UNIVERSITY OF
CAMBRIDGE

Applications of Geometric Algebra in Computer Vision

The geometry of multiple view tensors and 3D-reconstruction

Christian B.U. Perwass

January 2000

To my parents

Die menschliche Vernunft hat das besondere Schicksal in einer Gattung ihrer Erkenntnisse: daß sie durch Fragen belästigt wird, die sie nicht abweisen kann, denn sie sind ihr durch die Natur der Vernunft selbst aufgegeben, die sie aber auch nicht beantworten kann, denn sie übersteigen alles Vermögen der menschlichen Vernunft.

"Kritik der reinen Vernunft",
Immanuel Kant.

… well, let's scratch the surface …

Abstract

There are four main areas discussed in this thesis: Geometric Algebra, Projective Geometry, Multiple View Tensors and 3D-Reconstruction. Our discussion of Geometric Algebra is similar to that of Hestenes and Sobczyk [30]. Our construction differs mainly in how we define the inner and outer product, and how we work with the geometric product. The geometric algebra we construct here is finite dimensional, non-degenerate and universal.

Our discussion of projective geometry in terms of GA differs somewhat from Hestenes and Ziegler [31], in that we embed Euclidean space in projective space in a different way. Instead of using the *projective split* we employ reciprocal vectors to the same effect. Our approach is independent of the signature of an underlying orthonormal frame. We also use reciprocal frames to give concise descriptions of projections and intersections.

We discuss multiple view tensors from a geometric point of view. We show that in this way multiple view tensors can be expressed in a unified way, and that constraints on them can be found from simple geometric considerations. In the last part of this chapter we discuss *projective* reconstructions from trifocal tensors. We find that the consistency of a trifocal tensor has no particular influence on the quality of reconstruction.

We build upon our analysis of multiple view tensors in the last chapter where we discuss 3D-reconstruction. The algorithm we present uses point matches and vanishing points in order to create an affine 3D-reconstruction of a world scene from two static images taken with unknown cameras, from unknown positions. The algorithm we obtain is fast and robust. Its derivation shows how geometric constraints can be implemented in a straight forward manner using GA.

In this Ph.D. thesis I present most of the research I have done from October 1996 to October 1999 at Cambridge University, together with my supervisors Joan Lasenby[1] and Anthony Lasenby[2]. As the title suggests I concentrate on applications of *Geometric Algebra* (GA) in Computer Vision. However, I also developed a novel axiomatic derivation of GA which is of general relevance.

A number of chapters have been or are to be published together with Joan Lasenby. Therefore, the first person plural, i.e. "we", is used instead of "I" throughout.

[1] C.U. Engineering Department, Trumpington Street, Cambridge CB2 1PZ, U.K. Tel.: +44 (1223) 332639. Email: jl@eng.cam.ac.uk. Web: www-sigproc.eng.cam.ac.uk/˜jl.

[2] AP, Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE, U.K. Email: anthony@mrao.cam.ac.uk.

Contents

Chapter 1

Introduction

For once the disease of reading has laid upon the system it weakens it so that it falls an easy prey to that other scourge which dwells in the inkpot and festers in the quill. The wretch takes to writing[a].

"Orlando", Virginia Woolf.

[a] The scourge seems to have migrated to computers these days, but the effect is the same.

1.1 Structure of the thesis

Chapters 2, 3, 4 and 5 all have an introduction discussing the relevant literature and setting the context in which the respective chapter is to be read. At the end of each of these chapters some conclusions are also given. Therefore, each of these chapters can be read on its own, although results and concepts from previous chapters may be needed. Chapter 6 again summarises the most important results obtained from chapters 2 to 5, and adds some final thoughts. In the following we will give some more details about each of the chapters 2 to 5, separately.

Chapter 2: *A Construction of Geometric Algebra.*

> Here we present a construction of geometric algebra (GA), which is similar to that of Hestenes and Sobczyk in [30]. However, we start with different definitions for the inner and outer product, and we treat the geometric product in a different way. In [30] the properties of the geometric, inner and outer product are initially given with respect to vectors. How these products act on blades then has to be derived by reducing them to products on vectors, which can be quite cumbersome.

We choose a slightly different path. Initially we show that the geometric product is the sum of the commutator and anti-commutator product, which is purely notational. However, this is true for *any* two objects in the GA. We then find general identities for the commutator and anti-commutator products of three arbitrary elements of the GA. Later we show how the inner and outer product can be written in terms of commutator and anti-commutator products. This allows us to find the properties of the inner and outer product with respect to blades more easily than in [30].

Also note that we define what we mean by "vector" in GA in an algebraic way, which encodes the geometric meaning of a vector. This definition excludes degenerate vector spaces. We also assume that the vector space on which we build GA is finite dimensional. These properties lead to a universal, non-degenerate GA.

Chapter 3: *Projective Geometry.*

This chapter deals with the description of projective geometry in terms of GA. The use of GA in this field was introduced by Hestenes and Ziegler in 1991 [31]. We differ somewhat from their description, in that we do not use the *projective split* to go from projective to Euclidean space (see e.g. [37, 49, 48]). The embedding we present here does not assume any particlar signature for an underlying orthonormal basis of projective space.

The first part of this chapter gives an intuitive introduction to projective geometry. Then many formulae and concepts which are essential tools in chapters 4 and 5 are introduced. In particular, we use reciprocal frames to give concise descriptions of camera matrices, epipoles and projections. We also introduce a dual bracket which simplifies the notation of many equations, especially when used in conjunction with the meet operation.

Chapter 4: *The Geometry of Multiple View Tensors.*

In this chapter we apply the tools we have developed in the previous chapters. Multiple View tensors simply relate the images taken by a number of cameras of the same scene. These tensors are usually defined through the image point coordinates of image points in the different images which are projections of the same point in the world. Using matrix and tensor methods the geometry that is encoded in multiple view tensors, and the constraints on the tensors that follow from this, are difficult to find. With GA, on the other hand, we start directly with the underlying geometry of which the respective multiple view tensors are only one of many possible expressions. We are therefore able to express all multiple view tensors in a unified way and find their constraints comparatively easily. We end the chapter by showing that multiple view tensors cannot encode any information about the Euclidean space of the world scene. Therefore, in order to perform a 3D-reconstruction we need more information than just a single multiple view tensor for a particular camera setup. This leads directly to our next chapter where we use some additional information to create an affine 3D-reconstruction. This chapter is to be published in the forthcoming book edited by Prof. G. Sommer "Geometric Computing with Clifford Algebra" [51]. A more detailed discussion of a special set of constraints on the trifocal tensor was published previously [49, 48].

Chapter 5: *3D-Reconstruction.*

Here we present a 3D-reconstruction algorithm which gives an affine reconstruction of a scene from two static images if a set of image point matches and some parallel world lines are known. The algorithm exploits the fact that we can express the collineation of the plane at infinity in terms of a camera matrix. The algorithm we find is fast and robust and is investigated with synthetic and real data. There is also a program available on the enclosed CD which shows some reconstructions and discusses parts of the reconstruction algorithm in more detail (see appendix A). This chapter is also to be published in the book "Geometric Computing with Clifford Algebra" [50].

1.2 Geometric Algebra

Since GA is the essential tool for most of our calculations, we will discuss GA in two parts. The first part is meant to be an intuitive exposé of the most basic features of GA. In particular, we will concentrate on those concepts and formulae which are important to the analysis in later chapters. Therefore, we completely neglect *rotors*, for example. For a more complete introduction see [29, 30] and for other brief summaries see [37, 39, 22]. A good introduction geared towards the use in the computer sciences can be found in [11].

The second part of our introduction to GA is chapter 2, where we put GA on a mathematically more rigorous footing. If the reader is only interested in the Computer Vision aspects of this thesis, chapter 2 can be safely ignored. However, it will give the reader a much deeper understanding of GA.

The easiest way to understand GA is to show how it extends the functionality of standard vector algebra (SVA), which we assume all readers are familiar with. In SVA 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 $\mathcal{E}^3$, $\{e_1, e_2, e_3\}$ 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, SVA 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 $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathcal{E}^n$, then the inner product has the following properties:

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. 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. $\boldsymbol{a} \wedge \boldsymbol{b} = -\boldsymbol{b} \wedge \boldsymbol{a}$. 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 $\mathcal{E}^n$ is of grade n, simply because in $\mathcal{E}^n$ 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 SVA. 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 a linear combination 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*, $\boldsymbol{w} \in \mathcal{E}^3$ 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 $\mathcal{E}^3$ 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 $\mathcal{E}^4$ 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 a linear combination of k-blades, GA also defines a *multivector* which is the linear combination 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 the following chapters, multivectors will not be discussed here. We refer the interested reader to [30].

[1] A k-vector as defined here is also called a homogeneous multivector of grade k. This should not be confused with a “homogeneous vector” in projective geometry which is something quite different.

There is also a distributive law for the inner product with respect to the outer product. This can be described by two general formulas. Let $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\ldots\wedge\boldsymbol{a}_k$ be a k-blade and $B_{\langle l\rangle} = \boldsymbol{b}_1\wedge\ldots\wedge\boldsymbol{b}_l$ be an l-blade. We also define $[A_{\langle k\rangle}\backslash\boldsymbol{a}_i]$ to be the blade $A_{\langle k\rangle}$ with the element $\boldsymbol{a}_i$ taken out. Then it can be shown that for $k \leq l$ we have

$$\begin{aligned} A_{\langle k\rangle}\cdot B_{\langle l\rangle} &= ([A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\wedge\boldsymbol{a}_k)\cdot B_{\langle l\rangle} \\ &= [A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\cdot(\boldsymbol{a}_k\cdot B_{\langle l\rangle}) \end{aligned} \tag{1.1}$$

With this equation we can reduce the inner product between two general blades to a number of inner product operations between a blade and a vector. The inner product between a blade and a vector can then be expanded using the following formula.

$$\boldsymbol{a}\cdot B_{\langle l\rangle} = \sum_{i=1}^{l} (-1)^i\,(\boldsymbol{a}\cdot\boldsymbol{b}_i)\,[B_{\langle l\rangle}\backslash\boldsymbol{b}_i] \tag{1.2}$$

It may not be immediately clear how to use equations (1.1) and (1.2). A few examples, however, should clarify the situation. The following two equations employ equation (1.2).

$$\boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) = (\boldsymbol{a}\cdot\boldsymbol{b}_1)\boldsymbol{b}_2 - (\boldsymbol{a}\cdot\boldsymbol{b}_2)\boldsymbol{b}_1 \tag{1.3a}$$

$$\begin{aligned} \boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2\wedge\boldsymbol{b}_3) &= (\boldsymbol{a}\cdot\boldsymbol{b}_1)(\boldsymbol{b}_2\wedge\boldsymbol{b}_3) - (\boldsymbol{a}\cdot\boldsymbol{b}_2)(\boldsymbol{b}_1\wedge\boldsymbol{b}_3) \\ &\quad +(\boldsymbol{a}\cdot\boldsymbol{b}_3)(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) \end{aligned} \tag{1.3b}$$

For the next expansion we first use equation (1.1) and then equation (1.2).

$$\begin{aligned} (\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) &= \boldsymbol{a}_1\cdot\Big(\boldsymbol{a}_2\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)\Big) \\ &= \boldsymbol{a}_2\cdot\boldsymbol{b}_1\ \boldsymbol{a}_1\cdot\boldsymbol{b}_2\ -\ \boldsymbol{a}_2\cdot\boldsymbol{b}_2\ \boldsymbol{a}_1\cdot\boldsymbol{b}_1 \end{aligned} \tag{1.4}$$

As a slight extension of equation (1.1) it can be shown that for $k \leq l$,

$$\begin{aligned} &A_{\langle k\rangle}\cdot B_{\langle l\rangle} = \\ &\quad\sum_{\{j_i\}} \epsilon_{j_1 j_2\cdots j_l}\,\Big[A_{\langle k\rangle}\cdot(\boldsymbol{b}_{j_1}\wedge\boldsymbol{b}_{j_2}\wedge\ldots\wedge\boldsymbol{b}_{j_k})\Big]\ \boldsymbol{b}_{j_{k+1}}\wedge\ldots\wedge\boldsymbol{b}_{j_l}, \end{aligned} \tag{1.5}$$

where $\epsilon_{j_1 j_2\cdots j_l}$ is $+1$ if the $\{j_i\}$ form an even permutation of $\{1, 2, \ldots, l\}$, -1 if they form an odd permutation and 0 if any two indices are equal. For example,

$$\begin{aligned} (\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2\wedge\boldsymbol{b}_3) &= \Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)\Big]\boldsymbol{b}_3 \\ &\quad -\Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_3)\Big]\boldsymbol{b}_2 \\ &\quad +\Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_2\wedge\boldsymbol{b}_3)\Big]\boldsymbol{b}_1 \end{aligned} \tag{1.6}$$

Equations (1.3), (1.4) and (1.6) clearly show the *non-metric* side of the inner product. For example, in equation (1.3a) the inner product of a vector with a bivector results in a vector. In equation (1.3b) the inner product of a vector with a *trivector*[2] gives a 2-vector. Similarly for equations (1.4).

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. We also define that the inner product of a scalar with a k-vector is identically zero.

In chapter 3 it will be shown that intersections as well as the dual operation can be expressed in terms of the inner product. The Grassmann-Cayley (GC) algebra lacks such a universal operator and instead defines 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 vector $\boldsymbol{x} \in \mathcal{E}^3$ 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

$$\begin{aligned} \boldsymbol{a} \cdot \boldsymbol{x} &= \boldsymbol{a} \cdot \Big[\boldsymbol{a} \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2)\Big] \\ &= (\boldsymbol{a} \wedge \boldsymbol{a}) \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) \quad \text{from equation (1.4)} \\ &= 0 \end{aligned} \tag{1.7}$$

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

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{a} \cdot (\boldsymbol{b}_1 \wedge \boldsymbol{b}_2) \\ &= (\boldsymbol{a} \cdot \boldsymbol{b}_1) \boldsymbol{b}_2 - (\boldsymbol{a} \cdot \boldsymbol{b}_2) \boldsymbol{b}_1 \end{aligned} \tag{1.8}$$

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 spaces 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 $\boldsymbol{a}$.

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

By definition the inner product of two vectors 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, as will be shown in chapter 2. The geometric product of two vectors is written as $\boldsymbol{ab}$ and given by

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

Chapter 2

A Construction of Geometric Algebra

Horatio: *O day and night, but this is wonderous strange!*
Hamlet: *And therefore as a stranger give it welcome.*
There are more things in heaven and earth, Horatio,
Than are dreamt of in your philosophy.

"Hamlet", William Shakespeare.

William K. Clifford (1845–1879) did a lot of work on applications of Hermann G. Grassmann's (1809-1877) extensive algebra and William R. Hamilton's (1805–1865) quaternions [6]. Clifford realized that these two "geometric algebras" share some fundamental geometric concepts. Consequently he developed a more general geometric algebra in which extensive (or exterior) algebra and quaternions are contained [7].

In [5] Clifford develops a nice geometric link between Grassmann's extensive algebra and Hamilton's quaternions, which we will present here. His critique of quaternions is that they play two roles at the same time: they are operator but also subject of operation. This differs from the view "Ausdehnungslehre" (extensive algebra) is founded on, where both objects of a product play similar parts. In Grassmann's extensive algebra the product of two vectors a, b, written ab, represents the line through them. This product has the properties $a^2 = 0$ and $ab = -ba$.

Consider a system of 4 elements $\{e_0, e_1, e_2, e_3\}$ on which a polar product is defined, i.e. $e_i e_j = -e_j e_i$, for $i \neq j$. Let the $\{e_i\}$ represent points in 3D-space, whereby e_0 is the origin and e_1, e_2, e_3 are three points an infinite distance from e_0 in three mutually orthogonal directions. The three objects e_0e_1, e_0e_2, e_0e_3 therefore represent three mutually orthogonal lines.

Now define three operators $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ where $\boldsymbol{i}$ turns the line e_0e_2 into e_0e_3, $\boldsymbol{j}$ turns e_0e_3 into e_0e_1 and $\boldsymbol{k}$ turns e_0e_1 into e_0e_2. Since the product operation considered here is associative, we can write

$$\begin{aligned} (e_2e_3)(e_0e_2) &= e_2e_3e_0e_2 \\ &= -e_2^2\, e_0e_3. \end{aligned} \tag{2.1}$$

Hence, if we take $e_2^2 = -1$ (and not 0 as in extensive algebra) we can write $\boldsymbol{i} = e_2e_3$. The geometric interpretation of this is that the turning of e_0e_2 into e_0e_3 is equivalent to a translation along the line at infinity e_2e_3. Similarly, if we take $e_1^2 = e_3^2 = -1$ we have $\boldsymbol{j} = e_3e_1$ and $\boldsymbol{k} = e_1e_2$.

$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are just the quaternions, whose properties can now be derived from the more fundamental product between the elements $\{e_i\}$. We will call this product the *geometric product.* Furthermore, the operators $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are now distinct from the objects they operate on.

It seems that we have lost Grassmann's extensive algebra now, though. Quite the opposite: extensive algebra is now embedded in this more general geometric algebra. If we define a scalar product as

$$e_i \cdot e_j = \begin{cases} 0\,, & \text{if } i \neq j \\ \pm 1\,, & \text{if } i = j \end{cases} \tag{2.2}$$

and an exterior product as

$$e_i \wedge e_j = \begin{cases} e_ie_j\,, & \text{if } i \neq j \\ 0\,, & \text{if } i = j \end{cases} \tag{2.3}$$

the geometric product can be written as

$$e_ie_j = e_i \cdot e_j + e_i \wedge e_j. \tag{2.4}$$

If we only use the exterior product we return to Grassmann's extensive algebra. If we only use the scalar product we obtain J. Willard Gibbs's (1839–1903) vector algebra (appart from the cross product). In fact, the cross product is a first hint at the need of an exterior product, since the cross product of two vectors (in 3D) is nothing else but the dual of their exterior product (see e.g. [41, 30]).

We will call the algebra introduced by Clifford "geometric algebra" (GA), if it is not degenerate and vectors in the algebra are n-tuples in $\mathcal{R}^n$, i.e. Gibbs vectors. Today the ideas behind GA have been generalized and put into a more rigorous mathematical form. A general "Clifford algebra" is defined[1] as follows.

Let (V, Q) denote an arbitrary quadratic space, where V is a finite-dimensional vector space over some field $\mathcal{F}$. Let $\boldsymbol{A}$ be an associative algebra over $\mathcal{F}$ with identity 1 and $\nu : V \rightarrow \boldsymbol{A}$ an $\mathcal{F}$–linear embedding of V into $\boldsymbol{A}$. Then the pair $(\boldsymbol{A}, \nu)$ is said to be a *Clifford algebra* for (V, Q) when

1. $\boldsymbol{A}$ is generated as an algebra by $\{\nu(v) : v \in V\}$ and $\{\lambda 1 : \lambda \in \mathcal{F}\}$,
2. $(\nu(v))^2 = -Q(v)1$, for all $v \in V$.

What we called geometric algebra above is therefore a Clifford algebra on a non-degenerate quadratic space (V, Q), where V is a "Gibbs vector space" and Q is the geometric product. Grassmann's extensive algebra can be reproduced by setting $Q \equiv 0$. That is, the square of any vector is zero. General discussions of Clifford algebras can be found in [21, 23, 52]. Applications of Clifford algebras to Physics and Engineering are usually given in the framework of geometric algebra [41, 54, 30, 38, 11].

A different approach to geometric or Clifford algebra is discussed by Sommen in [64, 65]. He does not start off with a general geometric algebra in which vectors are defined. Instead he defines what he calls a "radial algebra" in which vector–*variables* are defined as general objects. There is no particular meaning given to these vector–variables (hence "variables") appart from their algebraic behaviour. In the same way bivector–variables, etc. are defined.

For example, vector–variables are defined through the property that their anti-commutator is a scalar (in some field). Bivector–variables are defined by saying that the commutator of a bivector–variable with a vector–variable gives a "new" vector variable. In this way an infinite dimensional vector space can be created from a single vector– and bivector–variable. It can then be shown that there is an algebra isomorphism between this radial algebra and a Clifford algebra.

2.1 Introduction

The construction of GA we give here is probably most similar to that of Hestenes and Sobczyk [30]. Our construction differs mainly in how we define the inner and outer product, and how we work with the geometric product. Our strategy is as follows. We start with the geometric product, which we

[1] We follow here the notation of [21].

assume is associative. We then define the commutator and anti-commutator product, and show that the geometric product can be written as their sum. This is, of course, purely notational. However, this may be a better description of the true nature of the geometric product than to define it only through its properties on vectors.

Next we define what we mean by vectors in an algebraic sense. This also defines the bilinear (and quadratic) form on vectors to be the anti-commutator product. We can then choose an orthonormal basis of such a vector space and show that the generated GA is universal. Universality means that the GA stays unchanged under a change of basis. Hence, we can define inner and outer product in general on some orthonormal basis. This allows us to show the relation between inner and outer product, and the commutator and anti-commutator product, which will be very helpful when we discuss the properties of blades.

2.2 The Foundations

2.2.1 The Fundamental Axioms

We will denote a geometric algebra by $\mathcal{G}$. $\mathcal{G}$ is a finite-dimensional vector space over the scalar field $\mathcal{R}$, where $\mathcal{R}$ denotes the reals. Elements of $\mathcal{G}$ are called multivectors and elements of $\mathcal{R}$ scalars. Furthermore, a product is defined on $\mathcal{G}$, the "geometric product", which is associative and distributive, also with respect to scalars. In particular note that the geometric product of a scalar with a multivector commutes. The identity of $\mathcal{G}$ with respect to the geometric product is $1 \in \mathcal{R}$. We also assume that $\mathcal{G}$ is closed under the geometric product. $\mathcal{G}$ is therefore a vector space and an associative algebra with identity.

The geometric product is written by juxtaposition of two multivectors. For example, the geometric product of two multivectors A and B is written AB. The afore mentioned axioms of $\mathcal{G}$ are summarized in the following.

Axiom 2.2.1 Vector Space Axioms. *Let $A, B \in \mathcal{G}$ and $\alpha, \beta \in \mathcal{R}$, then*

$$\begin{array}{ll}
A + B \in \mathcal{G} & \textit{GA is closed under addition} \\
A + B = B + A & \textit{Commutativity} \\
A + (B + C) = (A + B) + C & \textit{Associativity} \\
A + 0 = A & \textit{Unique additive identity } 0 \\
A + (-A) = 0 & \textit{Unique additive inverse element} \\
1A = A1 = A & \textit{Unique scalar identity } 1 \\
\alpha A = A\alpha & \textit{Commutativity with scalars} \\
\alpha(\beta A) = (\alpha\beta)A & \textit{Associativity with scalars} \\
\alpha(A + B) = (\alpha A) + (\beta B) & \\
(\alpha + \beta)A = (\alpha A) + (\beta A) & \textit{Distributivity with scalars}
\end{array} \tag{2.5}$$

Axiom 2.2.2 Geometric Product Axioms. *Let $A, B, C \in \mathcal{G}$ and $\alpha, \beta \in \mathcal{R}$, then*

$$\begin{array}{ll}
(AB) \in \mathcal{G} & \mathcal{G} \textit{ is closed under geometric product} \\
(AB)C = A(BC) & \textit{Associativity} \\
A(B + C) = (AB) + (AC) & \\
(B + C)A = (BA) + (CA) & \textit{Distributivity} \\
(\alpha A)B = A(\alpha B) = \alpha(AB) & \textit{Associativity with scalars}
\end{array} \tag{2.6}$$

Conventionally the geometric product takes precedence over addition. That is

$$AB + C = (AB) + C \tag{2.7}$$

Note that the geometric product is *not* commutative. That is, in general $AB \neq BA$.

2.2.2 The Commutator and Anti-Commutator Products

Definition 2.2.1 *Let $A, B \in \mathcal{G}$, then the* ***commutator product*** *of A and B is defined as*

$$A \underline{\times} B = \tfrac{1}{2}(AB - BA) \tag{2.8}$$

From this definition it follows directly that

$$A \underline{\times} B = -B \underline{\times} A \tag{2.9}$$

Definition 2.2.2 *Let $A, B \in \mathcal{G}$, then the **anti-commutator product** of A and B is defined as*

$$A \overline{\times} B = \tfrac{1}{2}(AB + BA) \tag{2.10}$$

Therefore, we have

$$A \overline{\times} B = B \overline{\times} A \tag{2.11}$$

The geometric product can thus be expressed in terms of the commutator product and the anti-commutator product. Let $A, B \in \mathcal{G}$, then

$$\begin{aligned} AB &= A \underline{\times} B + A \overline{\times} B \\ &= \tfrac{1}{2}(AB - BA) + \tfrac{1}{2}(AB + BA) \\ &= \tfrac{1}{2}AB + \tfrac{1}{2}AB + \tfrac{1}{2}BA - \tfrac{1}{2}BA \\ &= AB \end{aligned} \tag{2.12}$$

The geometric product combines the commuting and anti-commuting parts of two multivectors. We will see that the commutator and anti-commutator products are very useful when working with vectors.

We will now derive some identities for objects of the type $(A \underline{\times} B) \underline{\times} C$. These will be very useful later on. For example,

$$\begin{aligned} (A \underline{\times} B) \underline{\times} C &= \tfrac{1}{2}(AB - BA) \underline{\times} C \\ &= \tfrac{1}{4}\Big[(AB - BA)C - C(AB - BA)\Big] \\ &= \tfrac{1}{4}\Big[ABC - BAC - CAB + CBA\Big] \\ &= \tfrac{1}{4}\Big[\ (ABC - ACB) + (CBA - BCA) \\ &\qquad +(ACB - CAB) + (BCA - BAC)\Big] \\ &\qquad \text{(add and subtract } ACB \text{ and } BCA\text{)} \\ &= \quad \tfrac{1}{4}\Big[A(BC - CB) - (BC - CB)A\Big] \\ &\quad + \tfrac{1}{4}\Big[(AC - CA)B - B(AC - CA)\Big] \\ &= A \underline{\times} (B \underline{\times} C) + (A \underline{\times} C) \underline{\times} B \end{aligned} \tag{2.13}$$

In step 4 of the above calculation the terms could have also been arranged as follows,

$$\begin{aligned}
(A\underline{\times}B)\underline{\times}C &= \tfrac{1}{4}\Big[\ (ABC+ACB)+(CBA+BCA) \\
&\qquad +(-ACB-CAB)+(-BCA-BAC)\Big] \\
&= \quad \tfrac{1}{4}\Big[A(BC+CB)+(BC+CB)A\Big] \\
&\quad - \ \tfrac{1}{4}\Big[(AC+CA)B+B(AC+CA)\Big] \\
&= A\overline{\times}(B\overline{\times}C)-(A\overline{\times}C)\overline{\times}B
\end{aligned} \tag{2.14}$$

That is, there are two possible expansions for $(A\underline{\times}B)\underline{\times}C$. Being able to choose one or the other will turn out to be quite useful. In the following we will list the expansions for all possible combinations of three multivectors.

$$(A\underline{\times}B)\underline{\times}C = A\underline{\times}(B\underline{\times}C)+(A\underline{\times}C)\underline{\times}B \tag{2.15a}$$

$$(A\underline{\times}B)\underline{\times}C = A\overline{\times}(B\overline{\times}C)-(A\overline{\times}C)\overline{\times}B \tag{2.15b}$$

$$(A\underline{\times}B)\overline{\times}C = A\overline{\times}(B\underline{\times}C)+(A\overline{\times}C)\underline{\times}B \tag{2.16a}$$

$$(A\underline{\times}B)\overline{\times}C = A\underline{\times}(B\overline{\times}C)-(A\underline{\times}C)\overline{\times}B \tag{2.16b}$$

$$(A\overline{\times}B)\underline{\times}C = A\underline{\times}(B\overline{\times}C)-(A\overline{\times}C)\underline{\times}B \tag{2.17a}$$

$$(A\overline{\times}B)\underline{\times}C = A\overline{\times}(B\underline{\times}C)+(A\underline{\times}C)\overline{\times}B \tag{2.17b}$$

$$(A\overline{\times}B)\overline{\times}C = A\overline{\times}(B\overline{\times}C)-(A\underline{\times}C)\underline{\times}B \tag{2.18a}$$

$$(A\overline{\times}B)\overline{\times}C = A\underline{\times}(B\underline{\times}C)+(A\overline{\times}C)\overline{\times}B \tag{2.18b}$$

Note that equation (2.15a) is just the Jacobi identity,

$$A\underline{\times}(B\underline{\times}C)+B\underline{\times}(C\underline{\times}A)+C\underline{\times}(A\underline{\times}B)=0$$

2.2.3 Vectors

Now that we have given the fundamental axioms and some definitions, we can structure $\mathcal{G}$. We will do this by defining what we mean by a vector in an algebraic sense.

