Gaussian deviation in pixels.

2. The trifocal tensor is calculated in one of two ways from the available point matches:

   (a) using the 7pt algorithm, or

   (b) using the MinFact algorithm.

3. The epipoles and the camera matrices are extracted from the trifocal tensor. The camera matrices are evaluated using Hartleys recomputation method [1].

4. The points are reconstructed using a version of what is called "Method 3" in [10] and [11] adapted for three views. This uses a SVD to solve for the homogeneous reconstructed point algebraically using a set of camera matrices. In [10] and [11] this algorithm was found to perform best of a number of reconstruction algorithms.

5. This reconstruction still contains an unknown projective transformation. Therefore it cannot be compared directly with the original object. However, since only synthetic data is used here, the 3D-points of the original object are known exactly. Therefore, a projective transformation matrix that best transforms the reconstructed points into the true points can be calculated. Then the reconstruction can be compared with the original 3D-object geometrically.

6. 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 3 shows the mean distance between the original points and the reconstructed points in 3D-space in some arbitrary units[10], 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. Figures 4 and 5 both show the mean of the percentage differences between the components of the true and the calculated trifocal tensors as a function of added noise in pixels. Figure 4 compares the trifocal tensors found with the 7pt and the MinFact algorithms. This shows

[10] The particular object used was 2 units wide, 1 unit deep and 1.5 units high in 3D-space. The Y-axis measures in the same units.

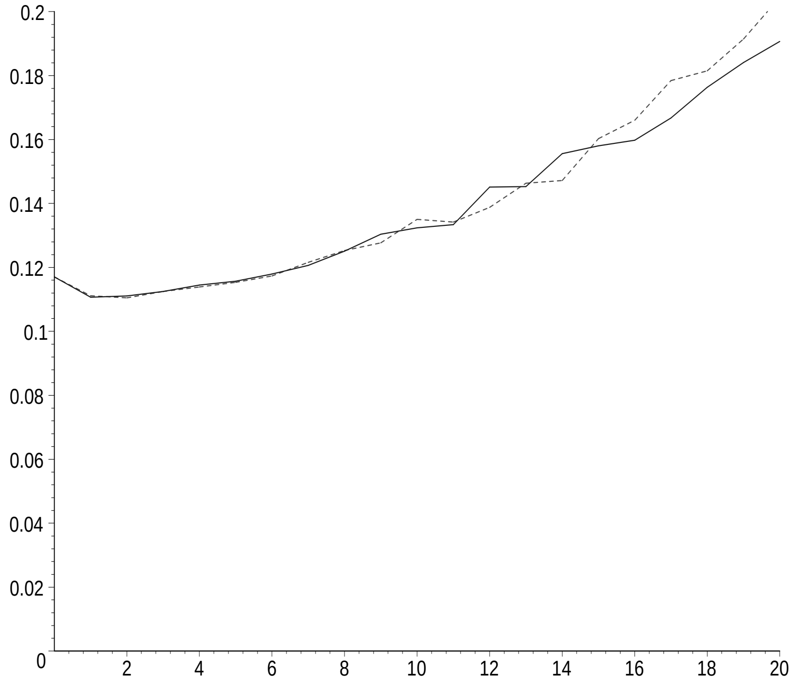

Figure 3: Mean distance between original points and reconstructed points in arbitrary units as a function of mean Gaussian error in pixels introduced by the cameras. The solid line shows the values using the MinFact algorithm, and the dashed line the values for the 7pt algorithm.

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 (shown enlarged in figure 5).

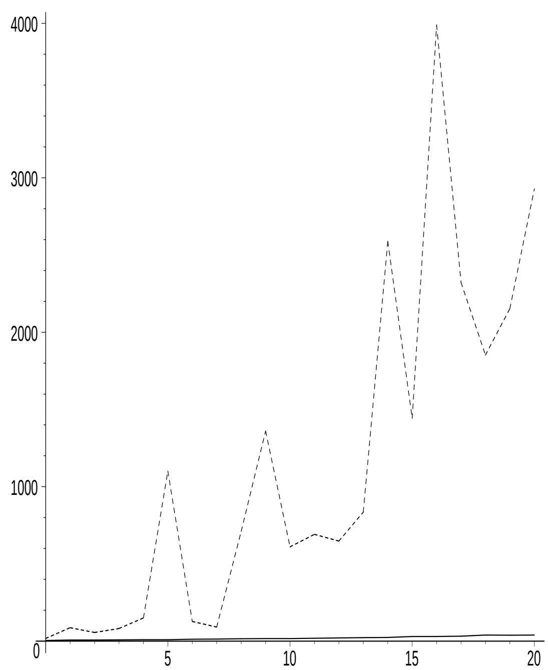

Figure 4: Mean difference between elements of calculated and true tensors in percent. Solid line shows values for trifocal tensor calculated with 7pt algorithm, and dashed line shows values for trifocal tensor calculated with MinFact algorithm.

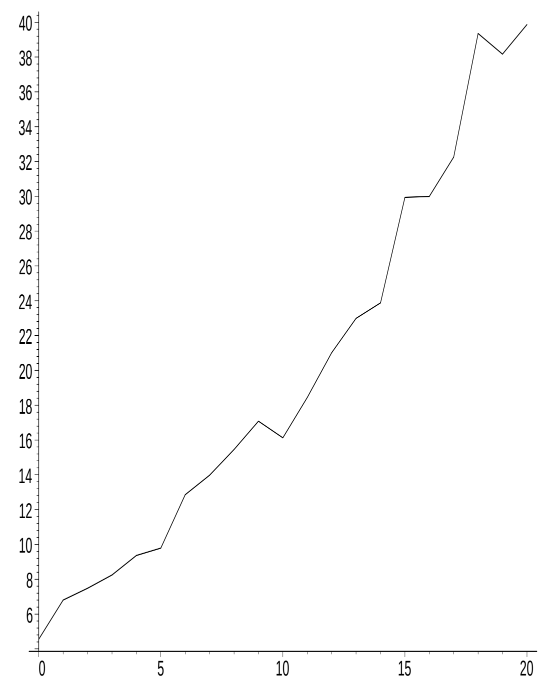

Figure 5: Mean difference between elements of true trifocal tensor and trifocal tensor calculated with 7pt algorithm in percent.

7 Conclusion

It was shown here how the GA approach to the trifocal tensor problem leads to a clear geometrical understanding of the same. 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 showed the advantage of the GA approach over a GC algebra approach, due to GA's *inner product*.

The data presented in section 6 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 is equally as good, in terms of reconstruction ability, as an inconsistent trifocal tensor that satisfies the trilinearity relationships quite well. 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 4) 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[11].

[11] In fact it was found by the authors that an initial reconstruction is almost always flat and located at one of the camera image planes. A projective transformation was then necessary to "unfold" the reconstruction.

References

[1] R. I. Hartley, "Lines and Points in Three Views and the Trifocal Tensor," *International Journal of Computer Vision*, pp. 125–140, 1997.

[2] A. Shashua, "Trilinear Tensor: the Fundamental Construct of Multiple-View Geometry and its Applications," in *Algebraic frames for the Perception-Action Cycle* (G. Sommer and J. Koenderink, eds.), no. 1315 in Lecture Notes in Computer Science, 1997.

[3] O. Faugeras and B. Mourrain, "On the Geometry and Algebra of the Point and Line Correspondences between N Images," Tech. Rep. 2665, INRIA, Sophia Antipolis, 1995.

[4] T. Papadopoulo and O. Faugeras, "A New Characterization of the Trifocal Tensor." On INRIA Sophia Antipolis Web-Site.

[5] O. Faugeras and T. 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*, vol. 356, no. 1740, pp. 1123–1152, 1998.

[6] J. 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] J. Lasenby and A. N. Lasenby, "Estimating Tensors for Matching over Multiple Views," *Phil. Trans. R. Soc. Lond. A*, vol. 356, no. 1740, pp. 1267–1282, 1998.

[8] D. Hestenes and R. Ziegler, "Projective Geometry with Clifford Algebra," *Acta Applicandae Mathematicae*, vol. 23, pp. 25–63, 1991.

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

[10] C. Rothwell, G. Csurka, and O. Faugeras, "A Comparison of Projective Reconstruction Methods for Pairs of Views," Tech. Rep. 2538, INRIA, Sophia Antipolis, 1995.

[11] C. Rothwell, O. Faugeras, and G. Csurka, "A Comparison of Projective Reconstruction Methods for Pairs of Views," *Computer Vision and Image Understanding*, vol. 68-1, pp. 37–58, 1997.