Definition 2.2.3 *Define a subspace $\mathcal{V}^n \subset \mathcal{G}$ of finite dimension n. Elements of $\mathcal{V}^n$ will be called vectors and they obey the following conditions. Let $x, y \in \mathcal{V}^n$ be* linearly independent, *then*

$$x \mathbin{\overline{\times}} y \in \mathcal{R} \tag{2.19a}$$

$$x \mathbin{\underline{\times}} y \notin \mathcal{R} \tag{2.19b}$$

$$x \mathbin{\underline{\times}} y \notin \mathcal{V}^n \tag{2.19c}$$

Furthermore, let $x, x' \in \mathcal{V}^n$ be linearly dependent, *then*

$$x \mathbin{\overline{\times}} x' \in \mathcal{R} \quad \textit{and} \quad x \mathbin{\overline{\times}} x' \neq 0 \tag{2.20}$$

Properties (2.19a) and (2.20) together define a bilinear form on $\mathcal{V}^n$. The properties $x \mathbin{\overline{\times}} x' \neq 0$ (from (2.20)) and (2.19b) ensure that $\mathcal{V}^n$ is not degenerate. Finally, (2.19c) gives elements of $\mathcal{V}^n$ the right properties for our purposes. Note that we also have $x \mathbin{\underline{\times}} x' = 0$. This follows from the definition of linear dependence and the axioms of $\mathcal{G}$.

There are at most n linearly independent vectors in $\mathcal{V}^n$. A set of n linearly independent vectors $\{x_1, x_2, \ldots, x_n\} \in \mathcal{V}^n$ generates $\mathcal{V}^n$ and is called a basis of $\mathcal{V}^n$. That a basis for $\mathcal{V}^n$ exists is a standard proof, see e.g. [35, 9].

Two vectors $x, y \in \mathcal{V}^n$ are called *orthogonal* if $x \mathbin{\overline{\times}} y = 0$. They are called parallel if $x \mathbin{\underline{\times}} y = 0$. We will call a basis of $\mathcal{V}^n$ orthogonal if all vectors in the basis are mutually orthogonal. If furthermore the elements of an orthogonal basis square to ±1, the basis will be called *orthonormal.* We can construct an orthonormal basis out of an arbitrary basis of $\mathcal{V}^n$ by using the Gram-Schmidt method. Since this is always possible we can always find an orthonormal basis for some finite dimensional $\mathcal{V}^n$.

Let $B_E := \{e_1, e_2, \ldots, e_n\}$ be an orthonormal basis of $\mathcal{V}^n$, then

$$e_i \mathbin{\overline{\times}} e_j = \lambda_i \, \delta_{ij} \tag{2.21}$$

where δ_{ij} is the Kronecker delta function and $\lambda_i = \pm 1$. Note that $\lambda_i = 0$ is not possible by definition 2.2.3.

Definition 2.2.4 *With regard to equation (2.21), the set $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is called the* **signature** *of $\mathcal{V}^n$.*

If we want to emphasize the signature of a vector space we write $\mathcal{V}^{r,s}$ to denote a vector space of dimension $r + s$ whose signature has r entries $+1$ and s entries -1.

From equation (2.21) and equation (2.12) it follows directly that if $i \neq j$

$$\begin{aligned} e_i e_j &= \; e_i \mathbin{\underline{\times}} e_j \; + \; e_i \mathbin{\overline{\times}} e_j \; = \; e_i \mathbin{\underline{\times}} e_j \,, \\ e_j e_i &= \; e_j \mathbin{\underline{\times}} e_i \; + \; e_j \mathbin{\overline{\times}} e_i \; = \; e_j \mathbin{\underline{\times}} e_i \,. \end{aligned} \tag{2.22}$$

Since $e_i \mathbin{\underline{\times}} e_j \; = - \, e_j \mathbin{\underline{\times}} e_i$, it follows that

$$e_i e_j = -e_j e_i \tag{2.23}$$

2.2.4 Basis Blades

Definition 2.2.5 *Let $B_E = \{e_1, \ldots, e_n\}$ be an orthonormal basis of $\mathcal{V}^n$, and let $N_{\mathcal{V}^n} = \{1, 2, \ldots, n\}$. For each non-empty subset μ of $N_{\mathcal{V}^n}$ a **basis blade** is defined as*

$$e_\mu \equiv e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k} \,, \qquad \mu = \{\mu_1, \mu_2, \ldots, \mu_k\} \tag{2.24}$$

*If the set μ is also ordered as $\mu_1 < \mu_2 < \ldots < \mu_k < n$ we call e_μ a **reduced basis blade**. By convention, $e_\emptyset$ is the identity* 1 *in $\mathcal{G}$, where $\emptyset$ is the empty subset of $N_{\mathcal{V}^n}$. $e_\emptyset$ is also called the* empty blade. *We write $|\mu|$ to denote the number of elements in μ. The **grade** of e_μ is defined as $|\mu|$.*

Let $P_{\mathcal{V}^n}$ be the set of all reduced subsets of $N_{\mathcal{V}^n}$. Note that $|P_{\mathcal{V}^n}| = 2^n$. Then the set of reduced basis blades corresponding to $P_{\mathcal{V}^n}$ forms a basis of a 2^n dimensional GA. Such a GA will be denoted by $\mathcal{G}^n$. That the dimension of $\mathcal{G}^n$ is indeed 2^n follows from the definition of $\mathcal{V}^n$, 2.2.3 (in particular equation (2.19c)).

The importance of equation (2.19c) becomes clear in the following example. Let $\{e_0, e_1, e_2, e_3\}$ be the *associated Pauli matrices* defined as

$$e_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \;\; e_1 = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \;\; e_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \;\; e_3 = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}.$$

Clearly e_0 is isomorph to the scalar identity 1, and $\{e_1, e_2, e_3\}$ span a three dimensional vector space. It is easy to verify that

$$e_0^2 = e_0 \approx 1 \,, \;\; e_1^2 = e_2^2 = e_3^2 \approx -1 \,, \;\; e_i e_j = -e_j e_i \,, i \neq j \tag{2.25}$$

where $i, j \in \{1, 2, 3\}$. However, we also have

$$e_{i_1} e_{i_2} = e_{i_3} \,, \quad e_1 e_2 e_3 \approx -1, \tag{2.26}$$

where $\{i_1, i_2, i_3\}$ form an even permutation of $\{1, 2, 3\}$. Therefore, the $\{e_i\}$ only satisfy properties (2.19a), (2.19b) and (2.20), but not property (2.19c). From equation (2.26) it also follows that the reduced basis blades that can be formed with $\{e_1, e_2, e_3\}$ only span a $2^{(3-1)}$ dimensional GA. Thus, this GA is not universal [21].

The advantage of using a universal GA is that it stays unchanged under a change of basis. This is clearly very important for the applications we want to discuss in later chapters. It can be shown (e.g. [21]) that a Clifford algebra is universal if and only if it is of maximum dimension. Thus, since $\mathcal{G}^n$, as defined above, is of maximum dimension (2^n), it is universal.

Definition 2.2.6 *The reduced basis blade of highest grade in some $\mathcal{G}^n$ is called a* pseudoscalar *and is denoted by $I_{\mathcal{V}^n} = e_1 e_2 \ldots e_n$. If it is clear with which vector space we work, we will also simply write I.*

A general multivector $A \in \mathcal{G}^n$ can be expressed as a linear combination of a set of reduced basis blades. That is,

$$A = \sum_i \alpha^i e_{\mu^i} \,, \quad \mu^i \in P_{\mathcal{V}^n} \,, \tag{2.27}$$

where $\alpha^i \in \mathcal{R}$, and μ^i is the i^{th} element in $P_{\mathcal{V}^n}$. Recall that $P_{\mathcal{V}^n}$ is the set of all reduced subsets of $N_{\mathcal{V}^n}$. That is, $e_{\mu^i} = e_{\mu^i_1} e_{\mu^i_2} \ldots e_{\mu^i_k}$. Hence, equation (2.27) is a sum over a basis of $\mathcal{G}^n$.

It will be necessary later on to move the position of a reduced basis blade within a geometric product. For example, we need to know how $e_\mu e_\nu$ is related to $e_\nu e_\mu$. We know that if $i \neq j$, $e_i e_j = -e_j e_i$. Using the associativity of the geometric product we can quite easily find the following equations. Let $\mu \in P_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$, then

$$e_\mu e_r = \begin{cases} (-1)^{|\mu|} \; e_r e_\mu & \text{if } r \notin \mu \\ (-1)^{(|\mu|-1)} \; e_r e_\mu & \text{if } r \in \mu \end{cases} \tag{2.28}$$

In general, if $\mu, \nu \in P_{\mathcal{V}^n}$ have r elements in common we have

$$\begin{aligned} e_\mu e_\nu &= (-1)^{|\mu|(|\nu|-r)} \, (-1)^{(|\mu|-1)r} \; e_\nu e_\mu \\ &= (-1)^{(|\mu||\nu|-r)} \; e_\nu e_\mu \end{aligned} \tag{2.29}$$

The following definitions will introduce some more nomenclature regarding basis blades.

Definition 2.2.7 *Define* $P^{+}_{\mathcal{V}^n} := \{\mu \in P_{\mathcal{V}^n} : |\mu|$ *is even*$\}$, *the set of all reduced subsets of* $N_{\mathcal{V}^n}$ *with an even number of elements. Similarly,* $P^{-}_{\mathcal{V}^n} := \{\mu \in P_{\mathcal{V}^n} : |\mu|$ *is odd*$\}$. *For a reduced basis blade of even or odd grade we also write* e_{μ^+} *or* e_{μ^-}, *respectively.*

Definition 2.2.8 *An* ***even multivector*** *is a linear combination of even reduced basis blades. An* ***odd multivector*** *is a linear combination of odd reduced basis blades.*

Definition 2.2.9 *A* ***homogeneous multivector*** *is a linear combination of reduced basis blades of equal grade. If we define* $P^{r}_{\mathcal{V}^n} := \{\mu \in P_{\mathcal{V}^n} : |\mu| = r\}$, *the set of all reduced subsets of* $N_{\mathcal{V}^n}$ *with* r *elements, then a homogeneous multivector* A *of grade* r *can be written as*

$$A = \sum_{i}^{|P^r_{\mathcal{V}^n}|} \alpha^i e_{\mu^i}\,, \quad \mu^i \in P^r_{\mathcal{V}^n}.$$

Note that a scalar behaves like a reduced basis blade of grade zero because $1 \approx e_\emptyset$ and the grade of $e_\emptyset$ is $|\emptyset| = 0$. This allows us to formulate the following lemma.

Lemma 2.2.1 *Let* $\mu, \nu, \sigma \in P_{\mathcal{V}^n}$ *and* $e_\mu, e_\nu \in \mathcal{G}^n$. *Also let* $\mu = \mu' \cup \sigma$ *and* $\nu = \nu' \cup \sigma$, *where* $\mu' \cap \nu' = \emptyset$. *That is,* e_μ *and* e_ν *have* e_σ *in common. Then* $e_\mu e_\nu = \alpha e_\tau$, *where* $\tau = \mu' \cup \nu'$ *and* α *is a scalar factor. Note that* $|\tau| = |\mu| + |\nu| - 2|\sigma|$.

Proof.

$$\begin{aligned} e_\mu e_\nu &= (-1)^x\,(e_{\mu'} e_\sigma)\,(-1)^y\,(e_{\nu'} e_\sigma) \\ &= (-1)^{(xy\,|\mu'||\nu'|)} e_{\mu'} e_{\nu'} (e_\sigma e_\sigma) \end{aligned} \tag{2.30}$$

where $(-1)^{(|\mu'||\nu'|)}$ is the sign introduced by interchanging e_σ and $e_{\nu'}$ and $(-1)^x$ and $(-1)^y$ are the signs introduced by untangling e_μ and e_ν, respectively. From the associativity of the geometric product and with the help of equation (2.28) it can be seen quite easily that $(e_\sigma)^2$ is a scalar. Therefore, we get the proposed result. ■

Theorem 2.2.1 *The set of all even multivectors of some $\mathcal{G}^n$ forms a subalgebra of $\mathcal{G}^n$. That is, it is a vector space and is closed under the geometric product. This is not the case for the set of all odd multivectors.*

Proof. We have to show that the geometric product and the sum of any two even multivectors is also an even multivector. This is obviously the case for the sum, since it does not change the grade of a reduced basis blade. From lemma 2.2.1 it follows that the geometric product of any two even reduced basis blades will give an even reduced basis blade, or the empty blade. Hence, the set of even multivectors of some $\mathcal{G}^n$ forms a subalgebra of $\mathcal{G}^n$.

Lemma 2.2.1 also shows that the geometric product of two odd reduced basis blades gives an even reduced basis blade. Therefore, the set of odd multivectors of some $\mathcal{G}^n$ is not a subalgebra of $\mathcal{G}^n$. ■

2.3 Inner and Outer Products

2.3.1 The Outer Product

Definition 2.3.1 *Let $\mu, \nu \in P_{\mathcal{V}^n}$, then the **outer product** of reduced basis blades $e_\mu, e_\nu \in \mathcal{G}^n$ is written $e_\mu \wedge e_\nu$ and defined as*

$$e_\mu \wedge e_\nu \equiv \begin{cases} e_\mu e_\nu & \text{if } \mu \cap \nu = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

That is, the outer product reduces to the geometric product if the two reduced basis blades have no e-components in common, otherwise it results in zero. The outer product is extended to general multivectors through the following two definitions.

Definition 2.3.2 *Let $\mu, \nu \in P_{\mathcal{V}^n}$ and accordingly $e_\mu, e_\nu \in \mathcal{G}^n$. Also let $\alpha \in \mathcal{R}$. Then*

$$(\alpha e_\mu) \wedge e_\nu \equiv e_\mu \wedge (\alpha e_\nu) \equiv \alpha (e_\mu \wedge e_\nu)$$

The outer product of a scalar α with a reduced basis blade e_μ is therefore given by

$$\alpha \wedge e_\mu = (\alpha e_\emptyset) \wedge e_\mu = \alpha (e_\emptyset \wedge e_\mu) = \alpha e_\mu \tag{2.31}$$

Definition 2.3.3 *Let $A, B, C \in \mathcal{G}^n$, then*

$$\begin{aligned} A\wedge(B+C) &\equiv (A\wedge B) + (A\wedge C) \\ (B+C)\wedge A &\equiv (B\wedge A) + (C\wedge A) \end{aligned}$$

Corollary 2.3.1 *The outer product is associative. That is, if $A, B, C \in \mathcal{G}^n$ are general multivectors, then $(A\wedge B)\wedge C = A\wedge(B\wedge C)$.*

Proof. We first show the associativity of the outer product for reduced basis blades. Let $\mu, \nu, \tau \in P_{\mathcal{V}^n}$ and $e_\mu, e_\nu, e_\tau \in \mathcal{G}^n$. Also $\mu \cap \nu = \emptyset$, $e_\mu \cap \tau = \emptyset$ and $\nu \cap \tau = \emptyset$. Then,

$$\begin{aligned} (e_\mu\wedge e_\nu)\wedge e_\tau &= (e_\mu e_\nu)\wedge e_\tau \\ &= (e_\mu e_\nu)\, e_\tau = e_\mu\,(e_\nu e_\tau) \\ &= e_\mu\wedge(e_\nu\wedge e_\tau) \end{aligned}$$

Because of the distributivity of the outer product, associativity also holds for general multivectors. ■

Therefore, we may write a reduced basis blade as the outer product of its constituent elements. That is,

$$e_\mu = e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k} = e_{\mu_1}\wedge e_{\mu_2}\wedge\ldots\wedge e_{\mu_k}$$

Thus the outer product may also be regarded as a **step-up operator** which either increases the grade of a reduced basis blade or gives zero. That is,

$$e_\mu\wedge e_r = e_{\mu'} \quad \text{iff } r \notin \mu \tag{2.32}$$

where $|\mu'| = |\mu| + 1$.

The outer product between a reduced basis blade and some element $e_r \in \mathcal{G}^n$ can also be expressed in terms of the commutator and anti-commutator products. This will become very useful later on when we discuss vectors.

Theorem 2.3.1 *Let $\mu \in P_{\mathcal{V}^n}$, $r \in N_{\mathcal{V}^n}$ and hence $e_\mu, e_r \in \mathcal{G}^n$. Then the outer product between e_μ and e_r can be written as*

$$e_\mu\wedge e_r = \begin{cases} e_\mu \mathbin{\overline{\times}} e_r & \textit{if } |\mu| \textit{ is even or zero} \\ e_\mu \mathbin{\underline{\times}} e_r & \textit{if } |\mu| \textit{ is odd} \end{cases}$$

Proof. We will split the proof into two parts: for even and odd reduced basis blades. Let $\mu^+ \in P^+_{\mathcal{V}^n}$, $\mu^- \in P^-_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$ with corresponding reduced basis blades $e_{\mu^+}, e_{\mu^-}, e_r \in \mathcal{G}^n$.

$$\begin{aligned} e_{\mu^+} \overline{\times} e_r &= \tfrac{1}{2}(e_{\mu^+}e_r + e_r e_{\mu^+}) \\ &= \begin{cases} \tfrac{1}{2}(e_{\mu^+}e_r - e_{\mu^+}e_r) & \text{if } r \in \mu^+ \\ \tfrac{1}{2}(e_{\mu^+}e_r + e_{\mu^+}e_r) & \text{if } r \notin \mu^+ \end{cases} \\ &= \begin{cases} 0 & \text{if } r \in \mu^+ \\ e_{\mu^+}e_r & \text{if } r \notin \mu^+ \end{cases} \end{aligned} \tag{2.33}$$

Furthermore,

$$\begin{aligned} e_{\mu^-} \underline{\times} e_r &= \tfrac{1}{2}(e_{\mu^-}e_r - e_r e_{\mu^-}) \\ &= \begin{cases} \tfrac{1}{2}(e_{\mu^-}e_r - e_{\mu^-}e_r) & \text{if } r \in \mu^+ \\ \tfrac{1}{2}(e_{\mu^-}e_r + e_{\mu^-}e_r) & \text{if } r \notin \mu^+ \end{cases} \\ &= \begin{cases} 0 & \text{if } r \in \mu^- \\ e_{\mu^-}e_r & \text{if } r \notin \mu^- \end{cases} \end{aligned} \tag{2.34}$$

This shows that the proposed result satisfies the conditions of the definition of the outer product. ■

Corollary 2.3.2 *Let $\mu \in P_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$ with $r \notin \mu$. Then*

$$e_\mu \wedge e_r = (-1)^{|\mu|} e_r \wedge e_\mu$$

Proof. The proof follows directly from the definition of the outer product. Since $r \notin \mu$ we have $e_\mu \wedge e_r = e_\mu e_r$. Now, moving e_r to the other side of e_μ involves $|\mu|$ number of switches of e's. Hence, we get a factor of $(-1)^{|\mu|}$. ■

Corollary 2.3.3 *Let $\mu, \nu \in P_{\mathcal{V}^n}$ and $\mu \cap \nu = \emptyset$ then*

$$e_\mu \wedge e_\nu = (-1)^{(|\mu||\nu|)} e_\nu \wedge e_\mu$$

Proof. The proof follows directly from the associativity of the outer product and corollary 2.3.2.

2.3.2 The Inner Product

Definition 2.3.4 *Let $\mu, \nu \in P_{\mathcal{V}^n}$, then the **inner product** of reduced basis blade $e_\mu, e_\nu \in \mathcal{G}^n$ is written $e_\mu{\cdot}e_\nu$ and defined as*

$$e_\mu{\cdot}e_\nu \equiv \begin{cases} e_\mu e_\nu & \textit{if } \mu \cap \nu \neq \emptyset \textbf{ and} \\ & \mu \cap \nu = \begin{cases} \mu & 0 < |\mu| \leq |\nu| \\ \nu & 0 < |\nu| < |\mu| \end{cases} \\ 0 & \textit{otherwise} \end{cases}$$

The condition $\mu \cap \nu \neq \emptyset$ simply says that the inner product is not the outer product. This ensures that $\mu \neq \emptyset$ *and* $\nu \neq \emptyset$. The other condition ensures that the inner product is only non-zero if all the es of one reduced basis blade are also contained in the other simple blade.

The inner product is extended to general multivectors through the following two definitions.

Definition 2.3.5 *Let $\mu, \nu \in P_{\mathcal{V}^n}$ and accordingly $e_\mu, e_\nu \in \mathcal{G}^n$. Also let $\alpha \in \mathcal{R}$. Then*

$$(\alpha e_\mu){\cdot}e_\nu \equiv e_\mu{\cdot}(\alpha e_\nu) \equiv \alpha(e_\mu{\cdot}e_\nu)$$

The inner product of a scalar α with a reduced basis blade e_μ is therefore given by

$$\alpha{\cdot}e_\mu = (\alpha e_\emptyset){\cdot}e_\mu = \alpha(e_\emptyset{\cdot}e_\mu) = 0 \tag{2.35}$$

Definition 2.3.6 *Let $A, B, C \in \mathcal{G}^n$, then*

$$\begin{aligned} A{\cdot}(B + C) &\equiv (A{\cdot}B) + (A{\cdot}C) \\ (B + C){\cdot}A &\equiv (B{\cdot}A) + (C{\cdot}A) \end{aligned}$$

As for the outer product, the inner product between a reduced basis blade and some element $e_r \in \mathcal{G}^n$ can also be expressed in terms of the commutator and anti-commutator products.

Theorem 2.3.2 *Let $\mu \in P_{\mathcal{V}^n}$, $r \in N_{\mathcal{V}^n}$ and hence $e_\mu, e_r \in \mathcal{G}^n$. Then the inner product between e_μ and e_r can be written as*

$$e_\mu{\cdot}e_r = \begin{cases} e_\mu \underline{\times} e_r & \textit{if } |\mu| \textit{ is even or zero} \\ e_\mu \overline{\times} e_r & \textit{if } |\mu| \textit{ is odd} \end{cases}$$

Proof. Again we will split the proof into two parts: for even and odd reduced basis blades. Let $\mu^+ \in P^+_{\mathcal{V}^n}$, $\mu^- \in P^-_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$. Also $e_{\mu^+}, e_{\mu^-}, e_r \in \mathcal{G}^n$.

$$\begin{aligned}
e_{\mu^+} \underline{\times} e_r &= \tfrac{1}{2}(e_{\mu^+} e_r - e_r e_{\mu^+}) \\
&= \begin{cases} \tfrac{1}{2}(e_{\mu^+} e_r + e_{\mu^+} e_r) & \text{if } r \in \mu^+ \\ \tfrac{1}{2}(e_{\mu^+} e_r - e_{\mu^+} e_r) & \text{if } r \notin \mu^+ \end{cases} \\
&= \begin{cases} e_{\mu^+} e_r & \text{if } r \in \mu^+ \\ 0 & \text{if } r \notin \mu^+ \end{cases}
\end{aligned} \tag{2.36}$$

Furthermore,

$$\begin{aligned}
e_{\mu^-} \overline{\times} e_r &= \tfrac{1}{2}(e_{\mu^-} e_r + e_r e_{\mu^-}) \\
&= \begin{cases} \tfrac{1}{2}(e_{\mu^-} e_r + e_{\mu^-} e_r) & \text{if } r \in \mu^- \\ \tfrac{1}{2}(e_{\mu^-} e_r - e_{\mu^-} e_r) & \text{if } r \notin \mu^- \end{cases} \\
&= \begin{cases} e_{\mu^-} e_r & \text{if } r \in \mu^- \\ 0 & \text{if } r \notin \mu^- \end{cases}
\end{aligned} \tag{2.37}$$

This shows that the proposed result satisfies the conditions of the definition of the inner product. ■

Corollary 2.3.4 *Let $\mu \in P_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$ with $r \in \mu$. Then*

$$e_\mu \cdot e_r = (-1)^{(|\mu|-1)}\, e_r \cdot e_\mu$$

Proof. The proof follows directly from the definition of the inner product. Since $e_r \in e_\mu$ we have $e_\mu \cdot e_r = e_\mu\, e_r$. Now, moving e_r to the other side of e_μ involves $|\mu| - 1$ switches of e's (because e_r is already contained in e_μ). Hence, we get a factor of $(-1)^{(|\mu|-1)}$. ■

Lemma 2.3.1 *Let $\mu \in P_{\mathcal{V}^n}$ and $r \in N_{\mathcal{V}^n}$ with $r \in \mu$. Also, let r be at position i in μ. Denote the corresponding element in e_μ by e_{μ_i}. Then*

$$e_\mu \cdot e_r = (-1)^{(|\mu|-i)}\, (e_r)^2\, e_{\mu'}$$

where μ' contains all elements of μ apart from r.

Proof. The proof is straightforward. Since we assume that r is contained in μ the inner product $e_\mu \cdot e_r$ reduces to the geometric product by definition. Furthermore, we assume that the element in e_μ identical to e_r is at position i. Hence, if we move this element (e_{μ_i}) to the right end of the reduced basis blade we have to make $|\mu| - i$ switches, which introduces the same number of (-1) factors. Since e_{μ_i} is identical to e_r by definition, they form a scalar object and we are left with the proposed equation. ■

This lemma shows that the inner product can be regarded as a **step-down operator**, which either decreases the grade of a reduced basis blade or gives zero.

2.4 Further Development

2.4.1 The Grade-Projection Operator

In this section we will introduce the grade-projection operator which will allow us to derive some useful identities.

Definition 2.4.1 *Let $\mu, \nu \in P_{\mathcal{V}^n}$ and accordingly $e_\mu, e_\nu \in \mathcal{G}^n$. Then the grade-projection bracket is defined as*

$$\langle e_\mu e_\nu \rangle_r \equiv \begin{cases} e_\mu e_\nu & \textit{if } (e_\mu e_\nu) \textit{ is of grade } r \\ 0 & \textit{otherwise} \end{cases}$$

If r is zero, there may be no index given, i.e. $\langle e_\mu e_\nu \rangle_0 \equiv \langle e_\mu e_\nu \rangle$.

The definition of the grade-projection bracket is extended to general multivectors through the following two definitions.

Definition 2.4.2 *Let $A \in \mathcal{G}^n$ and $\alpha \in \mathcal{R}$. Then*

$$\langle \alpha A \rangle_r \equiv \alpha \langle A \rangle_r$$

Definition 2.4.3 *Let $A, B \in \mathcal{G}^n$. Then*

$$\langle A + B \rangle_r \equiv \langle A \rangle_r + \langle B \rangle_r$$

A general multivector is a linear combination of reduced basis blades. Therefore, a grade-projection bracket applied to a general multivector will only retain the linear combination of those reduced basis blades contained in the multivector, which are of the specified grade.

Using the grade-projection bracket we can expand the geometric product between two multivectors $A, B \in \mathcal{G}^n$ as

$$AB = \sum_{k=0}^{n} \langle AB \rangle_k \tag{2.38}$$

Let $\mu^i \in P^r_{\mathcal{V}^n}$ and $\nu^j \in P^s_{\mathcal{V}^n}$, then

$$A = \sum_{i}^{|P^r_{\mathcal{V}^n}|} \alpha^i e_{\mu^i}\,, \quad B = \sum_{i}^{|P^s_{\mathcal{V}^n}|} \beta^i e_{\nu^i}, \tag{2.39}$$

are two homogeneous multivectors of grade r and s, respectively. From lemma 2.2.1 it follows that their geometric product can be expanded as

$$AB = \langle AB \rangle_{|r-s|} + \langle AB \rangle_{|r-s|+2} + \cdots + \langle AB \rangle_{r+s}. \tag{2.40}$$

This becomes clear if we substitute equation (2.39) into equation (2.40).

$$AB = \sum_{i,j} \alpha^i \beta^j \; e_{\mu^i} e_{\nu^j} \tag{2.41}$$

If $\mu^i \subseteq \nu^j$ or $\nu^j \subseteq \mu^i$, then we get an element of grade $|r-s|$ in the above sum. On the other hand, if $\mu^i \cap \nu^j = \emptyset$ then the corresponding element in the sum is of grade $r+s$. For $|\mu^i \cap \nu^j| = i$, i.e. μ and ν have i elements in common, we obtain elements of grade $r+s-2i$.

Theorem 2.4.1 *The inner and outer product between two reduced basis blades $e_\mu, e_\nu \in \mathcal{G}^n$ can also be expressed in bracket notation.*

$$e_\mu \cdot e_\nu = \langle e_\mu e_\nu \rangle_{|\,|\mu|-|\nu|\,|} \quad \textit{if } \mu \neq \emptyset \textbf{\textit{ and }} \nu \neq \emptyset \tag{2.42a}$$

$$e_\mu \wedge e_\nu = \langle e_\mu e_\nu \rangle_{|\mu|+|\nu|} \tag{2.42b}$$

Proof. To see that the expressions for the inner and outer product are true, recall the definition of the grade projection bracket. Let $e_\mu, e_\nu \in \mathcal{G}^n$ with $\mu \neq \emptyset$ and $\nu \neq \emptyset$. Then $\langle e_\mu e_\nu \rangle_{|\,|\mu|-|\nu|\,|}$ is only non-zero if the grade of $(e_\mu e_\nu)$ is $|\,|\mu| - |\nu|\,|$. This can only be the case if either reduced basis blade is completely contained within the other. However, this is just the definition of the inner product. Similarly for the outer product. Here $\langle e_\mu e_\nu \rangle_{k+l}$ is only non-zero if the grade of $(e_\mu e_\nu)$ is $k+l$. This is only possible if e_μ and e_ν have no elements in common. ■

The bracket notation can also be used to find some general identities. We will derive two important identities here. Let $\mu \in P^r_{\mathcal{V}^n}$, $\nu \in P^s_{\mathcal{V}^n}$ and $\tau \in P^t_{\mathcal{V}^n}$, such that $e_\mu, e_\nu, e_\tau \in \mathcal{G}^n$ are of grade r, s and t, respectively.

$$\begin{aligned}(e_\mu{\cdot}e_\nu){\cdot}e_\tau &= \langle e_\mu e_\nu \rangle_{|r-s|}{\cdot}e_\tau \\ &= \langle e_\mu e_\nu e_\tau \rangle_{||r-s|-t|}\end{aligned} \tag{2.43}$$

Assume that $s \geq r+t$. Then $||r-s|-t| = (s-r-t) = |r-|s-t||$. Therefore,

$$\begin{aligned}(e_\mu{\cdot}e_\nu){\cdot}e_\tau &= \langle e_\mu e_\nu e_\tau \rangle_{|r-|s-t||} \\ &= e_\mu{\cdot}\langle e_\nu e_\tau \rangle_{|s-t|} \\ &= e_\mu{\cdot}(e_\nu{\cdot}e_\tau)\end{aligned} \tag{2.44}$$

This is a kind of restricted associative law of the inner product.

$$\begin{aligned}(e_\mu{\wedge}e_\nu){\cdot}e_\tau &= \langle e_\mu e_\nu \rangle_{(r+s)}{\cdot}e_\tau \\ &= \langle e_\mu e_\nu e_\tau \rangle_{|(r+s)-t|}\end{aligned} \tag{2.45}$$

Assume that $t \geq r+s$. Then $|(r+s)-t| = (t-r-s) = |r-|s-t||$. Hence,

$$\begin{aligned}(e_\mu{\wedge}e_\nu){\cdot}e_\tau &= \langle e_\mu e_\nu e_\tau \rangle_{|r-|s-t||} \\ &= e_\mu{\cdot}\langle e_\nu e_\tau \rangle_{|s-t|} \\ &= e_\mu{\cdot}(e_\nu{\cdot}e_\tau)\end{aligned} \tag{2.46}$$

To summarise, we have found the following two identities using the bracket notation:

$$(e_\mu{\cdot}e_\nu){\cdot}e_\tau = e_\mu{\cdot}(e_\nu{\cdot}e_\tau) \quad \text{if } |\nu| \geq |\mu| + |\tau| \tag{2.47a}$$

$$(e_\mu{\wedge}e_\nu){\cdot}e_\tau = e_\mu{\cdot}(e_\nu{\cdot}e_\tau) \quad \text{if } |\tau| \geq |\mu| + |\nu| \tag{2.47b}$$

Lemma 2.4.1 *Let $\mu \in P_{\mathcal{V}^n}$ and let $\{\nu^i\}$ and $\{\tau^i\}$ be two partitions of μ. That is,*

$$\mu = \nu^1 \cup \nu^2 \cup \cdots \cup \nu^k = \tau^1 \cup \tau^2 \cup \cdots \cup \tau^l \tag{2.48}$$

Note that the subsets $\{\nu^i\}$ *do not necessarily contain the same number of elements. The same is true for the* $\{\tau^i\}$. $e_\mu \in \mathcal{G}^n$ *can therefore be written as*

$$e_\mu = e_{\nu^1} \wedge e_{\nu^2} \wedge \ldots \wedge e_{\nu^k} = e_{\tau^1} \wedge e_{\tau^2} \wedge \ldots \wedge e_{\tau^l} \tag{2.49}$$

Let these two parameterisations of e_μ *be mixed arbitrarily within a scalar projection bracket. For example, we might have the following mixing.*

$$\langle e_{\nu^{i_1}} e_{\tau^{j_1}} \ldots e_{\tau^{j_l}} e_{\nu^{i_k}} \rangle$$

This bracket has to be non-zero because it contains twice e_μ, *and* $(e_\mu)^2$ *is a scalar. Now we can move the reduced basis blade at the very right all the way to the beginning of the bracket without changing the result of the bracket. That is,*

$$\langle e_{\nu^{i_1}} e_{\tau^{j_1}} \ldots e_{\tau^{j_l}} e_{\nu^{i_k}} \rangle = \langle e_{\nu^{i_k}} e_{\nu^{i_1}} e_{\tau^{j_1}} \ldots e_{\tau^{j_l}} \rangle$$

Proof. We know that e_μ is included twice within the bracket. Therefore, when moving $e_{\nu^{i_k}}$ all the way through the bracket, it will move twice through all those e's it does not contain and once through those e's it contains. Hence, an even number of (-1) factors is introduced and thus the overall sign of the bracket is not changed. ■

We can extend this lemma to general multivectors.

Theorem 2.4.2 *The cyclic reordering property. Let* $A, B, C \in \mathcal{G}^n$ *be three general multivectors. Then*

$$\langle ABC \rangle = \langle CAB \rangle$$

Proof. The proof can be found quite easily by using the distributivity of the grade-projection bracket, and then applying lemma 2.4.1. In general the geometric product ABC can be written as a linear combination of reduced basis blades. Let $\mu^i, \nu^i, \tau^i \in P_{\mathcal{V}^n}$, $\alpha^i, \beta^i, \rho^i \in \mathcal{R}$ and

$$A \equiv \sum_i \alpha^i \, e_{\mu^i} \; ; \quad B \equiv \sum_i \beta^i \, e_{\nu^i} \; ; \quad C \equiv \sum_i \rho^i \, e_{\tau^i},$$

then

$$ABC = \sum_{i,j,k} \alpha^i \beta^j \rho^k \, e_{\mu^i} e_{\nu^j} e_{\tau^k}.$$

Also let $\mathcal{S}$ be the set of those sets of indices $\{i, j, k\}$ for which $(e_{\mu^i} e_{\nu^j} e_{\tau^k})$ is of grade zero, i.e. a scalar. Then

$$\langle ABC \rangle = \sum_{\{i,j,k\} \in \mathcal{S}} \alpha^i \beta^j \rho^k \, e_{\mu^i} e_{\nu^j} e_{\tau^k}.$$

Each term in the above sum can be expressed as some $(e_\sigma)^2$ of which the respective $e_{\mu^i} e_{\nu^j} e_{\tau^k}$ are just a parameterisation. Hence, it follows from lemma 2.4.1 that in this case we have

$$e_{\mu^i} e_{\nu^j} e_{\tau^k} = e_{\tau^k} e_{\mu^i} e_{\nu^j} \; ; \quad \{i, j, k\} \in \mathcal{S}.$$

Therefore,

$$\begin{aligned} \langle ABC \rangle &= \sum_{\{i,j,k\} \in \mathcal{S}} \alpha^i \beta^j \rho^k \, e_{\mu^i} e_{\nu^j} e_{\tau^k} \\ &= \sum_{\{i,j,k\} \in \mathcal{S}} \rho^k \alpha^i \beta^j \, e_{\tau^k} e_{\mu^i} e_{\nu^j} \\ &= \langle CAB \rangle \end{aligned}$$

■

2.4.2 The Reversion Operator

Definition 2.4.4 *Let $\mu \in P_{\mathcal{V}^n}$ and write the corresponding reduced basis blade $e_\mu \in \mathcal{G}^n$ as $e_\mu = e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k}$. Then the* ***reverse*** *of e_μ, written $\tilde{e}_\mu$, is defined as*

$$\tilde{e}_\mu = e_{\mu_k} e_{\mu_{k-1}} \ldots e_{\mu_2} e_{\mu_1}$$

The reversion operation is extended to general multivectors by the following two definitions.

Definition 2.4.5 *Let $e_\mu \in \mathcal{G}^n$ and $\alpha \in \mathcal{R}$. Then*

$$(\alpha e_\mu)^\sim \equiv \alpha \tilde{e}_\mu$$

Definition 2.4.6 *Let $A, B \in \mathcal{G}^n$, then*

$$(A + B)^\sim \equiv \tilde{A} + \tilde{B}$$

It follows that the reverse of a general multivector is the sum of the reverses of its constituent reduced basis blades.

Lemma 2.4.2 *Let $\mu \in P^k_{\mathcal{V}^n}$ and $e_\mu \in \mathcal{G}^n$. Then,*

$$\tilde{e}_\mu = (-1)^{k(k-1)/2}\, e_\mu$$

Proof.

$$\begin{aligned} \tilde{e}_\mu &= e_{\mu_k} \wedge e_{\mu_{k-1}} \wedge \ldots \wedge e_2 \wedge e_1 \\ &= (-1)^{k-1}(-1)^{k-2} \cdots (-1)^2 (-1)^1\, e_\mu \\ &= (-1)^{k(k-1)/2}\, e_\mu \end{aligned} \tag{2.50}$$

■

Lemma 2.4.3 *Let $\mu \in P^k_{\mathcal{V}^n}$ and $\nu \in P^l_{\mathcal{V}^n}$, such that $e_\mu, e_\nu \in \mathcal{G}^n$ can be written as $e_\mu = e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k}$ and $e_\nu = e_{\nu_1} e_{\nu_2} \ldots e_{\nu_k}$. Then,*

$$(e_\mu e_\nu)\tilde{} = \tilde{e}_\nu \tilde{e}_\mu$$

Proof.

$$\begin{aligned} (e_\mu e_\nu)\tilde{} &= \Big((e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k})(e_{\nu_1} e_{\nu_2} \ldots e_{\nu_l})\Big)\tilde{} \\ &= (e_{\mu_1} e_{\mu_2} \ldots e_{\mu_k} e_{\nu_1} e_{\nu_2} \ldots e_{\nu_l})\tilde{} \\ &= e_{\nu_l} \ldots e_{\nu_2} e_{\nu_1} e_{\mu_k} \ldots e_{\mu_2} e_{\mu_1} \\ &= \tilde{e}_\nu \tilde{e}_\mu \end{aligned} \tag{2.51}$$

■

Definition 2.4.7 *The* ***inverse*** *of a reduced basis blade $e_\mu \in \mathcal{G}^n$, written e_μ^{-1}, is defined such that*

$$e_\mu e_\mu^{-1} = 1$$

Therefore, we can write the inverse of a reduced basis blade e_μ explicitly as

$$e_\mu^{-1} = \frac{\tilde{e}_\mu}{e_\mu \tilde{e}_\mu} \tag{2.52}$$

Since $\mathcal{G}^n$ is not degenerate and thus $e_\mu \tilde{e}_\mu \in \mathcal{R}$ and $e_\mu \tilde{e}_\mu \neq 0$, the inverse is well defined. The definition of the inverse allows us to define division of two reduced basis blades.

Definition 2.4.8 *Let $\mu, \nu \in P_{\mathcal{V}^n}$ and $e_\mu, e_\nu \in \mathcal{G}^n$. Then the* ***division*** *of e_μ by e_ν is defined as*

$$\frac{e_\mu}{e_\nu} := e_\mu e_\nu^{-1} = \frac{e_\mu \tilde{e}_\nu}{e_\nu \tilde{e}_\nu}$$

2.5 More on Vectors in $\mathcal{G}^n$

In 2.2.3 we defined vectors in an algebraic way to obtain a GA which is appropriate for our purposes. In this section we want to generalize the concept of vectors in $\mathcal{G}^n$ by introducing r-vectors: homogeneous multivectors of grade r. Each complete set of r-vectors in some $\mathcal{G}^n$ forms a vector space, however, not one of the type as defined in 2.2.3 for $r > 1$.

In the second part of this section we will introduce *blades*. A blade is the outer product of a number of vectors of $\mathcal{V}^n$. We will then discuss some essential properties of blades. In particular, the expansion of the inner product of two blades is very important.

Definition 2.5.1 *An* ***r-vector*** *is a homogeneous multivector of grade r. A 1-vector will also simply be called a vector.*

Definition 2.5.2 *Two r-vectors A and B are called* ***orthogonal*** *if $A{\cdot}B = 0$. Furthermore, a set of r-vectors is called orthogonal if every r-vector in the set is orthogonal to all other r-vectors in the set.*

Definition 2.5.3 *An r-vector A is called* ***linearly dependent*** *on a set of r-vectors $\{B_i\}$ if there is a set of scalars $\{\alpha^i\}$ such that $A = \sum_i \alpha^i B_i$. If no such set of scalars exists then A is said to be* ***linearly independent*** *of the set $\{B_i\}$.*

Definition 2.5.4 *A set of r-vectors $\{A_i\}$ is called* ***mutually linearly independent*** *if no r-vector from the set is linearly dependent on the other r-vectors of the set.*

Definition 2.5.5 *A set of n mutually linearly independent r-vectors is called the* ***basis*** *of an* ***r-vector space*** *$\mathcal{V}_r^n$ of* ***dimension*** *n. If the set of r-vectors is also orthogonal, the basis is called an orthogonal basis. $\mathcal{V}_r^n$ is a subset of $\mathcal{G}^m$, containing all linear combinations of the basis r-vectors, but not their geometric or outer products. Note that an n-dimensional 1-vector space will still be written $\mathcal{V}^n$.*

In $\mathcal{G}^n$ there are ${C_n}^r$ r-vectors, where ${C_n}^r$ stands for the number of possible combinations of n objects into groups of r. Each complete set of r-vectors in some $\mathcal{G}^n$ forms a basis of a $d = {C_n}^r$ dimensional r-vector space $\mathcal{V}_r^d$.

Note that, for example, the geometric product of a 1-vector and a 2-vector will give a multivector with grade 1 and grade 3 components. This gives the motivation for the name "multivector" of general objects in $\mathcal{G}^n$: a multivector is a sum of vectors of different grade.

We will now consider 1-vectors in more detail. Let the $\{\alpha^i\}$ be a set of scalars and let $B_E = \{e_1, e_2, \ldots, e_n\}$ be an orthonormal basis of $\mathcal{V}^n$, as before. Then a (1-)vector $\boldsymbol{a}$ may be written as

$$\boldsymbol{a} = \sum_{i=1}^{n} \alpha^i e_i \tag{2.53}$$

Note that we will write vectors as lower case bold letters from now on. Furthermore, we will use the convention that if a superscript index is repeated as a subscript a summation over the range of the index is implied. That is, $\sum_{i=1}^{n} \alpha^i e_i \equiv \alpha^i e_i$. Because of the distributivity of the geometric, commutator and anti-commutator products they can be extended immediately to vectors.

$$\boldsymbol{ab} = \boldsymbol{a} \underline{\times} \boldsymbol{b} + \boldsymbol{a} \overline{\times} \boldsymbol{b} \tag{2.54}$$

Theorems 2.3.1 and 2.3.2 also extend immediately to vectors. Therefore, we can write equation (2.54) as

$$\boldsymbol{ab} = \boldsymbol{a} \wedge \boldsymbol{b} + \boldsymbol{a} \cdot \boldsymbol{b} \tag{2.55}$$

This is sometimes taken as the definition of the geometric product. However, using this definition as the starting point, makes the derivation of many properties of the inner and outer product more difficult.

Note that the inner product of two vectors is equivalent to the standard scalar product of vectors.

$$\begin{aligned} \boldsymbol{a} \cdot \boldsymbol{b} &= \boldsymbol{a} \overline{\times} \boldsymbol{b} \\ &= (\alpha^i e_i) \overline{\times} (\beta^j e_j) \\ &= \alpha^i \beta^i (e_i)^2 \in \mathcal{R} \end{aligned} \tag{2.56}$$

since $e_i \overline{\times} e_j = 0$ if $i \neq j$.

Definition 2.5.6 *Let the $\{\boldsymbol{a}_i\} \in \mathcal{V}^n$ be some set of vectors. Then a* ***blade*** *of* ***grade*** *k will be denoted $A_{\langle k \rangle}$ and is defined by*

$$A_{\langle k \rangle} = \boldsymbol{a}_{i_1} \wedge \boldsymbol{a}_{i_2} \wedge \ldots \wedge \boldsymbol{a}_{i_k} \tag{2.57}$$

From the definition of vectors and the distributivity of the outer product it follows that every k-blade is also a k-vector. However, a general k-vector cannot necessarily be expressed as a k-blade. Also note that every reduced basis blade is a blade, but the reverse does not hold in general.

Lemma 2.5.1 *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{V}^n$, then $\boldsymbol{a} \wedge \boldsymbol{b} = 0$ if and only if $\boldsymbol{a}$ and $\boldsymbol{b}$ are linearly dependent.*

Proof. First we show the "if" part.

$$\boldsymbol{a} \wedge \boldsymbol{b} = \boldsymbol{a} \wedge (\alpha \boldsymbol{a}) = \alpha(\boldsymbol{a} \wedge \boldsymbol{a}) = 0 \tag{2.58}$$

For the "only if" part we assume $\boldsymbol{a} \wedge \boldsymbol{b} = 0$, and $\boldsymbol{a}$ and $\boldsymbol{b}$ are linearly independent. Therefore, $\boldsymbol{a}$ has to have at least one element $e_r \in \mathcal{E}^n$ that is not contained in $\boldsymbol{b}$, or vice versa. But the outer product of that e_r with $\boldsymbol{b}$ (or $\boldsymbol{a}$) will be non-zero. This contradicts the assumptions. ■

Corollary 2.5.1 *Let the $\{\boldsymbol{a}_i\} \in \mathcal{V}^n$ be a set of k vectors. Then, if $\boldsymbol{a}_{i_1} \wedge \boldsymbol{a}_{i_2} \wedge \ldots \wedge \boldsymbol{a}_{i_k} \neq 0$, the set $\{\boldsymbol{a}_i\}$ forms a basis of a vector space $\mathcal{V}^k$.*

Proof. The proof follows immediately from lemma 2.5.1.

Definition 2.5.7 *Let the $\{\boldsymbol{a}_i\}$ form a basis of $\mathcal{V}^n$. Then the blade*

$$I_{a_n} = \boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_n$$

is called the ***characteristic pseudoscalar*** *of $\mathcal{G}^n$ for basis $\{\boldsymbol{a}_i\}$. If it is clear that we are working within a specific geometric algebra, the characteristic pseudoscalar for basis $\{\boldsymbol{a}_i\}$ will simply be written as I_a.*

Corollary 2.5.2 *Let I_a be a characteristic pseudoscalar of $\mathcal{G}^n$, and let $b \in \mathcal{V}^n$. Then $I_a \wedge b = 0$.*

Proof. From definition 2.5.7 we know that $I_a = \boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_n$ where the $\{\boldsymbol{a}_i\}$ form a basis of $\mathcal{V}^n$. Since $b \in \mathcal{V}^n$ it has to be linearly dependent on the $\{\boldsymbol{a}_i\}$. Hence, from lemma 2.5.1 it follows that $I_a \wedge b = 0$. ■

This tells us that a characteristic pseudoscalar of $\mathcal{G}^n$ spans the whole space. Note that the pseudoscalars of some $\mathcal{G}^n$ can only differ by a scalar factor. In fact, it is not too difficult to show the following relation. Let $B_a = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ be an arbitrary basis of $\mathcal{V}^n$, and let $B_E = \{e_1, e_2, \ldots, e_n\}$ be an orthonormal basis of the same $\mathcal{V}^n$. The vectors of basis B_a may therefore be expressed in terms of B_E as $\boldsymbol{a}_i = \alpha_i^j e_j$, where we sum implicitly over j. α_i^j is a matrix giving the linear mapping between B_E and B_a. Then it can be shown that

$$\boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_n = \det(\alpha_i^j)\; I_{\mathcal{V}^n}. \tag{2.59}$$

In the following we will always relate a characteristic pseudoscalar to the *standard pseudoscalar* $I = e_1 e_2 \ldots e_n$, once we have chosen an orthonormal basis B_E. In particular, we will write

$$I_a = \rho_a I, \tag{2.60}$$

where ρ_a is the determinant of the linear mapping matrix, as given above.

The grade-projection operator extends directly to general blades, due to its distributivity. Therefore, identities (2.47) can also be extended to blades. That is,

$$\begin{aligned} (A_{\langle r\rangle} \cdot B_{\langle s\rangle}) \cdot C_{\langle t\rangle} &= A_{\langle r\rangle} \cdot (B_{\langle s\rangle} \cdot C_{\langle t\rangle}) \quad \text{if } s \geq r + t && \text{(2.61a)} \\ (A_{\langle r\rangle} \wedge B_{\langle s\rangle}) \cdot C_{\langle t\rangle} &= A_{\langle r\rangle} \cdot (B_{\langle s\rangle} \cdot C_{\langle t\rangle}) \quad \text{if } t \geq r + s && \text{(2.61b)} \end{aligned}$$

In the following we will derive some vector identites. Equations (2.15) to (2.18) are very useful for this purpose, as we know from theorems 2.3.1 and 2.3.2 how to translate the inner and outer products into commutator and anti-commutator products.

Lemma 2.5.2 *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{V}^n$ and $A_{\langle k\rangle} \in \mathcal{G}^n$. Furthermore, assume that $A_{\langle k\rangle} \wedge \boldsymbol{a} \neq 0$ and $A_{\langle k\rangle} \cdot \boldsymbol{b} \neq 0$. Then*

$$(A_{\langle k\rangle} \wedge \boldsymbol{a}) \cdot \boldsymbol{b} = A_{\langle k\rangle} (\boldsymbol{a} \cdot \boldsymbol{b}) - (A_{\langle k\rangle} \cdot \boldsymbol{b}) \wedge \boldsymbol{a}$$

Proof. We will perform the proof by translating the inner and outer products into commutator and anti-commutator products according to theorems

2.3.1 and 2.3.2, and then using equations (2.15b) and (2.18a) to expand the expressions. We have to consider two cases: for k even and odd. First assume k is odd.

$$\begin{aligned}(A_{\langle k\rangle}\wedge\boldsymbol{a})\cdot\boldsymbol{b} &= (A_{\langle k\rangle}\underline{\times}\boldsymbol{a})\underline{\times}\boldsymbol{b} \\ &= A_{\langle k\rangle}\overline{\times}(\boldsymbol{a}\overline{\times}\boldsymbol{b}) - (A_{\langle k\rangle}\overline{\times}\boldsymbol{b})\overline{\times}\boldsymbol{a} \qquad \text{from (2.15b)} \\ &= A_{\langle k\rangle}(\boldsymbol{a}\cdot\boldsymbol{b}) - (A_{\langle k\rangle}\cdot\boldsymbol{b})\wedge\boldsymbol{a}\end{aligned} \tag{2.62}$$

Next we consider the case of even k.

$$\begin{aligned}(A_{\langle k\rangle}\wedge\boldsymbol{a})\cdot\boldsymbol{b} &= (A_{\langle k\rangle}\overline{\times}\boldsymbol{a})\overline{\times}\boldsymbol{b} \\ &= A_{\langle k\rangle}\overline{\times}(\boldsymbol{a}\overline{\times}\boldsymbol{b}) - (A_{\langle k\rangle}\underline{\times}\boldsymbol{b})\underline{\times}\boldsymbol{a} \qquad \text{from (2.18a)} \\ &= A_{\langle k\rangle}(\boldsymbol{a}\cdot\boldsymbol{b}) - (A_{\langle k\rangle}\cdot\boldsymbol{b})\wedge\boldsymbol{a}\end{aligned} \tag{2.63}$$

That is, independent of whether k is even or odd we always obtain the proposed result. ■

In a similar way we can prove the following identities.

$$A_{\langle k\rangle}\underline{\times}(\boldsymbol{a}\wedge\boldsymbol{b}) = (A_{\langle k\rangle}\wedge\boldsymbol{a})\cdot\boldsymbol{b} - (A_{\langle k\rangle}\wedge\boldsymbol{b})\cdot\boldsymbol{a} \tag{2.64a}$$

$$A_{\langle k\rangle}\overline{\times}(\boldsymbol{a}\wedge\boldsymbol{b}) = (A_{\langle k\rangle}\cdot\boldsymbol{a})\cdot\boldsymbol{b} + (A_{\langle k\rangle}\wedge\boldsymbol{a})\wedge\boldsymbol{b} \tag{2.64b}$$

$$(A_{\langle k\rangle}\cdot\boldsymbol{a})\cdot\boldsymbol{b} = -(A_{\langle k\rangle}\cdot\boldsymbol{b})\cdot\boldsymbol{a} \tag{2.65a}$$

$$(A_{\langle k\rangle}\wedge\boldsymbol{a})\wedge\boldsymbol{b} = -(A_{\langle k\rangle}\wedge\boldsymbol{b})\wedge\boldsymbol{a} \tag{2.65b}$$

Note that equation (2.65b) simply reflects the associativity of the outer product and equation (2.65a) can also be derived using equation (2.61b). Equations (2.64), on the other hand, are most easily derived in the way presented here. They give the two parts of the geometric product of some blade $A_{\langle k\rangle}$ with a bivector $(\boldsymbol{a}\wedge\boldsymbol{b})$. That is,

$$\begin{aligned}A_{\langle k\rangle}\,(\boldsymbol{a}\wedge\boldsymbol{b}) &= A_{\langle k\rangle}\underline{\times}(\boldsymbol{a}\wedge\boldsymbol{b}) + A_{\langle k\rangle}\overline{\times}(\boldsymbol{a}\wedge\boldsymbol{b}) \\ &= (A_{\langle k\rangle}\wedge\boldsymbol{a})\cdot\boldsymbol{b} - (A_{\langle k\rangle}\wedge\boldsymbol{b})\cdot\boldsymbol{a} \\ &\quad +(A_{\langle k\rangle}\cdot\boldsymbol{a})\cdot\boldsymbol{b} + (A_{\langle k\rangle}\wedge\boldsymbol{a})\wedge\boldsymbol{b}\end{aligned} \tag{2.66}$$

We will now turn our attention to a very important formula in GA: the distributivity of the inner product with respect to the outer product. That is, we want to know how to expand an expression like $(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot\boldsymbol{b}$. To simplify the notation we will need the following definition.

Definition 2.5.8 *Let the $\{\boldsymbol{a}_i\}$ be a basis of $\mathcal{V}^n$, and let a blade $A_{\langle k\rangle}$ be defined as $A_{\langle k\rangle} = \boldsymbol{a}_{i_1}\wedge\boldsymbol{a}_{i_2}\wedge\ldots\wedge\boldsymbol{a}_{i_k}$. Then the expression $[A_{\langle k\rangle}\backslash\boldsymbol{a}_{i_l}]$ refers to the blade $A_{\langle k\rangle}$ with the element $\boldsymbol{a}_{i_l}$ taken out. All other elements stay in the same order. Furthermore, the expression $[A_{\langle k\rangle}\backslash\{\boldsymbol{a}_{i_l},\ldots,\boldsymbol{a}_{i_m}\}]$ refers to the blade $A_{\langle k\rangle}$ with the set of elements $\{\boldsymbol{a}_{i_l},\ldots,\boldsymbol{a}_{i_m}\}$ taken out.*

We can now state the relevant theorem.

Theorem 2.5.1 *Let the $\{\boldsymbol{a}_i\}$ be a basis of $\mathcal{V}^n$, and let a blade $A_{\langle k\rangle} \in \mathcal{G}^n$ be defined as $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\boldsymbol{a}_2\wedge\ldots\wedge\boldsymbol{a}_k$. Also let $\boldsymbol{b} \in \mathcal{V}^n$. Then,*

$$A_{\langle k\rangle}\cdot\boldsymbol{b} = \sum_{i=0}^{k-1} (-1)^i(\boldsymbol{a}_{k-i}\cdot\boldsymbol{b})\,[A_{\langle k\rangle}\backslash\boldsymbol{a}_{k-i}]$$

Proof. The proof is straightforward, using lemma 2.5.2.

$$\begin{aligned}
A_{\langle k\rangle}\cdot\boldsymbol{b} &= ([A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\wedge\boldsymbol{a}_k)\cdot\boldsymbol{b} \\
&= [A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\,(\boldsymbol{a}_k\cdot\boldsymbol{b}) - \Big([A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\cdot\boldsymbol{b}\Big)\wedge\boldsymbol{a}_k \\
&= [A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\,(\boldsymbol{a}_k\cdot\boldsymbol{b}) \\
&\quad -\Big([A_{\langle k\rangle}\backslash\{\boldsymbol{a}_k,\boldsymbol{a}_{k-1}\}]\,(\boldsymbol{a}_{k-1}\cdot\boldsymbol{b})\Big)\wedge\boldsymbol{a}_k \\
&\quad +\Big([A_{\langle k\rangle}\backslash\{\boldsymbol{a}_k,\boldsymbol{a}_{k-1}\}]\cdot\boldsymbol{b}\Big)\wedge\boldsymbol{a}_{k-1}\wedge\boldsymbol{a}_k \\
&= [A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\,(\boldsymbol{a}_k\cdot\boldsymbol{b}) - [A_{\langle k\rangle}\backslash\boldsymbol{a}_{k-1}]\,(\boldsymbol{a}_{k-1}\cdot\boldsymbol{b}) \\
&\quad +\Big([A_{\langle k\rangle}\backslash\{\boldsymbol{a}_k,\boldsymbol{a}_{k-1}\}]\cdot\boldsymbol{b}\Big)\wedge\boldsymbol{a}_{k-1}\wedge\boldsymbol{a}_k \\
&= \sum_{i=0}^{k-1} (-1)^i(\boldsymbol{a}_{k-i}\cdot\boldsymbol{b})\,[A_{\langle k\rangle}\backslash\boldsymbol{a}_{k-i}]
\end{aligned} \tag{2.67}$$

■

Now we can expand the inner product of two blades using equation (2.61b). For example, let $A_{\langle k\rangle}, B_{\langle l\rangle} \in \mathcal{G}^n$ with $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\boldsymbol{a}_2\wedge\ldots\wedge\boldsymbol{a}_k$, and $k \leq l$.

$$\begin{aligned}
A_{\langle k\rangle}\cdot B_{\langle l\rangle} &= ([A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\wedge\boldsymbol{a}_k)\cdot B_{\langle l\rangle} \\
&= [A_{\langle k\rangle}\backslash\boldsymbol{a}_k]\cdot(\boldsymbol{a}_k\cdot B_{\langle l\rangle})
\end{aligned} \tag{2.68}$$

We can also apply equation (2.61b) in a different way. If $k \leq l$, then

$$\begin{aligned}
&A_{\langle k\rangle}\cdot B_{\langle l\rangle} = \\
&\quad \sum_{\{j_i\}} \epsilon_{j_1 j_2\cdots j_l}\,\Big[A_{\langle k\rangle}\cdot(\boldsymbol{b}_{j_1}\wedge\boldsymbol{b}_{j_2}\wedge\ldots\wedge\boldsymbol{b}_{j_k})\Big]\,\boldsymbol{b}_{j_{k+1}}\wedge\ldots\wedge\boldsymbol{b}_{j_l},
\end{aligned} \tag{2.69}$$

where $\epsilon_{j_1 j_2 \cdots j_l}$ is $+1$ if the $\{j_i\}$ form an even permutation of $\{1, 2, \ldots, l\}$, -1 if they form an odd permutation and 0 if any two indices are equal.

It is helpful to see some examples of such expansions, since they are among the most frequently used manipulations in GA.

$$
\begin{aligned}
\boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) &= (\boldsymbol{a}\cdot\boldsymbol{b}_1)\boldsymbol{b}_2 - (\boldsymbol{a}\cdot\boldsymbol{b}_2)\boldsymbol{b}_1 && \text{(2.70a)}\\
\boldsymbol{a}\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2\wedge\boldsymbol{b}_3) &= (\boldsymbol{a}\cdot\boldsymbol{b}_1)(\boldsymbol{b}_2\wedge\boldsymbol{b}_3) - (\boldsymbol{a}\cdot\boldsymbol{b}_2)(\boldsymbol{b}_1\wedge\boldsymbol{b}_3) && \text{(2.70b)}\\
&\quad +(\boldsymbol{a}\cdot\boldsymbol{b}_3)(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)
\end{aligned}
$$

Furthermore,

$$
\begin{aligned}
(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2) &= \boldsymbol{a}_1\cdot\Big(\boldsymbol{a}_2\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)\Big) && \\
&= \boldsymbol{a}_2\cdot\boldsymbol{b}_1\ \boldsymbol{a}_1\cdot\boldsymbol{b}_2 \ -\ \boldsymbol{a}_2\cdot\boldsymbol{b}_2\ \boldsymbol{a}_1\cdot\boldsymbol{b}_1 && \text{(2.71a)}\\
(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2\wedge\boldsymbol{b}_3) &= \Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_2)\Big]\boldsymbol{b}_3 && \\
&\quad -\Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_1\wedge\boldsymbol{b}_3)\Big]\boldsymbol{b}_2 && \text{(2.71b)}\\
&\quad +\Big[(\boldsymbol{a}_1\wedge\boldsymbol{a}_2)\cdot(\boldsymbol{b}_2\wedge\boldsymbol{b}_3)\Big]\boldsymbol{b}_1 &&
\end{aligned}
$$

Because reversion is distributive, it can be extended immediately to blades. That is, if $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\ldots\wedge\boldsymbol{a}_k$ then

$$
\tilde{A}_{\langle k\rangle} = \boldsymbol{a}_k\wedge\ldots\wedge\boldsymbol{a}_1 \tag{2.72}
$$

It is not so clear that the inverse operation can also be extended to blades. In order to show that it *can* be extended to blades we need to show that for any $A_{\langle k\rangle} \in \mathcal{G}^n$ with $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\ldots\wedge\boldsymbol{a}_k$, $A^2_{\langle k\rangle} = A_{\langle k\rangle}A_{\langle k\rangle}$ is a scalar.

Lemma 2.5.3 *Let $A_{\langle k\rangle} \in \mathcal{G}^n$ with $A_{\langle k\rangle} = \boldsymbol{a}_1\wedge\ldots\wedge\boldsymbol{a}_k$, where the $\{\boldsymbol{a}_i\}$ are mutually linearly independent. Then $A^2_{\langle k\rangle} = A_{\langle k\rangle}A_{\langle k\rangle}$ is a scalar.*

Proof. If $A_{\langle k\rangle}$ was of grade 1, the proof would be trivial, because then we would have

$$
A_{\langle 1\rangle}A_{\langle 1\rangle} = A_{\langle 1\rangle}\wedge A_{\langle 1\rangle} + A_{\langle 1\rangle}\cdot A_{\langle 1\rangle} = A_{\langle 1\rangle}\cdot A_{\langle 1\rangle}
$$

which is a scalar. However, since k can be chosen arbitrarily we only know that

$$
A_{\langle k\rangle}A_{\langle k\rangle} = A_{\langle k\rangle}\mathbin{\underline{\times}}A_{\langle k\rangle} + A_{\langle k\rangle}\mathbin{\overline{\times}}A_{\langle k\rangle} = A_{\langle k\rangle}\mathbin{\overline{\times}}A_{\langle k\rangle}
$$

For brevity we define

$$A_{\langle k-i\rangle} \equiv [A_{\langle k\rangle} \backslash \{\boldsymbol{a}_k, \boldsymbol{a}_{k-1}, \ldots, \boldsymbol{a}_{k-i+1}\}].$$

Assuming k is even, we can now expand $A_{\langle k\rangle} \mathbin{\overline{\times}} A_{\langle k\rangle}$ as follows with the help of equation (2.16b).

$$\begin{aligned} A_{\langle k\rangle} \mathbin{\overline{\times}} A_{\langle k\rangle} &= (A_{\langle k-1\rangle} \mathbin{\underline{\times}} \boldsymbol{a}_k) \mathbin{\overline{\times}} A_{\langle k\rangle} \\ &= A_{\langle k-1\rangle} \mathbin{\underline{\times}} \underbrace{(\boldsymbol{a}_k \mathbin{\overline{\times}} A_{\langle k\rangle})}_{\boldsymbol{a}_k \wedge A_{\langle k\rangle} = 0} - (A_{\langle k-1\rangle} \mathbin{\underline{\times}} A_{\langle k\rangle}) \mathbin{\overline{\times}} \boldsymbol{a}_k \\ &= -(A_{\langle k-1\rangle} \mathbin{\underline{\times}} A_{\langle k\rangle}) \mathbin{\overline{\times}} \boldsymbol{a}_k \end{aligned} \tag{2.73}$$

The term in brackets may be expanded further using equation (2.17a).

$$\begin{aligned} A_{\langle k-1\rangle} \mathbin{\underline{\times}} A_{\langle k\rangle} &= (A_{\langle k-2\rangle} \mathbin{\overline{\times}} \boldsymbol{a}_{k-1}) \mathbin{\underline{\times}} A_{\langle k\rangle} \\ &= A_{\langle k-2\rangle} \mathbin{\underline{\times}} \underbrace{(\boldsymbol{a}_{k-1} \mathbin{\overline{\times}} A_{\langle k\rangle})}_{\boldsymbol{a}_{k-1} \wedge A_{\langle k\rangle} = 0} - (A_{\langle k-2\rangle} \mathbin{\overline{\times}} A_{\langle k\rangle}) \mathbin{\underline{\times}} \boldsymbol{a}_{k-1} \\ &= -(A_{\langle k-2\rangle} \mathbin{\overline{\times}} A_{\langle k\rangle}) \mathbin{\underline{\times}} \boldsymbol{a}_{k-1} \end{aligned} \tag{2.74}$$

Hence,

$$\begin{aligned} A_{\langle k\rangle} A_{\langle k\rangle} &= (-1)^2 \Big((A_{\langle k-2\rangle} \mathbin{\overline{\times}} A_{\langle k\rangle}) \mathbin{\underline{\times}} \boldsymbol{a}_{k-1} \Big) \mathbin{\overline{\times}} \boldsymbol{a}_k \\ &= (-1)^{k-1} \Big((\cdots ((\boldsymbol{a}_1 \mathbin{\underline{\times}} A_{\langle k\rangle}) \mathbin{\overline{\times}} \boldsymbol{a}_2) \cdots) \mathbin{\underline{\times}} \boldsymbol{a}_{k-1} \Big) \mathbin{\overline{\times}} \boldsymbol{a}_k \\ &= \Big(\cdots ((A_{\langle k\rangle} \cdot \boldsymbol{a}_1) \cdot \boldsymbol{a}_2) \cdots \Big) \cdot \boldsymbol{a}_k \end{aligned} \tag{2.75}$$

The last expression is obviously a scalar. This may be shown in a similar way when k is odd. ■

Therefore, the inverse of a blade $A_{\langle k\rangle} \in \mathcal{G}^n$ is given by

$$A_{\langle k\rangle}^{-1} = \frac{\tilde{A}_{\langle k\rangle}}{A_{\langle k\rangle} \tilde{A}_{\langle k\rangle}} \tag{2.76}$$

The geometric division thus also extends to blades.

$$\frac{A_{\langle k\rangle}}{B_{\langle l\rangle}} \equiv A_{\langle k\rangle} B_{\langle l\rangle}^{-1} = \frac{A_{\langle k\rangle} \tilde{B}_{\langle l\rangle}}{B_{\langle l\rangle} \tilde{B}_{\langle l\rangle}} \tag{2.77}$$

2.6 Conclusions

There are many more features to be found in GA than we have presented here. However, we have derived the most fundamental parts of GA. Developing GA further from where we have stopped would be identical to other works (e.g. [30, 41]).

We have shown here how GA can be developed from a vector space on which we define a geometric product. In particular, we have defined what we mean by vector in an algebraic sense, rather than starting of with n-tuples in $\mathcal{R}^n$ on which we define a scalar product. The geometric product is also given in its most general form and we only show later that it becomes the sum of the inner and outer product in the case of vectors.

The commutator and anti-commutator identites (2.15) to (2.18) are also shown to be of great use in finding vector identites. In particular, they are helpful in deriving the formula for the distributivity of the inner product with respect to the outer product (theorem 2.5.1). Finding this expansion is much harder work in [30]. Some further vector identities were also presented (equations (2.64)) that are very useful when working with bivectors.

We believe we have shown here an interesting and novel way to construct GA which does not obscure its simple fundamental structure.

Chapter 3

Projective Geometry

'The sky is blue,' he said, 'the grass is green.' Looking up, he saw that, on the contrary, the sky is like the veils which a thousand Madonnas have let fall from their hair; and the grass fleets and darkens like a flight of girls fleeing the embraces of hairy satyrs from enchanted woods.

"Orlando", Virginia Woolf.

The structure of this chapter is as follows. First we give an intuitive motivation for projective geometry. Then we look at projective geometry in more detail and derive concepts and formulae which will be needed in later chapters.

Our approach to projective, affine and Euclidean geometry is from an applied point of view, because we will use projective geometry chiefly to describe lines and planes, and find their intersections. Very good general introductions to projective geometry can be found in [2, 58]. A more purely mathematical approach is taken in [1].

Our tool to describe geometry is GA. Nevertheless, our approach differs somewhat from that of Hestenes and Ziegler [31], in that we embed Euclidean space ($\mathcal{E}^3$) into projective space ($\mathcal{P}^3$) in a different way. Hestenes introduced the *projective split* which transforms a vector that lies in projective space into a bivector. This bivector is then taken to lie in Euclidean space. In order to make this work the orthonormal basis for $\mathcal{P}^3$ needs to have a special metric [49, 51].

Historically, Grassmann introduced his exterior algebra to describe geometric objects like points, lines and planes, in projective space. Exterior algebra is still widely used in Physics (see e.g. [20, 8]). An extension of Grassmann algebra, called Grassmann–Cayley or double algebra is used in Computer Vision [16, 18] and Robotics [69, 70]. To see the differences between Grassmann algebra, Grassmann–Cayley algebra and geometric algebra, we will give a short introduction to the former two.

Let $\mathcal{V}^n$ be an n-dimensional vector space over $\mathcal{R}$. The p^{th} exterior power of $\mathcal{V}^n$, denoted $\Lambda^p\mathcal{V}^n$, is a real vector space, whose elements are referred to as "p–vectors", where $p = 0, 1, \ldots, n$. We have $\Lambda^0\mathcal{V}^n = \mathcal{R}$ and $\Lambda^1\mathcal{V}^n = \mathcal{V}^n$. In general $\Lambda^p\mathcal{V}^n$ is the set of formal sums

$$\sum_\mu \alpha_\mu \, (x_{\mu_1}\wedge\ldots\wedge x_{\mu_p}), \tag{3.1}$$

where $\mu = \{\mu_1, \ldots, \mu_p\}$ and the $\{x_i\} \in \mathcal{V}^n$. The "generators" $\{x_{\mu_1}\wedge\ldots\wedge x_{\mu_p}\}$ are called *extensors of step p*. The exterior product has the same properties as the outer product in GA: associativity, distributivity and for $x, y \in \mathcal{V}^n$, $x\wedge x = 0$, $x\wedge y = -y\wedge x$.

There is also an inner product defined on $\Lambda^p\mathcal{V}^n$ as a bilinear mapping from $\Lambda^p\mathcal{V}^n \times \Lambda^p\mathcal{V}^n$ to $\mathcal{R}$. This is the same inner product as defined in GA. However, in Grassmann algebra there is no general bilinear mapping from $\Lambda^p\mathcal{V}^n \times \Lambda^q\mathcal{V}^n$ to $\Lambda^{|p-q|}\mathcal{V}^n$ as in GA.

GC algebra is basically Grassmann algebra together with a "meet" operation. The meet operation is dual to the exterior product. Note that the exterior product is also called "join" in GC algebra[1]. Duality here means that if we interchange $\mathcal{V}^n$ and its dual space $\mathcal{V}^{n\,*}$, then we also interchange join and meet.

The meet operation in GC algebra is only defined for two extensors A, B if step(A) + step$(B) \geq n$. In this case it has the same meaning as the meet in GA. However, GA offers a more general definition of the meet through its particular definition of a join. Nevertheless, in GC algebra the meet has much of the functionality of the inner product in GA. This is not too surprising since in GA the meet is defined through the inner product.

In conclusion we can say that GC algebra offers much of the same functionality as GA. However, operations like the meet, dual and the particular inner product of Grassmann algebra can all be expressed as (GA–)inner product operations in geometric algebra.

3.1 Why Projective Geometry?

Three dimensional Euclidean space ($\mathcal{E}^3$) is the space in which the largest part of classical Physics is set, and standard vector algebra is the tool used to work

[1] The "join" in GC algebra is not to be confused with the join in GA.

in this space. In the following chapters we want to analyse static camera configurations, which means that we also have to work in $\mathcal{E}^3$. One operation we need to perform quite often is to find the intersection between lines and planes or two planes, and such an operation is not defined directly in Euclidean space.

This is due to the fact that a vector in $\mathcal{E}^3$ has a double meaning: it can be interpreted to define a point in space but also a *direction*. If we interpret a vector as a direction, it defines a line that passes through the origin. The points on this line can simply be found by scaling the vector.

This might not seem to be a big problem until we realise that points and lines are really different things. A point defines a zero dimensional space, whereas a line defines a one dimensional space. However, in Euclidean space a vector is supposed to define both at the same time. As we know from Physics this problem can be circumvented quite successfully.

Still, in certain areas it shows its teeth. The intersection point of a line and a plane, for example, is simply the intersection of the linear spaces each define. Therefore, we should be able to find an intersection point directly if we could express a line as a one dimensional space and a plane as a two dimensional space, and then intersect the two spaces. This can be done in projective space, whereas in Euclidean space intersection calculations are not so simple.

We cannot use a vector in $\mathcal{E}^3$ to define a point, because a vector can always be scaled and thus represents a line through the origin. However, we can add a "dummy" dimension to $\mathcal{E}^3$ which keeps track of the scale of a vector. This extended space is called *projective space* $\mathcal{P}^3$, and the added "dummy" dimension will be called *projective dimension.* We can return from $\mathcal{P}^3$ to $\mathcal{E}^3$ simply by dividing through by the respective scale. A vector in $\mathcal{P}^3$ still defines a line in $\mathcal{P}^3$ through the origin. However, it *represents* a point in $\mathcal{E}^3$, since it always projects to the same vector in $\mathcal{E}^3$ independent of its scale.

Implementing Projective Space. Let $\{e_1, e_2, e_3\}$ be an orthonormal frame of $\mathcal{E}^3$ with signature $\{+++\}$, and let $\{e^1, e^2, e^3\}$ be the corresponding reciprocal frame. In general, the reciprocal frame $\{A^i\}$ of an arbitrary frame $\{A_i\}$ is defined through

$$A_i \cdot A^j = \delta_i^j \tag{3.2}$$

where δ_i^j is the Kronecker delta. A reciprocal frame can also be expressed in terms of the frame vectors, as will be shown later. In the following we will use latin indices to count from 1 to 3, and greek indices to count from 1 to 4. Also lower case letters in bold face are vectors in $\mathcal{E}^3$ and upper case letters vectors in $\mathcal{P}^3$.

Let $\boldsymbol{x} = \alpha^i e_i$ be a vector in $\mathcal{E}^3$. Then

$$\boldsymbol{x} \xrightarrow{\mathcal{P}^3} X = \alpha^i\, e_i + e_4 = \alpha^\mu\, e_\mu, \tag{3.3}$$

where $\alpha^4 = 1$ and $\xrightarrow{\mathcal{P}^3}$ means embedding into $\mathcal{P}^3$. We use here the Einstein summation convention, i.e. $\alpha^\mu e_\mu \equiv \sum_\mu \alpha^\mu e_\mu$. The $\{e_1, e_2, e_3, e_4\}$ form an orthonormal basis of $\mathcal{P}^3$ with signature $\{+ + ++\}$. X can be projected back into $\mathcal{E}^3$ via

$$X \xrightarrow{\mathcal{E}^3} \boldsymbol{x} = \frac{X \cdot e^i}{X \cdot e^4}\, e_i \tag{3.4}$$

In this way an overall scale of X is cancelled out. Therefore, scaling a vector in $\mathcal{P}^3$ does not change its projection into $\mathcal{E}^3$. Thus we have achieved what we set out to do: a vector in $\mathcal{P}^3$ represents a zero dimensional space, i.e. a point, in $\mathcal{E}^3$.

Equation (3.4) is only valid though, as long as $X \cdot e^4 \neq 0$. However, we can turn this to our advantage. Vectors in $\mathcal{P}^3$ for which $X \cdot e^4 = 0$ are called *points at infinity*, because they would project to infinity in $\mathcal{E}^3$. They are also called *directions*, because they do not correspond to a real point in $\mathcal{E}^3$. That is, in $\mathcal{P}^3$ we can distinguish between direction and position vectors.

This will turn out to have many advantages. One of them is that the intersection of two parallel lines, which is a point at infinity, is just a normal point in $\mathcal{P}^3$, albeit one which projects to infinity in $\mathcal{E}^3$.

Another bonus we get from working in $\mathcal{P}^3$ is that the origin of $\mathcal{E}^3$ is not a special point anymore. It is simply e_4. Therefore, vectors in $\mathcal{P}^3$ are also called *homogeneous* vectors.

Visualising Projective Space. Figure 3.1 visualises the projection from projective to Euclidean space for a two dimensional Euclidean space. In this case equation (3.4) becomes

$$X \xrightarrow{\mathcal{E}^2} \boldsymbol{x} = \frac{X \cdot e^i}{X \cdot e^3}\, e_i\, ; \quad i \in \{1, 2\}. \tag{3.5}$$

That is, $\mathcal{E}^2$ is a plane embedded in $\mathcal{P}^2$. When we work in $\mathcal{P}^2$ we use the whole 3D-space. Projecting back into $\mathcal{E}^2$ means projecting points or lines onto the embedded Euclidean plane, as indicated in figure 3.1. If we then only want to work in $\mathcal{E}^2$ we can neglect the projective dimension, i.e. e_3 in this example.

This is formalised in equation (3.5). First we divide X through by $X \cdot e^3$, which projects X onto the Euclidean plane, and then we neglect the projective dimension.

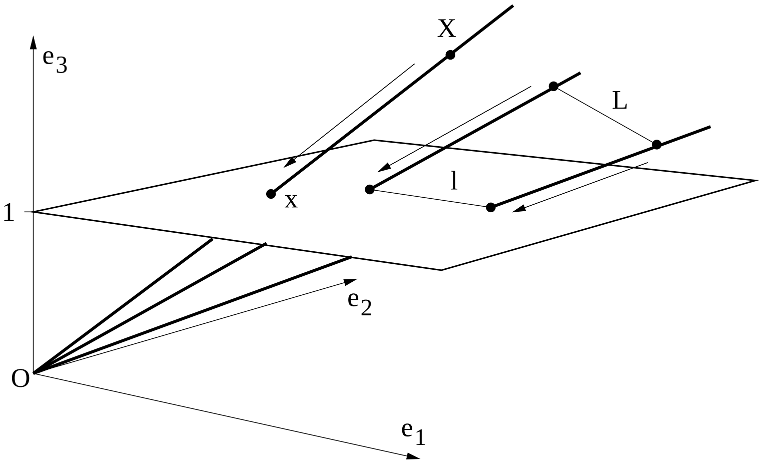


Figure 3.1: Projection of a point X and a line L from $\mathcal{P}^2$ into $\mathcal{E}^2$.

Why GA?. So far everything we have discussed can be expressed with standard vector algebra. So why GA? Well, points are easy enough to express, but lines or planes are a different matter.

As was shown above, a vector in $\mathcal{P}^3$ defines a 1D-space, but represents a 0D-space in $\mathcal{E}^3$. As will be shown later, a 2D-space in $\mathcal{P}^3$ represents a 1D-space, i.e. a line, in $\mathcal{E}^3$. Also a 3D-space in $\mathcal{P}^3$ represents a 2D-space, i.e. a plane, in $\mathcal{E}^3$. In other words, in $\mathcal{P}^3$ the dimensionality of all geometric entities (points, lines, planes) is increased by one. This is necessary because zero-dimensional spaces cannot be expressed directly algebraically.

In matrix algebra, points are represented by vectors, and lines and planes through the appropriate Plücker coordinates. Intersections can then be calculated directly. GA offers a coordinate free way to express points, lines and planes through the use of the *outer product*. The outer product combines two linear spaces in just the right way for our purposes, as will be shown later. That is, the outer product of two vectors in $\mathcal{P}^3$ defines a 2D-space which represents a line in $\mathcal{E}^3$. Similarly, the outer product of three vectors in $\mathcal{P}^3$ defines a 3D-space which represents a plane in $\mathcal{E}^3$. Writing these outer products in coordinate form, gives again the appropriate Plücker coordinates. Intersections between lines and planes can then be found with the "meet" operation, which will be defined later.

Note that GA is not the only algebra which can be used to express points, lines and planes in this way. For example, Grassmann–Cayley algebra is used for this purpose in [14, 17, 47].

3.2 Fundamentals

When we work with $\mathcal{P}^3$ we have to decide once and for all on a standard orthonormal basis. This *standard frame* of $\mathcal{P}^3$ is here defined to be the orthonormal set $\{e_1, e_2, e_3, e_4\}$. That is[2], $e_\mu \cdot e_\nu = \delta_{\mu\nu}$. Which orthonormal basis we choose is arbitrary, but we have to choose one to define the projective dimension, e_4 in this case, uniquely.

The *standard pseudoscalar* of $\mathcal{P}^3$ is defined as,

$$I \equiv e_1 \wedge e_2 \wedge e_3 \wedge e_4. \tag{3.6}$$

The inverse pseudoscalar I^{-1} is defined such that $II^{-1} = 1$. Hence,

$$I^{-1} \equiv e_4 \wedge e_3 \wedge e_2 \wedge e_1 \tag{3.7}$$

From the definition of the standard pseudoscalar it follows that

$$II = I^{-1}I^{-1} = 1\,, \qquad I = I^{-1}. \tag{3.8}$$

A vector in $\mathcal{P}^3$ will be called a *homogeneous* vector[3]. As before, homogeneous vectors in $\mathcal{P}^3$ will be written as capital letters, and their corresponding 3D-vectors in $\mathcal{E}^3$ as lower case letters in bold face.

Recall that an overall scalar factor of a homogeneous vector A cancels when A is projected down to $\mathcal{E}^3$. Since we are ultimately only interested in $\mathcal{E}^3$ 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.

A set $\{A_\mu\}$ of four homogeneous vectors forms a basis or *frame* of $\mathcal{P}^3$ 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

$$I_a = A_1 \wedge A_2 \wedge A_3 \wedge A_4 \tag{3.9}$$

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

$$I_a = \rho_a I \tag{3.10}$$

[2] Recall that we use greek indices when counting from 1 to 4, and latin indices when counting from 1 to 3.

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

where ρ_a is the *scale* of the A-frame. If $A_i = \alpha_i^j\, e_j$, then ρ_a is given by

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

The inverses of these two pseudoscalars are related by

$$I_a^{-1} = \rho_a^{-1} I^{-1} \tag{3.11}$$

From equations (3.8), (3.10) and (3.11) it follows that

$$I_a^{-1} = \rho_a^{-2} I_a \tag{3.12}$$

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 (2.68) the following important result follows;

$$\begin{aligned} A_\mu I_a &= A_\mu \cdot (A_1 \wedge A_2 \wedge A_3 \wedge A_4) \\ &= \sum_{\nu_1=1}^{4} (A_\mu \cdot A_{\nu_1})(A_{\nu_2} \wedge A_{\nu_3} \wedge A_{\nu_4}). \end{aligned} \tag{3.13}$$

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.* To a certain extent it is related to the bracket notation as used in Grassmann–Cayley algebra and GA [37]. There the bracket of a pseudoscalar P, say, is a scalar, defined as the dual of P in GA. That is, $[P] = P I^{-1}$; here however the dual bracket concept can produce something other than a scalar.

The dual bracket is defined as

$$[\![A_{\mu_1} A_{\mu_2} \cdots A_{\mu_n}]\!] \equiv (A_{\mu_1} \wedge A_{\mu_2} \wedge \ldots \wedge A_{\mu_n}) I^{-1} \tag{3.14a}$$

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

A very useful identity is the following.

$$\begin{aligned} [\![A_{\mu_1} A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot I^{-1} \\ &= A_{\mu_1} \cdot \Big((A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot I^{-1} \Big) \\ &= A_{\mu_1} \cdot [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] \end{aligned} \tag{3.15}$$

Similarly it may be shown that

$$\begin{aligned} [\![A_{\mu_1} A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] &= (A_{\mu_1} \wedge A_{\mu_2}) \cdot [\![A_{\mu_3} A_{\mu_4}]\!] \\ &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3}) \cdot [\![A_{\mu_4}]\!] \\ &= (A_{\mu_1} \wedge A_{\mu_2} \wedge A_{\mu_3} \wedge A_{\mu_4}) \cdot [\![1]\!] \end{aligned} \tag{3.16}$$

Note that $[\![1]\!] = I^{-1}$. Put simply, vectors may be "pulled" out of a dual bracket by taking the inner product of them with the remainder of the bracket.

3.3 Points, Lines and Planes

In this section we will show that points, lines and planes in $\mathcal{E}^3$ are represented by lines, planes and 3D-hyperplanes in $\mathcal{P}^3$, respectively. Recall that this increase in dimensionality is necessary to distinguish points and lines, because zero dimensional spaces, i.e. points, cannot be expressed directly algebraically. In the following it is therefore important to distinguish between the meaning of an algebraic object in $\mathcal{P}^3$, and what it represents in $\mathcal{E}^3$.

First we have to give a definition which relates projective and Euclidean space. The rest of this section will then investigate the effects this relation has.

Definition 3.3.1 *Euclidean space $\mathcal{E}^n$ is a subspace of projective space $\mathcal{P}^n$. Projective space extends the orthonormal basis of $\mathcal{E}^n$ $\{e_1, e_2, \ldots, e_n\}$ by one basis vector e_{n+1} such that $\{e_1, e_2, \ldots, e_n, e_{n+1}\}$ is also an orthonormal set. However, $\mathcal{P}^n$ does not contain the origin of this $(n+1)$–dimensional space. A vector $\boldsymbol{a} \in \mathcal{E}^n$ is said to be* embedded *in $\mathcal{P}^n$ through the operation*

$$\boldsymbol{a} \xrightarrow{\mathcal{P}^n} A = \boldsymbol{a} + e_{n+1}.$$

A vector $A \in \mathcal{P}^n$ is said to be projected into $\mathcal{E}^n$ *through the operation*

$$A \xrightarrow{\mathcal{E}^n} \boldsymbol{a} = \frac{A \cdot e^i}{A \cdot e^{n+1}}\, e_i \; ; \quad i \in \{1, 2, \dots, n\}.$$

If $A{\cdot}e^{n+1} = 0$, then A is said to lie on the plane at infinity in $\mathcal{P}^n$, and it projects to inifinity in $\mathcal{E}^n$.

3.3.1 Points

Definition 3.3.2 *Projective line. Let $A \in \mathcal{P}^3$, then the projective line represented by A is given by the set of points $\{X = \tau_a A \,:\, \tau_a \in \mathcal{R}\backslash\{0\}\}$, where $\mathcal{R}\backslash\{0\}$ denotes the set of real numbers without zero.*

Therefore, any homogeneous vector in $\mathcal{P}^3$ represents a projective line, and if two homogeneous vectors are linearly dependent, they represent the same projective line.

Lemma 3.3.1 *Let $A \in \mathcal{P}^3$, then a vector $X \in \mathcal{P}^3$ lies on the projective line represented by A if and only if it satisfies*

$$X \wedge A = 0$$

Proof. We have to show that $X \wedge A = 0$ iff $X = \tau_a A$, where $\tau_a \in \mathcal{R}\backslash\{0\}$. This follows directly from lemma 2.5.1 on page 33. ■

Theorem 3.3.1 *Every projective line in $\mathcal{P}^3$ given by some $A \in \mathcal{P}^3$ which does not lie on the plane at infinity, represents a unique point in $\mathcal{E}^3$ and vice versa.*

Proof. Let $A \in \mathcal{P}^3$ represent a projective line and A does not lie on the plane at infinity. The set of points that lie on this projective line is $\{X = \tau_a A \,:\, \tau_a \in \mathcal{R}\backslash\{0\}\}$. First we have to show that the $\{X\}$ project to the same point in $\mathcal{E}^3$ as A. Let $A \xrightarrow{\mathcal{E}^3} \boldsymbol{a}$, then

$$X \xrightarrow{\mathcal{E}^3} \boldsymbol{x} = \frac{X \cdot e^i}{X \cdot e^4}\, e_i = \frac{\tau_a\,(A \cdot e^i)}{\tau_a\,(A \cdot e^4)}\, e_i = \frac{A \cdot e^i}{A \cdot e^4}\, e_i = \boldsymbol{a} \tag{3.17}$$

This shows that all points on a projective line project to the same point in $\mathcal{E}^3$. It is also clear that this relation is unique. That the reverse is also true follows directly from the definition 3.3.1. ■

This theorem shows that there is a one to one correspondence between points in $\mathcal{E}^3$ and projective lines in $\mathcal{P}^3$. Therefore, we will call vectors in $\mathcal{P}^3$ "points".

There is also a dual representation of points in $\mathcal{P}^3$. From theorem 3.3.1 we know that all homogeneous vectors $X \in \mathcal{P}^3$ which represent the same point in $\mathcal{E}^3$ as some $A \in \mathcal{P}^3$ are given by

$$\begin{aligned} & X \wedge A = 0 \\ \iff \quad & [\![XA]\!] = 0 \\ \iff \quad & X \cdot [\![A]\!] = 0 \\ \iff \quad & X \cdot N_a = 0 \end{aligned} \tag{3.18}$$

where $N_a \equiv [\![A]\!]$ is called the *normal* of A. Therefore, a point can be given in two different ways: either directly (e.g. A) or as its normal (e.g. N_a). Thus, there are also two ways to define the set of vectors $X \in \mathcal{P}^3$ that represent the point $\boldsymbol{a} \in \mathcal{E}^3$.

$$X \wedge A = X \cdot N_a = 0. \tag{3.19}$$

3.3.2 Lines

In the following we show that the outer product of two homogeneous vectors, also called a *bivector*, defines a projective plane in $\mathcal{P}^3$, which represents a line in $\mathcal{E}^3$.

Definition 3.3.3 *Projective plane. Let $A, B \in \mathcal{P}^3$ be linearly independent. Then the projective plane represented by A and B is given by the set of points $\{X = \tau_a A + \tau_b B \,:\, (\tau_a, \tau_b) \in \mathcal{R}^2 \backslash \{0\}\}$. That is, τ_a and τ_b cannot be zero simultaneously.*

Lemma 3.3.2 *Let $A, B \in \mathcal{P}^3$ represent a projective plane, then a vector $X \in \mathcal{P}^3$ lies on this projective plane if and only if it satisfies*

$$X \wedge (A \wedge B) = 0$$

Proof. We have to show that $X \wedge (A \wedge B) = 0$ iff $X = \tau_a A + \tau_b B$. The "only if" part is clearly true, since $(\tau_a A + \tau_b B) \wedge (A \wedge B) = 0$. To prove the "if" part we assume that $X = \tau_a A + \tau_b B + \tau_c C$ where $A, B, C \in \mathcal{P}^3$ are mutually linearly independent and $(\tau_a, \tau_b, \tau_c) \in \mathcal{R}^3 \backslash \{0\}$. Then

$$X \wedge (A \wedge B) = (\tau_c C) \wedge (A \wedge B) \neq 0 \tag{3.20}$$

Therefore, the most general expression for X to satisfy $X \wedge (A \wedge B) = 0$ is $X = \tau_a A + \tau_b B$. ■

Theorem 3.3.2 *Let $A, B \in \mathcal{P}^3$ be linearly independent and let at least one of $\{A, B\}$ not lie on the plane at infinity. Then the projective plane $A \wedge B$ represents a unique line in $\mathcal{E}^3$ and vice versa.*

Proof. Let $A, B \in \mathcal{P}^3$ represent a projective plane, whereby at least one of $\{A, B\}$ does not lie on the plane at infinity. The set of points that lie on this plane is $\{X = \tau_a A + \tau_b B \; : \; (\tau_a, \tau_b) \in \mathcal{R}^2 \backslash \{0\}\}$. First we have to show that the $\{X\}$ project to points on the same line in $\mathcal{E}^3$. Let $\boldsymbol{a} \xrightarrow{\mathcal{P}^3} A = \boldsymbol{a} + e_4$ and $\boldsymbol{b} \xrightarrow{\mathcal{P}^3} B = \boldsymbol{b} + e_4$. Projecting $X = \tau_a A + \tau_b B$ down to $\mathcal{E}^3$ gives

$$\begin{aligned} X \xrightarrow{\mathcal{E}^3} \boldsymbol{x} \;&=\; \frac{X \cdot e^i}{X \cdot e^4}\, e_i \\ &=\; \frac{\tau_a\, \boldsymbol{a} + \tau_b\, \boldsymbol{b}}{\tau_a + \tau_b} \\ &=\; \frac{\tau_a\, (\boldsymbol{a} - \boldsymbol{b}) + (\tau_a + \tau_b)\, \boldsymbol{b}}{\tau_a + \tau_b} \\ &=\; \tfrac{\tau_a}{\tau_a + \tau_b} (\boldsymbol{a} - \boldsymbol{b}) + \boldsymbol{b} \end{aligned} \tag{3.21}$$

That is, every point on the projective plane represented by A and B projects to the line passing through $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathcal{E}^3$. It is also quite clear that this mapping is unique and that the reverse is also true. However, this part of the proof will be omitted here for brevity. ■

Therefore, we have a one to one correspondence between lines in $\mathcal{E}^3$ and projective planes in $\mathcal{P}^3$. We will therefore call bivectors in $\mathcal{P}^3$ "lines".

There is also a dual representation of a line. Let $L_{ab} = A \wedge B$, then the points X that lie on L_{ab} are given by

$$\begin{aligned} & X \wedge L_{ab} = 0 \\ \iff\; & [\![X L_{ab}]\!] = 0 \\ \iff\; & X \cdot [\![L_{ab}]\!] = 0 \\ \iff\; & X \cdot N_{ab} = 0 \end{aligned} \tag{3.22}$$

where $N_{ab} \equiv [\![L_{ab}]\!]$ is called the *normal* of L_{ab}. To summarise, we can say that a line may be represented in two ways: either directly (e.g. $L_{ab} = A \wedge B$) or via its normal (e.g. $N_{ab} \equiv [\![L_{ab}]\!]$). The points X that lie on a line are then defined by

$$X \wedge L_{ab} = X \cdot N_{ab} = 0 \tag{3.23}$$

3.3.3 Planes

The last fundamental geometric entities we discuss are planes. We will show that 3D-hyperplanes in $\mathcal{P}^3$ are defined by the outer product of three homogeneous vectors, also called *trivectors*, and that these hyperplanes represent 2D-planes in $\mathcal{E}^3$.

Definition 3.3.4 *Projective hyperplane. Let $A, B, C \in \mathcal{P}^3$ be linearly independent. Then the projective hyperplane represented by A, B and C is given by the set of points $\{X = \tau_a A + \tau_b B + \tau_c C \,:\, (\tau_a, \tau_b, \tau_c) \in \mathcal{R}^3 \backslash \{0\}\}$. That is, τ_a, τ_b and τ_c cannot be zero simultaneously.*

Lemma 3.3.3 *Let $A, B, C \in \mathcal{P}^3$ represent a projective hyperplane, then a vector $X \in \mathcal{P}^3$ lies on the projective hyperplane passing through A, B and C if and only if it satisfies*

$$X \wedge (A \wedge B \wedge C) = 0$$

Proof. We have to show that $X \wedge (A \wedge B \wedge C) = 0$ iff $X = \tau_a A + \tau_b B + \tau_c C$. The proof is analogous to the one given for lines (theorem 3.20) and will be omitted here for brevity. ■

Theorem 3.3.3 *Let $A, B, C \in \mathcal{P}^3$ be linearly independent and let at least one of $\{A, B, C\}$ not lie on the plane at infinity. Then the projective hyperplane $A \wedge B \wedge C$ represents a unique plane in $\mathcal{E}^3$ and vice versa.*

Proof. The proof is analogous to the one given for lines (theorem 3.3.2) and will be omitted here for brevity. ■

Again we see that there is a one to one correspondence between planes in $\mathcal{E}^3$ and projective hyperplanes in $\mathcal{P}^3$. We will therefore call trivectors in $\mathcal{P}^3$ "planes".

As for points and lines, there is a dual representation of planes. For example, the dual representation of plane $P_{abc} = A \wedge B \wedge C$ is given by $N_{abc} = [\![A \wedge B \wedge C]\!]$. Then those vectors X that lie on plane P_{abc} satisfy

$$X \wedge P_{abc} = X \cdot N_{abc} = 0 \tag{3.24}$$

3.4 Intersections

We will now disuss how intersections between lines and planes can be calculated in projective space using GA. To see how this may be done, we will start in $\mathcal{P}^2$. We know that in the corresponding 2D-Space $\mathcal{E}^2$, two lines intersect in a point if they are not parallel. The basis of $\mathcal{P}^2$ is defined as $\{e_1, e_2, e_3\}$ with signature $\{+++\}$.

3.4.1 Intersection of Lines in $\mathcal{P}^2$

Theorem 3.4.1 *Let $A, B, C, D \in \mathcal{P}^2$ and let $L_{ab} = A \wedge B$ and $L_{cd} = C \wedge D$, with dual representations $N_{ab} = [\![L_{ab}]\!]$ and $N_{cd} = [\![L_{cd}]\!]$. Then the intersection point X of lines L_{ab} and L_{cd} is given by*

$$X = N_{ab} \cdot L_{cd} = L_{ab} \cdot N_{cd}$$

Proof. If we can show that X lies on line L_{ab} *and* on line L_{cd}, then X has to be the intersection point of the two lines. Note that since we work in $\mathcal{P}^2$, $N_{ab} = [\![L_{ab}]\!]$ and $N_{cd} = [\![L_{cd}]\!]$ are vectors.

$$\begin{aligned} N_{ab} \cdot X &= N_{ab} \cdot (N_{ab} \cdot L_{cd}) \\ &= (N_{ab} \wedge N_{ab}) \cdot L_{cd} \\ &= 0 \end{aligned} \tag{3.25}$$

Therefore, X lies on line L_{ab}. Furthermore,

$$\begin{aligned} X \wedge L_{cd} &= (N_{ab} \cdot L_{cd}) \wedge L_{cd} \\ &= (N_{ab} \cdot C)\, D \wedge L_{cd} - (N_{ab} \cdot D)\, C \wedge L_{cd} \\ &= 0 \end{aligned} \tag{3.26}$$

Thus, X also lies on line L_{cd}. ■

3.4.2 Intersection of Parallel Lines in $\mathcal{P}^2$

For interest consider two parallel lines $L_{ab} = A\wedge B$ and $L_{cd} = C\wedge D$. First we will show that the direction of a line $L \in \mathcal{P}^2$ is given by $V \simeq L\cdot e^3$. Let $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{E}^2$ and $A = \tau_a(\boldsymbol{a} + e_3)$, $B = \tau_b(\boldsymbol{b} + e_3)$. Then $L_{ab} = A\wedge B$ represents the line through $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathcal{E}^2$. The direction of this line is therefore given by $V_{ab} = \boldsymbol{a} - \boldsymbol{b}$.

$$\begin{aligned}
L_{ab}\cdot e^3 &= A\,(B\cdot e^3) - B\,(A\cdot e^3)\\
&\simeq \frac{A}{A\cdot e^3} - \frac{B}{B\cdot e^3}\\
&= \frac{\tau_a\,(\boldsymbol{a} + e_3)}{\tau_a} - \frac{\tau_b\,(\boldsymbol{b} + e_3)}{\tau_b}\\
&= V_{ab}.
\end{aligned}\tag{3.27}$$

V_{ab} is indeed a direction because $A/(A\cdot e^3)$ and $B/(B\cdot e^3)$ have both a unit e_3 component, and thus $V_{ab}\cdot e^3 = 0$. L_{ab} can now also be written as

$$L_{ab} \simeq A\wedge V_{ab} \simeq B\wedge V_{ab}. \tag{3.28}$$

That is, V_{ab} can also be regarded as a point lying on line L_{ab}. In fact, it is the intersection point of line L_{ab} with the *plane at infinity* (P_∞). The plane at infinity contains all direction vectors.

If lines L_{ab} and L_{cd} are parallel, they point in the same direction. Therefore, we can also write $L_{cd} \simeq C\wedge V_{ab}$. The intersection point X_p of lines L_{ab} and L_{cd} is therefore given by

$$\begin{aligned}
X_p &= N_{ab}\cdot L_{cd}\\
&= [\![AB]\!]\cdot(C\wedge V_{ab})\\
&= \quad [\![ABC]\!]\,\frac{A}{A\cdot e^3} - \underbrace{[\![ABA]\!]}_{=0}\,\frac{C}{A\cdot e^3}\\
&\quad - \ [\![ABC]\!]\,\frac{B}{B\cdot e^3} - \underbrace{[\![ABB]\!]}_{=0}\,\frac{C}{B\cdot e^3}\\
&= [\![ABC]\!]\left(\frac{A}{A\cdot e^3} - \frac{B}{B\cdot e^3}\right)\\
&\simeq V_{ab}.
\end{aligned}\tag{3.29}$$

Note that because we work in $\mathcal{P}^2$, $[\![ABC]\!]$ is a scalar. This equation shows that two parallel lines intersect in P_∞, their intersection point giving their direction. Now we return to $\mathcal{P}^3$, which gives us the added complexity of planes.

3.4.3 Intersections with Planes in $\mathcal{P}^3$

Theorem 3.4.2 *Let $P_a, P_b, L \in \mathcal{P}^3$ with $P_a = A_1 \wedge A_2 \wedge A_3$, $P_b = B_1 \wedge B_2 \wedge B_3$ and $L = C_1 \wedge C_2$. Then the intersection line L_{ab} of planes P_a and P_b is given by*

$$L_{ab} = P_a \cdot [\![P_b]\!] = [\![P_a]\!] \cdot P_b$$

Furthermore, the intersection point X between plane P_a and line L is

$$X = L \cdot [\![P_a]\!] = [\![L]\!] \cdot P_a$$

Proof. The proof may be found in a similar way as for theorem 3.4.1. ■

3.4.4 Intersection of Lines in $\mathcal{P}^3$

The intersection of two lines in $\mathcal{P}^3$ cannot be found in an analogous way. This is due to the fact that in $\mathcal{P}^3$ two lines might not intersect at all. From an algebraic point of view the dual of a line also gives a bivector. Therefore, the inner product of a line with the dual of another gives a scalar and not a point as needed. In order to use the formula for the intersection of two lines as given in $\mathcal{P}^2$, we would have to know the plane in which the two lines lie, if they are co-planar[4] that is. An easier way to find the intersection point between two lines in $\mathcal{P}^3$ is through the following construction.

Let $L_{ab}, L_{cd} \in \mathcal{P}^3$ be two co-planar lines. As in the previous section it can be shown that the direction of line L_{ab} is given by $V_{ab} \simeq L_{ab} \cdot e^4$. Recall that V_{ab} is actually a point that lies on the line L_{ab}. Therefore, $L_{cd} \wedge V_{ab}$ gives the plane in which L_{ab} and L_{cd} lie. Of course, if L_{cd} and L_{ab} are parallel then this expression gives zero. Now we can find the intersection point between lines L_{ab} and L_{cd} simply by intersecting line L_{ab} with plane $L_{cd} \wedge V_{ab}$.

3.5 Reciprocal Vector Frames

In equation (3.2) we defined the reciprocal frame $\{A^\mu\}$ of a frame $\{A_\mu\}$ through

$$A_\mu \cdot A^\nu = \delta^\nu_\mu \tag{3.30}$$

[4] If two lines are not co-planar they also do not intersect.

It is now straightforward to give reciprocal frame vectors in terms of their respective frame vectors. From equation (3.15) it follows that

$$\begin{aligned} & [\![A_1 A_2 A_3 A_4]\!] = \rho_a \\ \iff \quad & A_{\mu_1} \cdot [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] = \rho_a \\ \iff \quad & A_{\mu_1} \cdot A^{\mu_1} = 1 \end{aligned} \tag{3.31}$$

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

$$A^{\mu_1} = \rho_a^{-1} \, [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!] \tag{3.32}$$

From these definitions of reciprocal frame vectors it follows directly that they satisfy the condition from equation (3.30). Equation (3.32) shows that a reciprocal frame vector is nothing other than the dual of a plane. It may therefore also be regarded as the *normal* of the plane that is its dual.

In Grassman-Cayley (GC) algebra these reciprocal vectors would be defined as elements of a *dual space*, which is indeed what is done in [14]. However, to the knowledge of the authors reciprocal frames have not been used to describe projections in GC algebra, as will be done here.

A reciprocal frame can be used to transform a vector from one frame into another. For example, let $X = \alpha^{\mu} A_{\mu}$. Then X can be written in another frame $\{B_{\mu}\}$ with reciprocal frame $\{B^{\mu}\}$ as

$$X = (\, X \cdot B^{\nu}\,) B_{\nu} = \alpha^{\mu} \, (\, A_{\mu} \cdot B^{\nu}\,) \, B_{\nu} = \beta^{\nu} \, B_{\nu} \tag{3.33}$$

where $A_{\mu} \cdot B^{\nu}$ is the matrix giving the linear mapping between frames A and B, and $\beta^{\nu} = \alpha^{\mu} \, A_{\mu} \cdot B^{\nu}$.

Note that the $\{A^{\mu}\}$ also form a basis of $\mathcal{P}^3$ since $A^1 \wedge A^2 \wedge A^3 \wedge A^4 \neq 0$. Therefore, X can also be given as $X = \alpha_{\mu} A^{\mu}$, where the $\{\alpha_{\mu}\}$ differ from the $\{\alpha^{\mu}\}$.

3.6 Line Frames

It will be important later not only to consider vector frames but also line frames. A line frame consists of the set of unique bivectors that can be created from a corresponding vector frame. In $\mathcal{P}^3$ at most 6 unique bivectors can be formed. Hence, a full line frame has 6 components. We define a full line frame $\{L_a^\mu\}$ based on the vector frame $\{A_\mu\}$ as

$$L_a^\mu \equiv \{A_2 \wedge A_3,\ A_3 \wedge A_1,\ A_1 \wedge A_2,\ A_2 \wedge A_4,\ A_4 \wedge A_1,\ A_3 \wedge A_4\}. \tag{3.34}$$

Note that greek indices used with line frames are assumed to count from 1 to 6. Latin indices, however, still count from 1 to 3. Later on we will need to use planar line frames, i.e. line frames that define a plane. We will see that the planar line frames we need are just of the form $\{L_a^i\}$, i.e. the first three components of a full line frame. Therefore, we can write the A-line frame $\{L_a^i\}$ as $L_a^{i_1} = A_{i_2} \wedge A_{i_3}$, where 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 from equation (3.16)

$$\begin{aligned} & [\![A_1 A_2 A_3 A_4]\!] = \rho_a \\ \iff \quad & (A_{\mu_1} \wedge A_{\mu_2}) \cdot [\![A_{\mu_3} A_{\mu_4}]\!] = \rho_a \\ \iff \quad & L_a^\nu \cdot L_\nu^a = 1 \end{aligned} \tag{3.35}$$

with no implicit summation over ν. This[5] defines the *reciprocal A-line frame* $\{L_\nu^a\}$ as

$$\begin{aligned} L_\nu^a = \rho_a^{-1} \Big\{ & [\![A_1 A_4]\!], [\![A_2 A_4]\!], [\![A_3 A_4]\!], \\ & [\![A_3 A_1]\!], [\![A_2 A_3]\!], [\![A_1 A_2]\!] \Big\} \end{aligned} \tag{3.36}$$

Hence,

$$L_a^\mu \cdot L_\nu^a = \delta_\nu^\mu \tag{3.37}$$

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

3.7 Plane Frames

Although we do not need them later on, it is interesting to look at plane frames, if only for completeness. The plane frame based on the A-vector frame is defined as

$$P_\mu^a \equiv \{A_4 \wedge A_2 \wedge A_3,\ A_4 \wedge A_3 \wedge A_1,\ A_4 \wedge A_1 \wedge A_2,\ A_2 \wedge A_3 \wedge A_1\}. \tag{3.38}$$

[5] Note how similar this derivation is to that of reciprocal vector frames (equation (3.31)).

Note that with respect to plane frames, greek indices are again assumed to count in the range of 1 to 4. Following an analogous procedure as in equation (3.35) it is easy to show that the corresponding *reciprocal plane frame* $\{P_a^\mu\}$ is given by

$$P_a^\mu \equiv \rho_a^{-1} \Big\{ [\![A_1]\!],\, [\![A_2]\!],\, [\![A_3]\!],\, [\![A_4]\!] \Big\}. \tag{3.39}$$

Therefore,

$$P_\mu^a \cdot P_a^\nu = \delta_\mu^\nu \tag{3.40}$$

3.8 Determinants

We will restrict our description of determinants to determinants of 3×3 matrices, because that is what we will need later on. However, the formalism given in the following may easily be extended to incorporate any type of determinant.

The determinant of a 3×3 matrix can be written in terms of the ϵ_{ijk} operator, which is defined as

$$\epsilon_{ijk} = \begin{cases} +1 & \text{if the } \{ijk\} \text{ form an even permutation of } \{123\} \\ 0 & \text{if any two indices of } \{ijk\} \text{ are equal} \\ -1 & \text{if the } \{ijk\} \text{ form an odd permutation of } \{123\} \end{cases} \tag{3.41}$$

Let $\alpha_1^{i_a}$, $\alpha_2^{i_b}$ and $\alpha_3^{i_c}$ give the three rows of a 3×3 matrix M. Then the determinant of M is

$$\det(M) = \epsilon_{i_a i_b i_c} \alpha_1^{i_a} \alpha_2^{i_b} \alpha_3^{i_c} \tag{3.42}$$

Note that there is an implicit summation over all indices. To simplify the notation we define

$$\left.\begin{array}{r} \det(\alpha_1^{i_a}, \alpha_2^{i_b}, \alpha_3^{i_c})_{i_a i_b i_c} \\ \det(\alpha_j^i)_{ij} \end{array}\right\} \equiv \epsilon_{i_a i_b i_c} \alpha_1^{i_a} \alpha_2^{i_b} \alpha_3^{i_c} = \det(M) \tag{3.43}$$

Furthermore, if the rows of the matrix M are written as vectors $\boldsymbol{a}_j = \alpha_j^i e_i$, then we may also write

$$\left.\begin{array}{r} \det(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3) \\ |\boldsymbol{a}_1 \boldsymbol{a}_2 \boldsymbol{a}_3| \end{array}\right\} \equiv \det(M) \tag{3.44}$$

Let the $\{A_\mu\}$ form a frame of $\mathcal{P}^3$, with reciprocal frame $\{A^\mu\}$. Then from the definition of the dual bracket it follows that

$$\epsilon_{i_a i_b i_c} = \rho_a^{-1} \llbracket A_{i_a} A_{i_b} A_{i_c} A_4 \rrbracket \tag{3.45}$$

Therefore, we may express a determinant for example as

$$\det(\alpha^i_j)_{ij} = \alpha_1^{i_a} \alpha_2^{i_b} \alpha_3^{i_c}\ \rho_a^{-1} \llbracket A_{i_a} A_{i_b} A_{i_c} A_4 \rrbracket \tag{3.46}$$

3.9 Meet and Join

The *meet* defines a generalised intersection operation. It is usually defined in terms of the *join*. In general terms the join is the union and the meet is the intersection of two spaces. In GA any blade can be treated as a pseudoscalar of a particular subspace.

The join of two blades A and B, written as $A \dot{\wedge} B$ 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 $A \dot{\wedge} B = e_1 \wedge e_2 \wedge e_3$. Also the join of two co-planar lines is the plane in which they are contained.

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 $A \vee B \simeq e_2$.

Definition 3.9.1 *Let A and B be two arbitrary blades, and let $J = A \dot{\wedge} B$, then*

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

If we take the meet of two planes or a plane and a line in $\mathcal{P}^3$, 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 the meet between two planes or a plane and a line we can write

$$\begin{aligned} A \vee B &= \llbracket \llbracket A \rrbracket \llbracket B \rrbracket \rrbracket \\ &= \llbracket A \rrbracket \cdot \llbracket \llbracket B \rrbracket \rrbracket \quad \text{from equation (3.16)} \\ &= \llbracket A \rrbracket \cdot B \\ &= A^* \cdot B \end{aligned} \tag{3.47}$$

Comparing this result with theorems 3.4.1 and 3.4.2, we see that the meet operation can indeed be used to calculate the intersection between two geometric objects.

From equation (3.47) follows another identity which will become very useful later on.

$$\begin{aligned} [\![A]\!] \vee [\![B]\!] &= \Big[\!\!\Big[[\![[\![A]\!]]\!] [\![[\![B]\!]]\!] \Big]\!\!\Big] \\ &= [\![AB]\!] \end{aligned} \tag{3.48}$$

where A and B are some arbitrary object. More details about meet and join may be found in [31] and [37].

3.10 Cameras and Projections

A pinhole camera can be defined by 4 homogeneous vectors in $\mathcal{P}^3$: one vector gives the optical centre and the other three define the image plane [37, 39]. Thus, the vectors needed to define a pinhole camera also define a frame for $\mathcal{P}^3$. Conventionally the fourth vector of a frame, eg. A_4, defines the optical centre, and the outer product of the other three ($\{A_1, A_2, A_3\}$) 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 as shown in figure 3.2. 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 (3.16) and (3.47) by

$$\begin{aligned} (X \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) &= (A_1 \wedge A_2 \wedge A_3) \cdot \left[(X \wedge A_4) I^{-1} \right] \\ &= \quad [\![X A_2 A_3 A_4]\!] A_1 \\ &\quad + [\![X A_3 A_1 A_4]\!] A_2 \\ &\quad + [\![X A_1 A_2 A_4]\!] A_3. \end{aligned} \tag{3.49}$$

Recall that we defined $\{i_1, i_2, i_3\}$ as an even permutation of $\{1, 2, 3\}$, and that $[\![XABC]\!] = X \cdot [\![ABC]\!]$. Then equation (3.49) becomes

$$\begin{aligned} (X \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) &= \sum_{i_3} \left[X \cdot [\![A_{i_1} A_{i_2} A_4]\!] \right] A_{i_3} \\ &= \rho_a \, (\, X \cdot A^i \,) \, A_i. \end{aligned} \tag{3.50}$$

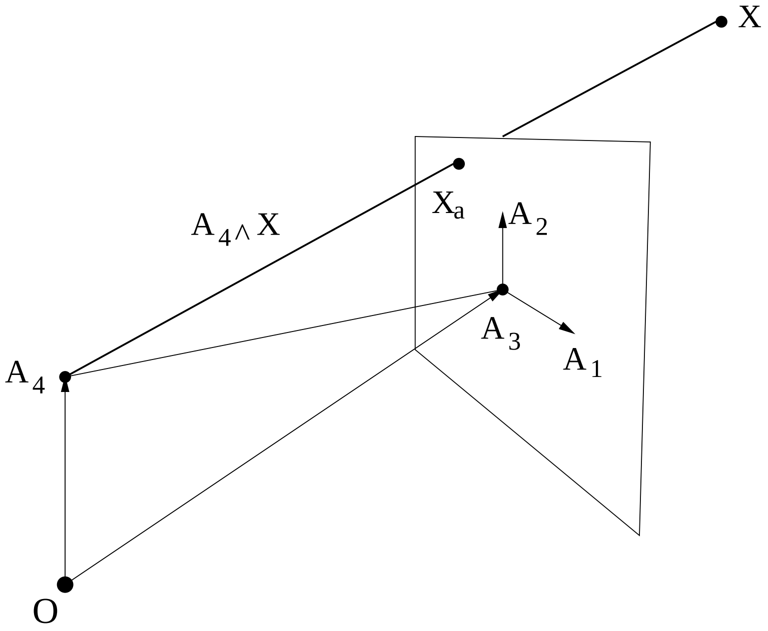


Figure 3.2: Example of the representation of a pinhole camera in $\mathcal{P}^3$. A_4 is the optical centre and $A_1 \wedge A_2 \wedge A_3$ defines the image plane.

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

$$\begin{aligned} X_a &\simeq (X \cdot A^i) A_i \\ &= (\zeta^\mu Z_\mu \cdot A^i) A_i \\ &= \zeta^\mu K^i_{\ \mu} A_i ; \qquad K^i_{\ \mu} \equiv Z_\mu \cdot A^i \end{aligned} \tag{3.51}$$

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

In [14] 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 duals *are* planes.

With the same method, lines can be projected onto an image plane. For example, let L be some line in $\mathcal{P}^3$. Then its projection onto the A-image plane is

$$(L \wedge A_4) \vee (A_1 \wedge A_2 \wedge A_3) \simeq (L \cdot L^a_i) L^i_a \tag{3.52}$$

3.11 Dual Representations of Lines and Points

Dual representations of lines and point will turn out to be quite useful analytically and also improve our understanding of the geometry. We will first derive the dual representation of the planar line basis $\{L_a^i\}$. Note that $I_a^{-1} = A^4 \wedge A^3 \wedge A^2 \wedge A^1$, since then $I_a I_a^{-1} = 1$. Also recall that $A_\mu \cdot A^\nu = \delta_\mu^\nu$.

$$\begin{aligned}
L_a^{i_1} &= A_{i_2} \wedge A_{i_3} \\
&= (A_{i_2} \wedge A_{i_3}) I_a^{-1} I_a \quad \text{since } I_a^{-1} I_a = 1 \\
&= \Big[(A_{i_2} \wedge A_{i_3}) \cdot (A^4 \wedge A^3 \wedge A^2 \wedge A^1)\Big] I_a \\
&= -(A^{i_1} \wedge A^4) I_a \quad \text{see equation (1.6), page 6} \\
&= -\rho_a \, [\![A^{i_1} A^4]\!] \\
&\simeq [\![A^{i_1} A^4]\!]
\end{aligned} \tag{3.53}$$

From equation (3.48) it follows that

$$[\![A^{i_1} A^4]\!] = [\![A^{i_1}]\!] \vee [\![A^4]\!]$$

That is, the line element $L_a^{i_1}$ can either be descibed by the outer product of two points along the line, i.e. $A_{i_2} \wedge A_{i_3}$, or by two planes[6] intersecting in $L_a^{i_1}$, i.e. $[\![A^{i_1}]\!] \vee [\![A^4]\!]$. The latter will be called the *dual representation* of a line. Obviously, there are not only two planes that intersect in line $L_a^{i_1}$. In fact, all pairs of planes that intersect in $L_a^{i_1}$ are a dual representation of that line. This set of planes will also be called a *pencil* of planes.

Another set of lines whose dual representation will be needed later on are the $\{A_i \wedge A_4\}$. Just as in equation (3.53)

$$\begin{aligned}
A_{i_1} \wedge A_4 &= \Big[(A_{i_1} \wedge A_4) \cdot I_a^{-1}\Big] I_a \\
&= \Big[(A_{i_1} \wedge A_4) \cdot (A^4 \wedge A^3 \wedge A^2 \wedge A^1)\Big] I_a \\
&= -\rho_a \, [\![A^{i_2} A^{i_3}]\!] \\
&\simeq [\![A^{i_2} A^{i_3}]\!]
\end{aligned} \tag{3.54}$$

[6] Note that $[\![A^4]\!] \simeq A_1 \wedge A_2 \wedge A_3$, the image plane of camera A. Also $[\![A^{i_1}]\!] \simeq A_{i_2} \wedge A_{i_3} \wedge A_4$. Since the meet operation picks out those elements that two objects have in common, it is clear that $[\![A^{i_1}]\!] \vee [\![A^4]\!] \simeq A_{i_2} \wedge A_{i_3} = L_a^{i_1}$.

Similarly, the dual representation of points can be found as follows

$$\begin{aligned} A_{\mu_1} &= \left[A_{\mu_1}\cdot I_a^{-1}\right] I_a \\ &= \left[A_{\mu_1}\cdot (A^4 \wedge A^3 \wedge A^2 \wedge A^1)\right] I_a \\ &= -\rho_a \; [\![A_a^{\mu_2} A_a^{\mu_3} A_a^4]\!] \\ &\simeq [\![A^{\mu_2} A^{\mu_3} A^4]\!] \end{aligned} \tag{3.55}$$

That is, the dual representation of a point is the intersection of three planes which intersect at just that point.

3.12 Epipoles

An *epipole* is the projection of the optical centre of one camera onto the image plane of another. Therefore epipoles contain important information about the relative placements of cameras.

As an example consider two cameras A and B represented by frames $\{A_i\}$ and $\{B_i\}$, respectively. The projection of the optical centre of camera B onto image plane A will be denoted E_{ab}. From equation (3.50) it follows that

$$E_{ab} = B_4 \cdot A^i A_i = \varepsilon^i_{ab} A_i \tag{3.56}$$

with $\varepsilon^i_{ab} \equiv B_4 \cdot A^i$. The only other epipole in this two camera set-up is E_{ba} given by

$$E_{ba} = A_4 \cdot B^i B_i = \varepsilon^i_{ba} B_i \tag{3.57}$$

with $\varepsilon^i_{ba} \equiv A_4 \cdot B^i$. If there are three cameras then each image plane contains two epipoles. With four cameras each image plane contains three epipoles. In general the total number of epipoles is $N(N-1)$ where N is the number of cameras present.

3.13 Camera Matrices

As was shown in equation (3.51) a camera matrix is defined by

$$K^{j\,\mathsf{b}}_{\mu} \equiv A_\mu \cdot B^j \tag{3.58}$$

where the $\{A_\mu\}$ are a frame of $\mathcal{P}^3$ and the $\{B^j\}$ are a subset of the reciprocal frame of $\{B_\mu\}$. Note that the raised, first index of $K_{j}^{i\,\mathsf{b}}$ indexes the row and the lowered, second index refers to the column of $K_{j}^{i\,\mathsf{b}}$. A point $X = \alpha^\mu A_\mu$ in $\mathcal{P}^3$ given in the A-frame is projected onto the B-image plane $(B_1 \wedge B_2 \wedge B_3)$ via,

$$\begin{aligned} X_b &= X \cdot B^j B_j \\ &= \alpha^\mu K_{\mu}^{j\,\mathsf{b}} B_j \end{aligned} \tag{3.59}$$

In general the projection of some vector X onto image plane P will be written as

$$X \xrightarrow{P} X_p \tag{3.60}$$

In this notation equation (3.59) becomes

$$X \xrightarrow{B} X_b = \alpha^\mu K_{\mu}^{j\,\mathsf{b}}\, B_j \tag{3.61}$$

There is also a dual representation of the A-frame.

$$A_{\mu_1} \simeq [\![A^{\mu_2} A^{\mu_3} A^{\mu_4}]\!] \tag{3.62}$$

where the $\{\mu_1, \mu_2, \mu_3, \mu_4\}$ are an even permutation of $\{1, 2, 3, 4\}$. Therefore, the camera matrix K^{b} can also be written as

$$K_{\mu_1}^{j\,\mathsf{b}} = [\![A^{\mu_2} A^{\mu_3} A^{\mu_4} B^j]\!] \tag{3.63}$$

The optical centre of camera A is given by A_4. The projection of A_4 onto the B-image plane therefore gives the epipole $E_{ba} = \varepsilon_{ba}^{j} B_j$ by definition. Hence,

$$E_{ba} = A_4 \cdot B^j\, B_j = K_{4}^{j\,\mathsf{b}}\, B_j \tag{3.64}$$

That is, the fourth column of K^{b} gives the coordinates of the epipole E_{ba}. The remaining 3×3 minor $K_{i}^{j\,\mathsf{b}}$ defines a plane collineation or *plane homography* between image planes A and B.

Let $\{B_\mu\}$ define a camera in $\mathcal{P}^3$ and $\{A_\mu\}$ be some other frame of the same projective space. Also, define A_4 to be the origin of $\mathcal{P}^3$. Then E_{ba} contains some information about the placement of camera B relative to the origin. Therefore, $A_4 \cdot B^j$ may be regarded as a *unifocal* tensor U_b.

$$U_b^i \equiv A_4 \cdot B^i = K_{4}^{i\,\mathsf{b}} \simeq [\![A^1 A^2 A^3 B^i]\!] \tag{3.65}$$

Obviously the unifocal tensor is of rank 1. The definition of a unifocal tensor is only done for completeness and is not strictly necessary since every unifocal tensor is also an epipole vector.

3.14 Conclusions

In this chapter we hope to have shown that GA is a very effective tool to describe geometry. In particular, GA seems to be a natural mathematical formalism for projective geometry. This is because in projective geometry points, lines and planes are represented by 1D, 2D and 3D-spaces, respectively, and GA allows us to express these spaces directly as algebraic objects through the outer product operation. Therefore, we are indeed not calculating with numbers but with points, lines and planes. Clearly, Grassmann algebra and Grassmann–Cayley algebra can also be used for this purpose. However, the existence of a general inner product in GA allows us to define a meet operation, which does not exist in Grassmann algebra. In GC algebra the meet is defined directly and not through a more fundamental operation like the inner product in GA.

As we have seen in this chapter, reciprocal frames are a powerful tool to express projections and intersections. In particular, we are able not only to work with vector frames but also with, line and plane frames. This allows us, for example, to project a line onto an image plane using directly a reciprocal line frame.

We also presented a novel way of relating projective, affine and Euclidean geometry in GA, which we believe is more intuitive and powerful than the method presented by Hestenes and Ziegler [31]. The advantage of our approach is that we do not need to assume any particular signature of an underlying orthonormal frame. Indeed, we do not need to assume an underlying orthonormal frame at all for our projection between projective and affine spaces to work.

Chapter 4

The geometry of multiple view tensors

Virtue itself turns vice, being misapplied,
And vice sometime's by action dignified.

"Romeo and Juliet", William Shakespeare.

4.1 Introduction

Multiple view tensors play a central role in many areas of Computer Vision. The *Fundamental Matrix*, *Trifocal Tensor* and *Quadfocal Tensor* have been investigated by many researchers using a number of different formalisms. For example, standard matrix analysis has been used in [26, 27, 59, 60, 33]. An analysis of multiple view tensors in terms of Grassmann-Cayley (GC) algebra can be found in [18, 47, 14, 15]. GA has also been applied to the problem [48, 49, 39, 37].

The importance of multiple view tensors is due to the geometric information they encode of a camera system. As the names suggest, for a two camera system we can find a *bifocal tensor* or fundamental matrix, for three cameras a *trifocal tensor* and for four cameras a *quadfocal tensor*. Finding the multiple view tensor of a camera system is also called "weak calibration". *Weak* calibration because a multiple view tensor allows us to find *projective* (not metric) reconstructions of point or line matches.

In order to calculate a mutliple view tensor, we need to know a minimum number of point or line matches. The minimum number of matches necessary depends on the tensor and whether lines or points are used. In general, point matches encode more information than line matches, and thus less point than line matches are needed.

The basic idea behind the definition of a multiple view tensor is very simple. In each case we consider the projections $\{X_1^i, X_2^i, \dots, X_n^i\}$ of some objects $\{X^i\}$ onto image planes $\{1, 2, \dots, n\}$. Then we define some multilinear *multiple view function* M such that the set of image sets $\{X_1^i, X_2^i, \dots, X_n^i\}$ forms the kernel of M, i.e.

$$M(X_1^i, X_2^i, \dots, X_n^i) = 0\,, \qquad \forall\; i.$$

In the case of two, three and four cameras, the function M can be represented by the contraction of image point coordinate vectors with a tensor.

The problem to which a lot of research has been dedicated, is then to find the appropriate tensor given a set of point or line matches [26, 24, 28, 42, 39, 48, 14, 17, 32]. Since in real applications such matches may contain numerical errors, an optimized tensor has to be found for the given data. As it turns out, multiple view tensors are an overparameterization of the actual multiple view functions. Therefore, constraints have to be found on the tensors to make them true representation of the corresponding multiple view functions. These constraints are usually of geometric nature. It makes therefore sense to investigate multiple view tensors with a geometric algebra, to understand the geometry they encode.

The advantage we have in using a geometric algebra as GA (or GC algebra), is that we can work with the underlying basis frames of the cameras and not just with the coordinates of projections of points and lines. This enables us to work directly with the geometry underlying the multiple view functions, of which the multiple view tensors are only an (overparameterized) "manifestation".

In this chapter we will show how GA can be used to give a unified *geometric* picture of multiple view tensors. It will be seen that with the GA approach multiple view tensors can be derived from simple geometric considerations. In particular, constraints on the internal structure of multiple view tensors will all be derived from the trivial fact that the intersection points of a line with three planes, all lie along a line. Our analysis will also show how closely linked the numerous different expressions for multiple view tensors are.

The structure of this chapter will be as follows. First we describe the fundamental matrix (bifocal tensor), the trifocal tensor and the quadfocal tensor in detail, investigating their derivations, inter-relations and other properties. Following on from these analytical investigations, we show how the self-consistency of a trifocal tensor influences its reconstruction quality. We end this chapter with some conclusions and a table summarising the main properties of the three multiple view tensors described here.

4.2 The Fundamental Matrix

4.2.1 Derivation

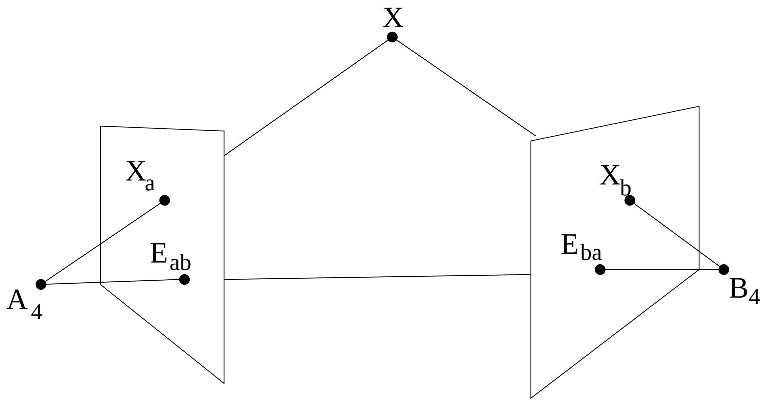


Figure 4.1: Point X projected onto two cameras producing images X_a and X_b. E_{ba} and E_{ab} are the respective epipoles of the cameras.

Let $\{A_\mu\}$ and $\{B_\mu\}$ define two cameras in $\mathcal{P}^3$ (see figure 4.1). A point X in $\mathcal{P}^3$ may be transformed into the A and B frames via

$$X = X \cdot A_a^\mu A_\mu = X \cdot B_b^\nu B_\nu \tag{4.1}$$

Recall that there is an implicit summation over μ and ν. From that follows that the line $A_4 \wedge X$ can also be written as

$$\begin{aligned} A_4 \wedge X &= X \cdot A_a^i \; A_4 \wedge A_i \\ &= \rho_a^{-1} \, A_4 \wedge X_a \end{aligned} \tag{4.2}$$

where $X_a = X \cdot A^i \, A_i$. Let X_a and X_b be the images of some point $X \in \mathcal{P}^3$ taken by cameras A and B, respectively. Then, since the lines from A and B to X intersect at X

$$\begin{aligned} 0 &= (\, A_4 \wedge X \wedge B_4 \wedge X \,) I^{-1} \\ &\simeq (\, A_4 \wedge X_a \wedge B_4 \wedge X_b \,) I^{-1} \\ &= \alpha^i \beta^j [\![A_4 A_i B_4 B_j]\!] \end{aligned} \tag{4.3}$$

where $\alpha^i \equiv X \cdot A^i$ and $\beta^j \equiv X \cdot B^j$ are the image point coordinates of X_a and X_b, respectively. Therefore, for a *Fundamental Matrix* defined as

$$F_{ij} \equiv [\![A_4 A_i B_4 B_j]\!] \tag{4.4}$$

we have

$$\alpha^i \beta^j F_{ij} = 0 \tag{4.5}$$

if the image points given by $\{\alpha^i\}$ and $\{\beta^j\}$ are images of the same point in space. Note, however, that equation (4.5) holds as long as X_a is the image of any point along $A_4 \wedge X_a$ and X_b is the image of any point along $B_4 \wedge X_b$. In other words, the condition in equation (4.5) only ensures that lines $A_4 \wedge X_a$ and $B_4 \wedge X_b$ are co-planar.

In the following let any set of indices of the type $\{i_1, i_2, i_3\}$ be an even permutation of $\{1, 2, 3\}$. From equation (3.54), p.62, it follows that

$$[\![B_4 B_{j_1}]\!] \simeq B^{j_2} \wedge B^{j_3}. \tag{4.6}$$

Thus, with the help of equation (3.16), p.48, equation (4.4) can also be written as

$$F_{ij_1} \simeq (A_i \wedge A_4) \cdot (B^{j_2} \wedge B^{j_3}) \tag{4.7}$$

This may be expanded to

$$\begin{aligned} F_{ij_1} &= (A_4 \cdot B^{j_2})(A_i \cdot B^{j_3}) - (A_4 \cdot B^{j_3})(A_i \cdot B^{j_2}) \\ &= U_b^{j_2} K_i^{j_3\,\mathsf{b}} - U_b^{j_3} K_i^{j_2\,\mathsf{b}} \end{aligned} \tag{4.8}$$

That is, the Fundamental Matrix is just the standard cross product between the epipole[1] $U_b^{\bullet}$ and the column vectors $K_i^{\bullet\,\mathsf{b}}$.

$$F_{i\bullet} \simeq U_b^{\bullet} \times K_i^{\bullet\,\mathsf{b}} \tag{4.9}$$

In order to have a unified naming convention the Fundamental Matrix will also be refered to as the *bifocal tensor*.

4.2.2 Rank of F

Note that we use the term "rank" in relation to tensors in order to generalize the notion of rank as used for matrices. That is, we describe a rank 2 matrix as a rank 2, 2-valence tensor.

[1] Recall that $U_b \equiv E_{ba}$.

The different meanings of "rank" both in use today maybe somewhat confusing. The root of this confusion is described in [55]. The *tensor calculus* already used by Einstein and Grossmann in 1913, had been invented by the Italian geometer Gregorio Ricci-Curbastro (1853–1925) in 1884. However, Ricci did not use the name "tensor". This term was introduced by Einstein and Grossmann. They also called the "order" of a tensor *rank*. The dutch geometer Jan Arnoldus Schouten (1883–1971) on the other hand, called Ricci's objects *affinors* and called their "orders" *valences*, borrowing this term from chemistry. We will use the term *tensor* in the sense of Einstein, *valence* in the sense of Schouten and *rank* as it is used for matrices. This terminology is also used in [61, 62] to discuss rank constraints on multiple view tensors.

In general a tensor may be decomposed into a linear combination of rank 1 tensors. The minimum number of terms necessary for such a decomposition gives the *rank* of the tensor. For example, a rank 1, 2-valence tensor M is created by combining the components $\{\alpha^i\}$, $\{\beta^i\}$ of two vectors as $M^{ij} = \alpha^i \beta^j$.

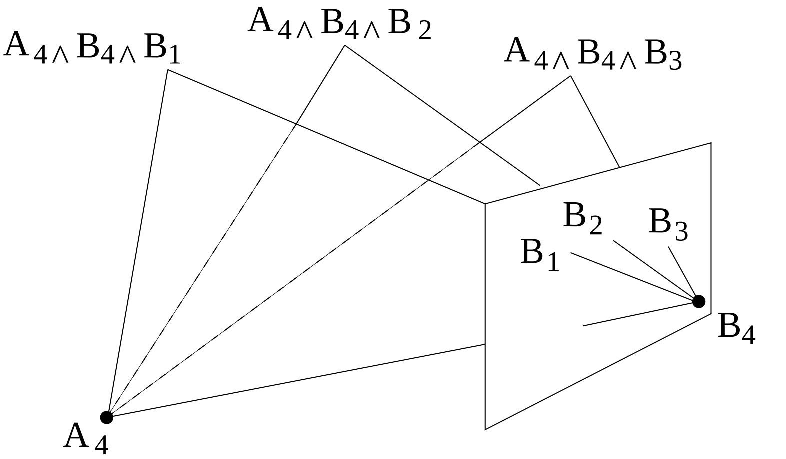

$A_4 \wedge B_4 \wedge B_1$
$A_4 \wedge B_4 \wedge B_2$
$A_4 \wedge B_4 \wedge B_3$
B_2
B_3
B_1
B_4
A_4

Figure 4.2: Geometric derivation of the rank of the fundamental matrix.

The rank of F can be found quite easily from geometric considerations. Equation (4.4) can also be written as

$$F_{ij} \simeq A_i \cdot [\![A_4 B_4 B_j]\!] \tag{4.10}$$

The expression $[\![A_4 B_4 B_j]\!]$ gives the normal to the plane $(A_4 \wedge B_4 \wedge B_j)$. This defines three planes, one for each value of j, all of which contain the line $A_4 \wedge B_4$ (see figure 4.2). Hence, all three normals lie in a plane. Furthermore, no two normals are identical since the $\{B_j\}$ are linearly independent by definition. It follows directly that at most two columns of F_{ij} can be linearly independent. Therefore, F is of rank 2.

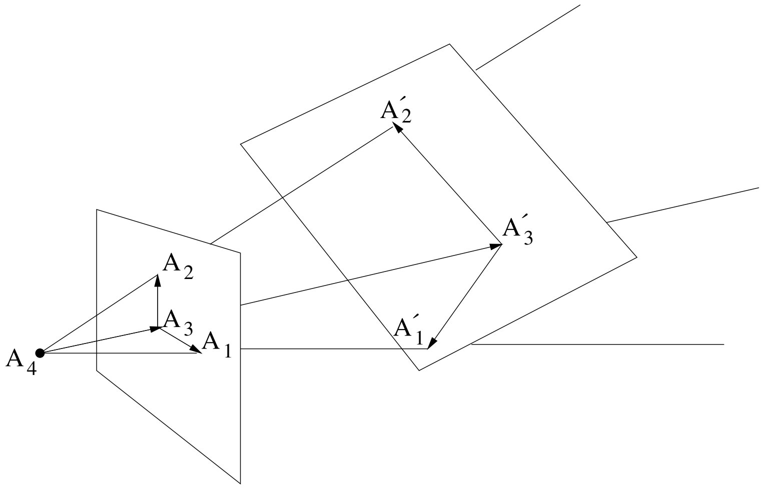


Figure 4.3: Change of image plane frame which leaves the fundamental matrix unchanged up to an overall scale.

The rank of the bifocal tensor F can also be arrived at through a *minimal* decomposition of F into rank 1 tensors. To achieve this we first define a new A-image plane frame $\{A'_i\}$ as

$$A'_i \equiv s(A_i + t_i A_4) \tag{4.11}$$

where s and the $\{t_i\}$ are some scalar components. Thus we have

$$\begin{aligned} A_4 \wedge A'_i &= s A_4 \wedge (A_i + t_i A_4) \\ &= s A_4 \wedge A_i \end{aligned} \tag{4.12}$$

Hence, F is left unchanged up to an overall scale factor under the transformation $A_i \longrightarrow A'_i$. In other words, the image plane bases $\{A_i\}$ and $\{B_j\}$ can be changed along the projective rays $\{A_4 \wedge A_i\}$ and $\{B_4 \wedge B_j\}$, respectively, without changing the bifocal tensor relating the two cameras (see figure 4.3). This fact limits the use of the bifocal tensor, since it cannot give any information about the actual placement of the image planes.

Define two bifocal tensors F and F' as

$$F_{ij} = [\![A_4 A_i B_4 B_j]\!] \tag{4.13a}$$
$$F'_{ij} = [\![A_4 A'_i B_4 B_j]\!] \tag{4.13b}$$

From equation (4.12) it follows directly that $F_{ij} \simeq F'_{ij}$. Since the $\{A'_i\}$ can be chosen arbitrarily along the line $A_4 \wedge A_i$ we may write

$$A'_i = (A_4 \wedge A_i) \vee P \tag{4.14}$$

where P is some plane in $\mathcal{P}^3$. $P = (B_4 \wedge B_1 \wedge B_2)$ seems a good choice, since then the $\{A'_i\}$ all lie in a plane together with B_4. The effect of this is that the projections of the $\{A'_i\}$ on image plane B will all lie along a line. The matrix $A'_i \cdot B^j$ therefore only has two linearly independent columns because the column vectors *are* the projections of the $\{A'_i\}$ onto image plane B. That is, $A'_i \cdot B^j$, which is the 3×3 minor of K^{b}, is of rank 2.

This matrix could only be of rank 1, if the $\{A'_i\}$ were to project to a single point on image plane B, which is only possible if they lie along a line in $\mathcal{P}^3$. However, then they could not form a basis for image plane A which they were defined to be.

Thus $A'_i \cdot B^j$ can minimally be of rank 2. Such a minimal form is what we need to find a minimal decomposition of F into rank 1 tensors using equation (4.8). Substituting $P = (B_4 \wedge B_1 \wedge B_2)$ into equation (4.14) gives

$$\begin{aligned} A'_i &= (A_4 \wedge A_i) \vee (B_4 \wedge B_1 \wedge B_2) \\ &= \llbracket A_4 A_i \rrbracket \cdot (B_4 \wedge B_1 \wedge B_2) \\ &= \llbracket A_4 A_i B_4 B_1 \rrbracket B_2 - \llbracket A_4 A_i B_4 B_2 \rrbracket B_1 + \llbracket A_4 A_i B_1 B_2 \rrbracket B_4 \\ &= F_{i1} B_2 - F_{i2} B_1 + \llbracket A_4 A_i B_1 B_2 \rrbracket B_4 \end{aligned} \tag{4.15}$$

Expanding F' in the same way as F in equation (4.8) and substituting the above expressions for the $\{A'_i\}$ gives

$$\begin{aligned} F'_{ij_1} &= (A_4 \cdot B^{j_2})(A'_i \cdot B^{j_3}) - (A_4 \cdot B^{j_3})(A'_i \cdot B^{j_2}) \\ &= \quad (A_4 \cdot B^{j_2})\Big[- F_{i2}(B_1 \cdot B^{j_3}) + F_{i1}(B_2 \cdot B^{j_3})\Big] \\ &\quad - (A_4 \cdot B^{j_3})\Big[- F_{i2}(B_1 \cdot B^{j_2}) + F_{i1}(B_2 \cdot B^{j_2})\Big] \\ &= \quad \varepsilon^{j_2}_{ba}\Big[- F_{i2}\, \rho_b \delta^{j_3}_1 + F_{i1}\, \rho_b \delta^{j_3}_2 \Big] \\ &\quad - \varepsilon^{j_3}_{ba}\Big[- F_{i2}\, \rho_b \delta^{j_2}_1 + F_{i1}\, \rho_b \delta^{j_2}_2 \Big] \\ &= \quad F_{i1}\Big[\varepsilon^{j_2}_{ba}\, \rho_b \delta^{j_3}_2 - \varepsilon^{j_3}_{ba}\, \rho_b \delta^{j_2}_2\Big] \\ &\quad - F_{i2}\Big[\varepsilon^{j_2}_{ba}\, \rho_b \delta^{j_3}_1 - \varepsilon^{j_3}_{ba}\, \rho_b \delta^{j_2}_1\Big] \end{aligned} \tag{4.16}$$

where we used $B_4 \cdot B^j = 0$ and $B_i \cdot B^j = \rho_b \delta^j_i$. Clearly, F_{i1}, F_{i2} and the expressions in the square brackets all represent vectors. Therefore, equation (4.16) expresses F' as a linear combination of two rank 1 tensors (matrices). This shows again that the bifocal tensor is of rank 2.

But why should we do all this work of finding a minimal decomposition of F if its rank can be found so much more easily from geometric considerations? There are two good reasons:

1. for the trifocal and quadfocal tensor, a minimal decomposition will be the easiest way to find the rank, and
2. such a decomposition is useful for evaluating F with a non-linear algorithm, since the self-consistency constraints on F are automatically satisfied.

4.2.3 Degrees of Freedom of F

Equation (4.16) is in fact a minimal parameterisation of the bifocal tensor. This can be seen by writing out the columns of F'.

$$F'_{i1} = -\varepsilon^3_{ba}\,\rho_b\,F_{i1}\,; \;\; F'_{i2} = -\varepsilon^3_{ba}\,\rho_b\,F_{i2}\,; \;\; F'_{i3} = \rho_b\,(\varepsilon^1_{ba}F_{i1} + \varepsilon^2_{ba}F_{i2}) \quad (4.17)$$

As expected, the third column (F_{i3}) is a linear combination of the first two. Since an overall scale is not important we can also write

$$F'_{i1} = F_{i1}\,; \quad F'_{i2} = F_{i2}\,; \quad F'_{i3} = -\bar{\varepsilon}^1_{ba}F_{i1} - \bar{\varepsilon}^2_{ba}F_{i2} \quad (4.18)$$

where $\bar{\varepsilon}^i_{ba} \equiv \varepsilon^i_{ba}/\varepsilon^3_{ba}$. This is the most general form of a rank 2, 3 × 3 matrix. Furthermore, since there are no more constraints on F_{i1} and F_{i2} this is also a minimal parameterisation of the bifocal tensor. That is, eight parameters are minimally necessary to form the bifocal tensor. It follows that since an overall scale is not important the bifocal tensor has *seven* degrees of freedom (DOF).

This DOF count can also be arrived at from more general considerations: each camera matrix has 12 components. However, since an overall scale is not important, each camera matrix adds only 11 DOF. Furthermore, the bifocal tensor is independent of the choice of basis. Therefore, it is invariant under a projective transformation, which has 16 components. But again, an overall scale is not important. Thus only 15 DOF can be subtracted from the DOF count due to the camera matrices. For two cameras we therefore have $2 \times 11 - 15 = 7$ DOF.

4.2.4 Transferring Points with F

The bifocal tensor can also be used to transfer a point in one image to a line in the other. Starting again from equation (4.4) the bifocal tensor can be written as

$$\begin{aligned} F_{ij} &= [\![A_i A_4 B_j B_4]\!] \\ &= (A_i \wedge A_4)\cdot[\![B_j B_4]\!] \\ &= (A_i \wedge A_4)\cdot L^b_j \end{aligned} \quad (4.19)$$

This shows that F_{ij} gives the components of the projection of line $(A_i \wedge A_4)$ onto image plane B. Therefore,

$$(A_i \wedge A_4) \xrightarrow{B} F_{ij} L^j_b. \tag{4.20}$$

Since $A_4 \xrightarrow{B} E_{ba}$ (the epipole on image plane B), $F_{ij} L^j_b$ defines an epipolar line.

Thus, contracting F with the coordinates of a point on image plane A, results in the homogeneous line coordinates of a line passing through the corresponding point on image plane B and the epipole E_{ba}.

$$\alpha^i F_{ij} = \lambda^b_j \tag{4.21}$$

where the $\{\alpha^i\}$ are some point coordinates and the $\{\lambda^b_j\}$ are the homogeneous line coordinates of an epipolar line.

4.2.5 Epipoles of F

Recall that if there are two cameras then two epipoles are defined;

$$E_{ab} \equiv B_4 \cdot A^i A_i = \varepsilon^i_{ab} A_i \tag{4.22a}$$

$$E_{ba} \equiv A_4 \cdot B^i B_i = \varepsilon^i_{ba} B_i \tag{4.22b}$$

Contracting F_{ij} with ε^i_{ab} gives

$$\begin{aligned} \varepsilon^i_{ab} F_{ij} &= \varepsilon^i_{ab} [\![A_4 A_i B_4 B_j]\!] \\ &= \rho_a [\![A_4 (\, B_4 \cdot A^i_a \, A_i) B_4 B_j]\!] \\ &= \rho_a [\![A_4 B_4 B_4 B_j]\!] \ ; \quad \text{from equation (4.2)} \\ &= 0 \end{aligned} \tag{4.23}$$

Similarly,

$$\varepsilon^j_{ba} F_{ij} = 0 \tag{4.24}$$

Therefore, vectors $\{\varepsilon^i_{ab}\}$ and $\{\varepsilon^j_{ba}\}$ can be regarded respectively as the left and right null spaces of matrix F. Given a bifocal tensor F, its epipoles can therefore easily be found using, for example, a singular value decomposition (SVD).

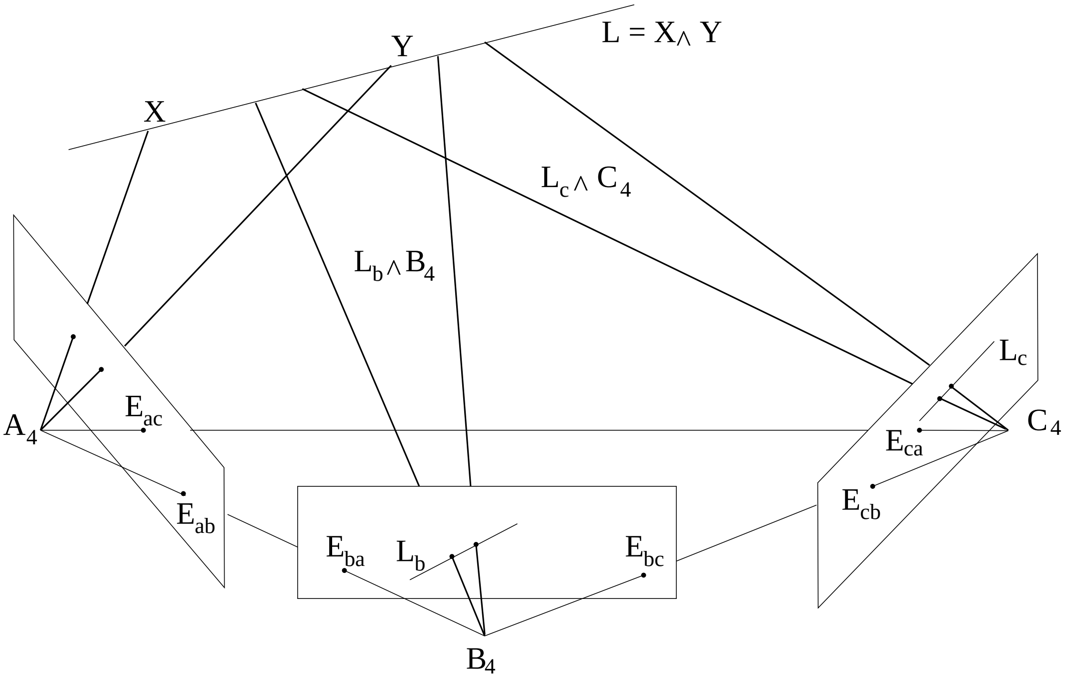


Figure 4.4: Line projected onto three image planes.

4.3 The Trifocal Tensor

4.3.1 Derivation

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 $\lambda^b_i L^i_b{\wedge}B_4$, up to a scalar factor, where $\lambda^b_i = L{\cdot}L^b_i$. But,

$$L^{i_1}_b{\wedge}B_4 = B_{i_2}{\wedge}B_{i_3}{\wedge}B_4 = [\![B^{i_1}]\!]$$

Intersecting planes $L{\wedge}B_4$ and $L{\wedge}C_4$ has to give L. Therefore, $(\lambda^b_i[\![B^i]\!]){\vee}(\lambda^c_j[\![C^j]\!])$ 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 4.4 shows this construction. Combining all these expressions gives

$$\begin{aligned} 0 &= (X{\wedge}A_4{\wedge}L)I^{-1} \\ &= \alpha^i\lambda^b_j\lambda^c_k\Big[\!\!\Big[(A_i{\wedge}A_4)([\![B^j]\!] \vee [\![C^k]\!])\Big]\!\!\Big] \qquad (4.25) \\ &= \alpha^i\lambda^b_j\lambda^c_k\Big[\!\!\Big[(A_i{\wedge}A_4)[\![B^jC^k]\!]\Big]\!\!\Big] \end{aligned}$$

where the identity from equation (3.48) was used. If the trifocal tensor $T_i^{\,jk}$ is defined as

$$T_i^{\,jk} = \Big[\!\!\Big[(A_i\wedge A_4)[\![B^jC^k]\!]\Big]\!\!\Big] \tag{4.26}$$

then, from equation (4.25) it follows that it has to satisfy $\alpha^i\lambda_j^b\lambda_k^cT_i^{\,jk} = 0$. This expression for the trifocal tensor can be expanded in a number of different ways. One of them is,

$$\begin{aligned} T_i^{\,jk} &= (A_i\wedge A_4)\cdot[\![[\![B^jC^k]\!]]\!] \\ &= (A_i\wedge A_4)\cdot(B^j\wedge C^k) \\ &= (A_4\cdot B^j)(A_i\cdot C^k) - (A_4\cdot C^k)(A_i\cdot B^j) \\ &= U_b^jK_{k_i}^c - U_c^kK_{j_i}^b \end{aligned} \tag{4.27}$$

where $K_{j_i}^b \equiv A_i\cdot B^j$ and $K_{k_i}^c \equiv A_i\cdot C^k$ are the camera matrix minors for cameras B and C, respectively, relative to camera A. This is the expression for the trifocal tensor given by Hartley in [26]. Note that the camera matrix for camera A would be written as $K_{j_\mu}^a \equiv A_\mu\cdot A^j \simeq \delta_i^j$. That is, $K^a = [I|0]$ in standard matrix notation. In many other derivations of the trifocal tensor (eg. [26]) 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.

4.3.2 Transferring Lines

The trifocal tensor can be used to transfer lines from two images to the third. That is, if the image of a line in $\mathcal{P}^3$ is known on two image planes, then its image on the third image plane can be found. This can be seen by expanding equation (4.26) in the following way,

$$\begin{aligned} T_i^{\,jk} &= [\![A_iA_4]\!]\cdot[\![B^jC^k]\!] \\ &= L_i^a\cdot[\![B^jC^k]\!] \end{aligned} \tag{4.28}$$

This shows that the trifocal tensor gives the homogeneous line components of the projection of line $[\![B^jC^k]\!]$ onto image plane A. That is,

$$[\![B^jC^k]\!] \xrightarrow{A} T_i^{\,jk}L_a^i \tag{4.29}$$

It will be helpful later on to define the following two lines.

$$\begin{aligned} T^{jk} &\equiv [\![B^j C^k]\!] &&(4.30a)\\ T_a^{jk} &\equiv T_i^{\,jk} L_a^i &&(4.30b)\end{aligned}$$

such that $T^{jk} \xrightarrow{A} T_a^{jk}$. Let the $\{\lambda_j^b\}$ and $\{\lambda_k^c\}$ be the homogeneous line coordinates of the projection of some line $L \in \mathcal{P}^3$ onto image planes B and C, respectively. Then recall that $\lambda_j^b \lambda_k^c [\![B^j C^k]\!]$ gives L up to an overall scalar factor, i.e.

$$L \simeq \lambda_j^b \lambda_k^c [\![B^j C^k]\!]\ ; \qquad \lambda_j^b \equiv L\cdot L_j^b \ \text{ and } \ \lambda_k^c \equiv L\cdot L_k^c \tag{4.31}$$

The image of L on image plane A, L_a, can therefore be found via

$$\begin{aligned} L_a &= L\cdot L_i^a L_a^i \\ &\simeq \lambda_j^b \lambda_k^c [\![B^j C^k]\!]\cdot L_i^a L_a^i \\ &= \lambda_j^b \lambda_k^c T_i^{\,jk} L_a^i \end{aligned} \tag{4.32}$$

Thus, we have

$$\lambda_i^a \simeq \lambda_j^b \lambda_k^c T_i^{\,jk} \tag{4.33}$$

4.3.3 Transferring Points

It is also possible to find the image of a point on one image plane if its image is known on the other two. To see this, the expression for the trifocal tensor needs to be expanded in yet another way. Substituting the dual representation of line $A_{i_1} \wedge A_4$, i.e. $[\![A^{i_2} A^{i_3}]\!]$ into equation (4.26) gives

$$\begin{aligned} T_{i_1}^{\,jk} &= \Big[\!\!\Big[(A_{i_1}\wedge A_4)[\![B^j C^k]\!]\Big]\!\!\Big] \\ &\simeq \Big[\!\!\Big[[\![A^{i_2} A^{i_3}]\!][\![B^j C^k]\!]\Big]\!\!\Big] \\ &= [\![A^{i_2} A^{i_3}]\!]\cdot(B^j \wedge C^k) \\ &= [\![A^{i_2} A^{i_3} B^j C^k]\!] \end{aligned} \tag{4.34}$$

It can be shown that this form of the trifocal tensor is equivalent to the determinant form given by Heyden in [32]. Now only one more step is needed to see how the trifocal tensor may be used to transfer points.

$$\begin{aligned} T_{i_1}^{\,jk} &\simeq [\![A^{i_2} A^{i_3} B^j C^k]\!] \\ &= [\![A^{i_2} A^{i_3} B^j]\!]\cdot C^k \\ &= X_{i_1}^{\,j\,\mathsf{T}}\cdot C^k\ ; \quad X_{i_1}^{\,j\,\mathsf{T}} \equiv [\![A^{i_2} A^{i_3} B^j]\!] \end{aligned} \tag{4.35}$$

Note that the points $\{X_{i_1}^{j\,\mathsf{T}}\}$ are defined through their dual representation as the set of intersection points of lines $\{A_{i_1} \wedge A_4\}$ ($\simeq \{[\![A^{i_2}A^{i_3}]\!]\}$) and planes $\{[\![B^j]\!]\}$ ($\simeq \{L_b^j \wedge B_4\}$). Let $L = X \wedge Y$ be a line in $\mathcal{P}^3$. Then

$$X \xrightarrow{A} X_a = \alpha^i A_i \tag{4.36a}$$

$$L \xrightarrow{B} L_B = \lambda_j^b L_b^j \tag{4.36b}$$

Hence

$$\begin{aligned} X &\simeq (\alpha^{i_1} \underbrace{A_{i_1} \wedge A_4}_{[\![A^{i_2}A^{i_3}]\!]}) \vee (\lambda_j^b \underbrace{L_b^j \wedge B_4}_{[\![B^j]\!]}) \\ &\simeq \textstyle\sum_{i_1} \alpha^{i_1} \lambda_j^b \Big([\![A^{i_2}A^{i_3}]\!] \vee [\![B^j]\!] \Big) \\ &= \textstyle\sum_{i_1} \alpha^{i_1} \lambda_j^b [\![A^{i_2}A^{i_3}B^j]\!] \\ &= \alpha^{i_1} \lambda_j^b \, X_{i_1}^{j\,\mathsf{T}} \end{aligned} \tag{4.37}$$

Now, the projection of X onto image plane C is simply

$$\begin{aligned} X_c &= X \cdot C^k \, C_k \\ &\simeq \alpha^i \lambda_j^b \, X_{i_1}^{j\,\mathsf{T}} \cdot C^k \, C_k \\ &\simeq \alpha^i \lambda_j^b \, T_i^{\ jk} \, C_k \end{aligned} \tag{4.38}$$

That is,

$$\eta^k \simeq \alpha^i \lambda_j^b \, T_i^{\ jk} \tag{4.39}$$

with $\eta^k \equiv X \cdot C^k$. Similarly we also have,

$$\beta^k \simeq \alpha^i \lambda_k^c \, T_i^{\ jk} \tag{4.40}$$

Therefore, if the image of a point and a line through that point are known on two image planes, respectively, then the image of the point on the third image plane can be calculated. Note that the line defined by the $\{\lambda_j^b\}$ can be any line that passes through the image of X on image plane B. That is, we may choose the point $(0, 0, 1)$ as the other point the line passes through. Then we have

$$\lambda_1^b = \beta^2 \, ; \quad \lambda_2^b = -\beta^1 \, ; \quad \lambda_3^b = 0 \tag{4.41}$$

Hence, equation (4.39) becomes

$$\eta^k \simeq \alpha^i (\beta^2 T_i^{\ 1k} - \beta^1 T_i^{\ 2k}) \tag{4.42}$$

and equation (4.40) becomes

$$\beta^k \simeq \alpha^i(\eta^2 T_i^{\ j1} - \eta^1 T_i^{\ j2}) \tag{4.43}$$

4.3.4 Rank of T

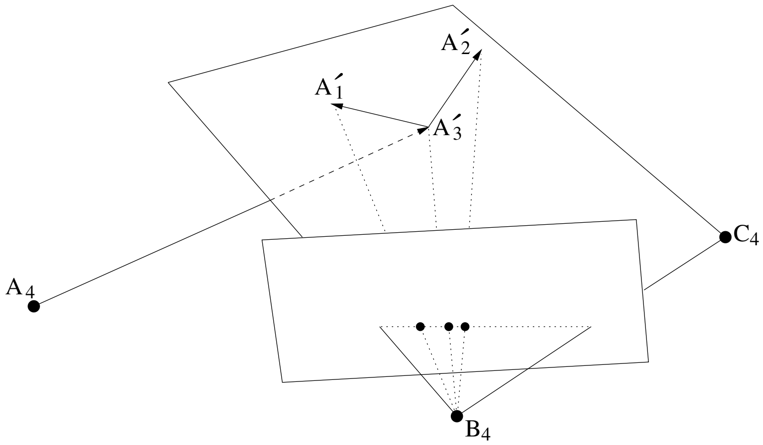


Figure 4.5: Change of image plane frame which leaves the trifocal tensor unchanged but gives minimum rank for camera matrices K^b and K^c.

Finding the rank of T is somewhat harder than for the bifocal tensor, mainly because there is no simple geometric construction which yields its rank. As was mentioned before the rank of a tensor is given by the minimum number of terms necessary for a linear decomposition of it in terms of rank 1 tensors[2]. As for the bifocal tensor, the transformation $A_i \to A'_i = s(A_i + t_i A_4)$ leaves the trifocal tensor unchanged up to an overall scale. A good choice for the $\{A'_i\}$ seems to be

$$A'_i = (A_i \wedge A_4) \vee (B_3 \wedge B_4 \wedge C_4) \tag{4.44}$$

since then all the $\{A'_i\}$ lie in a plane together with B_4 and C_4 (see figure 4.5). Therefore, the camera matrix minors $K_i^{j\ \mathsf{b}} = A'_i \cdot B^j$ and $K_i^{k\ \mathsf{c}} = A'_i \cdot C^k$ are of rank 2. As was shown before, this is the minimal rank camera matrix minors can have. To see how this may help to find a minimal decomposition of T recall equation (4.27);

$$T_i^{\ jk} = U_b^j K_{k_i}^c - U_c^k K_{j_i}^b$$

[2] For example, a rank 1 3-valence tensor is created by combining the components $\{\alpha^i\}$, $\{\beta^i\}$, $\{\eta^i\}$ of three vectors as $T^{ijk} = \alpha^i \beta^j \eta^k$.

This decomposition of T shows that its rank is at most 6, since U_b and U_c are vectors, and K^c and K^b cannot be of rank higher than 3. Using the above choice for K^b and K^c however shows that the rank of T is 4, since then the rank of the camera matrices is minimal, and we thus have a minimal linear decomposition of T.

The rank of T was also derived by Shashua and Avidan [61] where it is used to relate sets of trifocal tensors. The rank result is also used in [62] to show that there are no configurations of world points whose matching projections produce an ambiguous solution for the trifocal tensor.

4.3.5 Degrees of Freedom of T

As for the bifocal tensor we can also write down an explicit parameterisation for the trifocal tensor. Starting with equation (4.44) we get

$$\begin{aligned} A'_i &= (A_i \wedge A_4) \vee (B_3 \wedge B_4 \wedge C_4) \\ &= [\![A_i A_4]\!] \cdot (B_3 \wedge B_4 \wedge C_4) \\ &= [\![A_i A_4 B_4 C_4]\!] B_3 - [\![A_i A_4 B_3 C_4]\!] B_4 + [\![A_i A_4 B_3 B_4]\!] C_4 \\ &= \alpha_i^1 B_3 + \alpha_i^2 B_4 + \alpha_i^3 C_4 \end{aligned} \tag{4.45}$$

where α_i^1, α_i^2 and α_i^3 are defined appropriately. The trifocal tensor may be expressed in terms of the $\{A'_i\}$ as follows (see equation (4.27)).

$$\begin{aligned} T_i^{jk} &= (A_4 \cdot B^j)(A'_i \cdot C^k) - (A_4 \cdot C^k)(A'_i \cdot B^j) \\ &= \quad (A_4 \cdot B^j)\Big[\alpha_i^1 B_3 \cdot C^k + \alpha_i^2 B_4 \cdot C^k\Big] \\ &\quad - \ (A_4 \cdot C^k)\Big[\alpha_i^1 B_3 \cdot B^j + \alpha_i^3 C_4 \cdot B^j\Big] \\ &= \quad \varepsilon_{ba}^{j}\Big[\alpha_i^1 B_3 \cdot C^k + \alpha_i^2 \varepsilon_{cb}^{k}\Big] \\ &\quad - \ \varepsilon_{ca}^{k}\Big[\alpha_i^1 \rho_b \delta_3^j + \alpha_i^3 \varepsilon_{bc}^{j}\Big] \end{aligned} \tag{4.46}$$

This decomposition of T has $5 \times 3 + 3 \times 3 - 1 = 23$ DOF. The general formula for finding the DOF of T gives $3 \times 11 - 15 = 18$ DOF. Therefore, equation (4.46) is an overdetermined parameterisation of T. However, it will still satisfy the self-consistency constraints of T.

Note that the decomposition of T into rank 1 tensors in equation (4.46) only implements the rank 2 constraints of the camera matrix minors. Since equation (4.46) does not yield a minimal parameterisation of T, there have to be additional constraints on T.

Recall that a decomposition of F into rank 1 tensors *does* give a minimal parameterisation of F. Hence, the rank 2 constraint on the camera matrix minor in equation (4.8) is the only constraint that exists for F.

4.3.6 Constraints on T

To understand the structure of T further, we will derive self-consistency constraints for T. Heyden derives the constraints on T using the "quadratic p-relations" [32]. In GA these relations can easily be established from geometric considerations.

The simplest constraint on T may be found as follows. Recall equation (4.35), where the trifocal tensor was expressed in terms of the projection of points $X_{i_1}^{j\,\top} = \llbracket A_a^{i_2} A_a^{i_3} B^j \rrbracket$ onto image plane C, i.e.

$$T_{i_1}^{jk} = X_{i_1}^{j\,\top} \cdot C^k$$

Now consider the following trivector.

$$\begin{aligned}
&X_{i_1}^{j_a\top} \wedge X_{i_1}^{j_b\top} \wedge X_{i_1}^{j_c\top} \\
&= \Big(\llbracket A_a^{i_2} A_a^{i_3} \rrbracket \vee \llbracket B^{j_a} \rrbracket \Big) \\
&\quad \wedge \Big(\llbracket A_a^{i_2} A_a^{i_3} \rrbracket \vee \llbracket B^{j_b} \rrbracket \Big) \wedge \Big(\llbracket A_a^{i_2} A_a^{i_3} \rrbracket \vee \llbracket B^{j_c} \rrbracket \Big) \\
&= 0
\end{aligned} \tag{4.47}$$

The first step follows from equation (3.48). It is clear that this expression is zero because we take the outer product of the intersection points of line $\llbracket A_a^{i_2} A_a^{i_3} \rrbracket$ with the planes $\llbracket B^{j_1} \rrbracket$, $\llbracket B^{j_2} \rrbracket$ and $\llbracket B^{j_3} \rrbracket$. In other words, this equation says that the intersection points of a line with three planes all lie along a line (see figure 4.6).

When projecting the three intersection points onto image plane C they still have to lie along a line. That is,

$$\begin{aligned}
0 &= (X_i^{j_a\top} \cdot C^{k_a})(X_i^{j_b\top} \cdot C^{k_b})(X_i^{j_c\top} \cdot C^{k_c}) C_{k_a} \wedge C_{k_b} \wedge C_{k_c} \\
\Longleftrightarrow \; 0 &= T_i^{j_a k_a} T_i^{j_b k_b} T_i^{j_c k_c} \llbracket C_{k_a} C_{k_b} C_{k_c} C_4 \rrbracket_c \\
&= \epsilon_{k_a k_b k_c} T_i^{j_a k_a} T_i^{j_b k_b} T_i^{j_c k_c} \\
&= \det(T_i^{jk})_{jk}
\end{aligned} \tag{4.48}$$

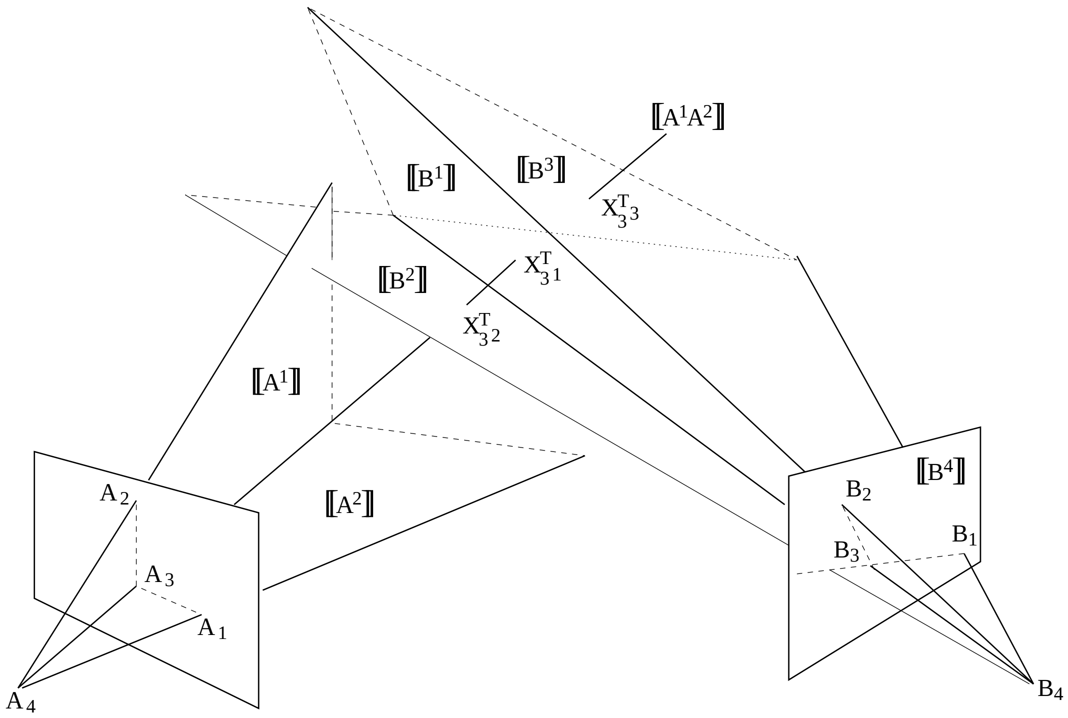

$\llbracket A^1A^2 \rrbracket$
$\llbracket B^1 \rrbracket$
$\llbracket B^3 \rrbracket$
X^T_{33}
$\llbracket B^2 \rrbracket$
X^T_{31}
X^T_{32}
$\llbracket A^1 \rrbracket$
$\llbracket A^2 \rrbracket$
A_2
A_3
A_1
A_4
B_2
$\llbracket B^4 \rrbracket$
B_1
B_3
B_4

Figure 4.6: This demonstrates the constraint from equation (4.47) for $i_2 = 1$, $i_3 = 2$ and $j_a = 1$, $j_b = 2$, $j_c = 3$. The figure also visualises the use of the dual bracket to describe planes and lines.

4.3.7 Relation between T and F

We mentioned before that the quadratic p-relations can be used to find constraints on T [32]. The equivalent expressions in GA are of the form

$$\llbracket B^1 B^2 \rrbracket \wedge \llbracket A^1 A^2 A^3 \rrbracket \wedge \llbracket B^1 B^2 C^1 \rrbracket = 0 \tag{4.49}$$

This expression is zero because $\llbracket B^1 B^2 \rrbracket \wedge \llbracket B^1 B^2 C^1 \rrbracket = 0$. This becomes obvious immediately from a geometric point of view: the intersection point of line $\llbracket B^1 B^2 \rrbracket$ with plane $\llbracket C^1 \rrbracket$, i.e. $\llbracket B^1 B^2 C^1 \rrbracket$, clearly lies on line $\llbracket B^1 B^2 \rrbracket$, and thus their outer product is zero.

In the following we will write $T_{i_1}^{jk\,\mathsf{XYZ}}$ to denote the trifocal tensor

$$T_{i_1}^{jk\,\mathsf{XYZ}} = \llbracket X^{i_2} X^{i_3} Y^j Z^k \rrbracket$$

We will similarly write $F_{i_1 j_1}^{\mathsf{XY}}$ to denote the bifocal tensor

$$F_{i_1 j_1}^{\mathsf{XY}} = \llbracket X^{i_2} X^{i_3} Y^{j_2} Y^{j_3} \rrbracket$$

If no superscripts are given then $T_i^{\;jk}$ and F_{ij} take on the same meaning as before. That is,

$$\begin{aligned} T_i^{\;jk} &\equiv T_i^{\;jk\,\mathsf{ABC}} && (4.50\text{a}) \\ F_{ij} &\equiv F_{ij}^{\mathsf{AB}} && (4.50\text{b}) \end{aligned}$$

We can obtain a constraint on T by expanding equation (4.49).

$$\begin{aligned} 0 &= [\![B^1B^2]\!]\wedge[\![A^1A^2A^3]\!]\wedge[\![B^1B^2C^1]\!] \\ &= \quad [\![A^1A^2B^1B^2]\!][\![B^1B^2A^3C^1]\!] \\ &\quad + [\![A^3A^1B^1B^2]\!][\![B^1B^2A^2C^1]\!] \\ &\quad + [\![A^2A^3B^1B^2]\!][\![B^1B^2A^1C^1]\!] \\ &= F_{33}T_3^{\;31\,\mathsf{BAC}} + F_{23}T_3^{\;21\,\mathsf{BAC}} + F_{13}T_3^{\;11\,\mathsf{BAC}} \\ &= F_{i3}T_3^{\;i1\,\mathsf{BAC}} \end{aligned} \tag{4.51}$$

Note that there is an implicit summation over i, because it is repeated as a (relative) superscript. Of course, we could have chosen different indices for the reciprocal B vectors and the reciprocal C vector. Therefore, we can obtain the following relation between the trifocal tensor and the bifocal tensor.

$$F_{ij}T_j^{\;ik\,\mathsf{BAC}} = 0 \tag{4.52}$$

Again there is an implicit summation over the i index but not over the j index. From this equation it follows that the three column vectors of the bifocal tensor give the three "left" null vectors of the three matrices $T_i^{\;\bullet\bullet}$, respectively. Equation (4.52) has two main uses: it can be used to find some epipoles of the trifocal tensor via equations (4.23) and (4.24), but it also serves to give more constraints on T since $\det F = 0$.

The columns of F may be found from equation (4.52) using, for example, an SVD. However, since the columns are found separately they will not in general be scaled consistently. Therefore, F found from equation (4.52) has only a limited use. Nonetheless, we can still find the correct left null vector of F, i.e. ε^i_{ab}, because each column is consistent in itself. Note also that, the determinant of F is still zero, since the rank of F cannot be changed by scaling its columns separately. We cannot use this F, though, to find the right null vector, i.e. ε^i_{ba}, or to check whether image points on planes A and B are images of the same world point. Finding a consistent F is not necessary to find the right null vector of F, as will be shown later on. Therefore, unless we need to find a bifocal tensor from T which we can use to check image point pair matches, a consistent F is not necessary. A consistent F can, however, be found as shown in the following.

We can find the bifocal tensor row-wise in the following way.

$$\begin{aligned} 0 &= \llbracket A^{i_2}A^{i_3}\rrbracket \wedge \llbracket B^1B^2B^3\rrbracket \wedge \llbracket A^{i_2}A^{i_3}C^k\rrbracket \\ &= F_{i_1 j}T_{i_1}^{jk} \end{aligned} \tag{4.53}$$

Knowing F row-wise and column-wise we can find a consistently scaled bifocal tensor. What remains is to find T^{BAC} from T. To do so we define the following intersection points in terms of the lines $T^{i_a j_a} \equiv \llbracket B^{i_a}C^{j_a}\rrbracket$(see equation (4.30a)).

$$\begin{aligned} p(i_a j_a, i_b j_b) &\equiv (A_4 \wedge T^{i_a j_a}) \vee T^{i_b j_b} \\ &= \Big\llbracket \Big\llbracket A_4 \llbracket B^{i_a}C^{j_a}\rrbracket \Big\rrbracket \Big\llbracket \llbracket B^{i_b}C^{j_b}\rrbracket \Big\rrbracket \Big\rrbracket \\ &= \Big\llbracket \Big(A_4 \cdot \Big\llbracket \llbracket B^{i_a}C^{j_a}\rrbracket \Big\rrbracket \Big) B^{i_b}C^{j_b} \Big\rrbracket \\ &= \Big\llbracket \Big(A_4 \cdot (B^{i_a} \wedge C^{j_a})\Big) B^{i_b}C^{j_b} \Big\rrbracket \\ &= \Big\llbracket \quad (A_4 \cdot B^{i_a})C^{j_a}B^{i_b}C^{j_b} \\ &\qquad\quad -(A_4 \cdot C^{j_a})B^{i_a}B^{i_b}C^{j_b} \Big\rrbracket \\ &= \varepsilon^{i_a}_{ba}\llbracket C^{j_a}B^{i_b}C^{j_b}\rrbracket + \varepsilon^{j_a}_{ca}\llbracket B^{i_a}C^{j_b}B^{i_b}\rrbracket \end{aligned} \tag{4.54}$$

Two useful special cases are

$$p(i_1 j, i_2 j) = \varepsilon^{j}_{ca}\llbracket B^{i_1}C^{j}B^{i_2}\rrbracket \tag{4.55a}$$

$$p(i j_1, i j_2) = \varepsilon^{i}_{ba}\llbracket C^{j_1}B^{i}C^{j_2}\rrbracket \tag{4.55b}$$

The projection of $p(i_1 j, i_2 j)$ onto image plane A, denoted by $p_a(i_1 j, i_2 j)$ gives

$$\begin{aligned} p_a(i_2 k, i_3 k) &= \varepsilon^{k}_{ca}\Big(A^j \cdot \llbracket B^{i_2}C^{k}B^{i_3}\rrbracket\Big)A_j \\ &= \varepsilon^{k}_{ca}\llbracket A^j B^{i_2}C^{k}B^{i_3}\rrbracket A_j \\ &= -\varepsilon^{k}_{ca}\llbracket B^{i_2}B^{i_3}A^{j}C^{k}\rrbracket A_j \\ &= -\varepsilon^{k}_{ca}T_{i_1}^{jk\,\mathsf{BAC}}A_j \end{aligned} \tag{4.56}$$

We can also calculate $p_a(j_a k_a, j_b k_b)$ by immediately using the projections of the T^{jk} onto image plane A (see equation (4.30b)). That is,

$$\begin{aligned} p_a(j_a k_a, j_b k_b) &= (A_4 \wedge T_a^{i_a j_a}) \vee T_a^{i_b j_b} \\ &= T_{i_a}^{j_a k_a}T_{i_b}^{j_b k_b}(A_4 \wedge L_a^{i_a}) \vee L_a^{i_b} \\ &= T_{i_a}^{j_a k_a}T_{i_b}^{j_b k_b}(A_4 \wedge \llbracket A_a^{i_a}A_a^{4}\rrbracket) \vee \llbracket A_a^{i_b}A_a^{4}\rrbracket \\ &= T_{i_a}^{j_a k_a}T_{i_b}^{j_b k_b}\Big\llbracket \Big(A_4 \cdot (A_a^{i_a} \wedge A_a^{4})\Big) A_a^{i_b}A_a^{4} \Big\rrbracket \\ &\simeq T_{i_a}^{j_a k_a}T_{i_b}^{j_b k_b}\llbracket A_a^{i_a}A_a^{i_b}A_a^{4}\rrbracket \end{aligned} \tag{4.57}$$

From the definition of the dual bracket we have

$$A_{i_3} = [\![A_a^{i_1} A_a^{i_2} A_a^4]\!]_a$$

Therefore, from equation (4.57) we find

$$p_a(j_1k, j_2k) \simeq (T_{i_1}^{\;j_1k} T_{i_2}^{\;j_2k} - T_{i_2}^{\;j_1k} T_{i_1}^{\;j_2k}) A_{i_3} \tag{4.58}$$

Equating this with equation (4.56) gives

$$T_{j_3}^{\;i_3k\,\mathsf{BAC}} \simeq (\varepsilon_{ca}^k)^{-1} (T_{i_1}^{\;j_1k} T_{i_2}^{\;j_2k} - T_{i_2}^{\;j_1k} T_{i_1}^{\;j_2k}) \tag{4.59}$$

Since ε_{ca}^k can be found from T (as will be shown later) we can find T^{BAC} from T up to an overall scale. Equation (4.59) may also be written in terms of the standard cross product.

$$T_{j_3}^{\;\bullet k\,\mathsf{BAC}} \simeq (\varepsilon_{ca}^k)^{-1} (T_{\bullet}^{\;j_1k} \times T_{\bullet}^{\;j_2k}) \tag{4.60}$$

Had we used equation (4.55b) instead of equation (4.55a) in the previous calculation, we would have obtained the following relation.

$$T_{k_3}^{\;ji_3\,\mathsf{CBA}} \simeq (\varepsilon_{ba}^j)^{-1} (T_{i_1}^{\;jk_1} T_{i_2}^{\;jk_2} - T_{i_2}^{\;jk_1} T_{i_1}^{\;jk_2}) \tag{4.61}$$

Or, in terms of the standard cross product

$$T_{k_3}^{\;j\bullet\,\mathsf{CBA}} \simeq (\varepsilon_{ba}^j)^{-1} (T_{\bullet}^{\;jk_1} \times T_{\bullet}^{\;jk_2}) \tag{4.62}$$

Hence, we can also obtain T^{CBA} from T up to an overall scale. Note that since

$$\begin{aligned} T_{i_1}^{\;jk\,\mathsf{ABC}} &= [\![A^{i_2} A^{i_3} B^j C^k]\!] \\ &= -[\![A^{i_2} A^{i_3} C^k B^j]\!] \\ &= -T_{i_1}^{\;kj\,\mathsf{ACB}} \end{aligned} \tag{4.63}$$

we have found *all* possible trifocal tensors for a particular camera setup from T.

Equations (4.60) and (4.62) simply express that the projections of the intersection points between some lines onto image plane A are the same as the intersection points between the projections of the same lines onto image plane A. This implies that independent of the intersection points, i.e. the components of $T_i^{\;jk}$, equations (4.60) and (4.62) will always give a self-consistent tensor, albeit not necessarily one that expresses the correct camera geometry.

4.3.8 Second Order Constraints

There are more constraints on T which we will call "second order" because they are products of determinants of components of T. Their derivation is more involved and can be found in [48] and [49]. Here we will only state the results. These constraints may be used to check the self-consistency of T when it is calculated via a non-linear method.

$$\begin{aligned} 0 = & \quad |T_a^{j_a k_a} T_a^{j_b k_a} T_a^{j_a k_b}|\; |T_a^{j_b k_b} T_a^{j_a k_c} T_a^{j_b k_c}| \\ - & \quad |T_a^{j_a k_a} T_a^{j_b k_a} T_a^{j_b k_b}|\; |T_a^{j_a k_b} T_a^{j_a k_c} T_a^{j_b k_c}| \end{aligned} \tag{4.64}$$

$$\begin{aligned} 0 = & \quad |T_a^{j_a k_a} T_a^{j_a k_b} T_a^{j_b k_a}|\; |T_a^{j_b k_b} T_a^{j_c k_a} T_a^{j_c k_b}| \\ - & \quad |T_a^{j_a k_a} T_a^{j_a k_b} T_a^{j_b k_b}|\; |T_a^{j_b k_a} T_a^{j_c k_a} T_a^{j_c k_b}| \end{aligned} \tag{4.65}$$

$$\begin{aligned} 0 = & \quad |T_a^{i_a j_a} T_a^{i_b j_a} T_a^{i_a j_b}|\; |T_a^{i_b j_b} T_a^{i_a j_b} T_a^{i_b j_c}| \\ - & \quad |T_a^{i_a j_a} T_a^{i_b j_a} T_a^{i_b j_b}|\; |T_a^{i_a j_b} T_a^{i_a j_c} T_a^{i_b j_b}| \end{aligned} \tag{4.66}$$

Where the determinants are to be interpreted as

$$|T_a^{j_a k_a} T_a^{j_b k_a} T_a^{j_a k_b}| = \det(T_{i_a}^{\;j_a k_a}, T_{i_b}^{\;j_b k_a}, T_{i_c}^{\;j_a k_b})_{i_a i_b i_c}$$

4.3.9 Epipoles

The epipoles of T can be found indirectly via the relation of bifocal tensors to T (e.g. equation (4.52)). Also recall that the right null vector of some F_{ij}^{XY} is ε_{yx}^{j}, whereas the left null vector is ε_{xy}^{i} (equations (4.23) and (4.24)). From equation (4.53) we know that

$$F_{ij}\; T_i^{\;jk} = 0$$

When calculating F from this equation, we cannot guarantee that the rows are scaled consistently. Nevertheless, this does not affect the right null space of F. Hence, we can find ε_{ba}^{j} from this F. In the following we will list the necessary relations to find all epipoles of T.

$$\begin{aligned} 0 &= [\![A^{i_2} A^{i_3}]\!] \wedge [\![B^1 B^2 B^3]\!] \wedge [\![A^{i_2} A^{i_3} C^k]\!] \\ &= F_{i_1 j}\; T_{i_1}^{\;jk} \qquad\qquad\qquad\qquad \rightarrow \varepsilon_{ba}^{j} \end{aligned} \tag{4.67a}$$

$$\begin{aligned} 0 &= [\![A^{i_2} A^{i_3}]\!] \wedge [\![C^1 C^2 C^3]\!] \wedge [\![A^{i_2} A^{i_3} B^j]\!] \\ &= F_{i_1 k}^{AC}\; T_{i_1}^{\;jk} \qquad\qquad\qquad\qquad \rightarrow \varepsilon_{ca}^{k} \end{aligned} \tag{4.67b}$$

$$\begin{aligned} 0 &= \llbracket B^{i_2} B^{i_3} \rrbracket \wedge \llbracket A^1 A^2 A^3 \rrbracket \wedge \llbracket B^{i_2} B^{i_3} C^k \rrbracket \\ &= F^{\mathsf{BA}}_{i_1 j} \, T_{i_1}^{jk\,\mathsf{BAC}} \qquad\qquad \rightarrow \varepsilon^j_{ab} \end{aligned} \tag{4.68a}$$

$$\begin{aligned} 0 &= \llbracket B^{i_2} B^{i_3} \rrbracket \wedge \llbracket C^1 C^2 C^3 \rrbracket \wedge \llbracket B^{i_2} B^{i_3} A^j \rrbracket \\ &= F^{\mathsf{BC}}_{i_1 k} \, T_{i_1}^{jk\,\mathsf{BAC}} \qquad\qquad \rightarrow \varepsilon^k_{cb} \end{aligned} \tag{4.68b}$$

$$\begin{aligned} 0 &= \llbracket C^{i_2} C^{i_3} \rrbracket \wedge \llbracket A^1 A^2 A^3 \rrbracket \wedge \llbracket C^{i_2} C^{i_3} B^j \rrbracket \\ &= F^{\mathsf{CA}}_{i_1 k} \, T_{i_1}^{jk\,\mathsf{CBA}} \qquad\qquad \rightarrow \varepsilon^k_{ac} \end{aligned} \tag{4.69a}$$

$$\begin{aligned} 0 &= \llbracket C^{i_2} C^{i_3} \rrbracket \wedge \llbracket B^1 B^2 B^3 \rrbracket \wedge \llbracket C^{i_2} C^{i_3} A^k \rrbracket \\ &= F^{\mathsf{CB}}_{i_1 j} \, T_{i_1}^{jk\,\mathsf{CBA}} \qquad\qquad \rightarrow \varepsilon^j_{bc} \end{aligned} \tag{4.69b}$$

By $\rightarrow \varepsilon^j_{xy}$ we denote the epipole that can be found from the respective relation. Note that since

$$\begin{aligned} F^{\mathsf{XY}}_{i_1 j_1} &= \llbracket X^{i_2} X^{i_3} Y^{j_2} Y^{j_3} \rrbracket \\ &= \llbracket Y^{j_2} Y^{j_3} X^{i_2} X^{i_3} \rrbracket \\ &= F^{\mathsf{YX}}_{j_1 i_1} \end{aligned} \tag{4.70}$$

we have also found all fundamental matrices.

4.4 The Quadfocal Tensor

4.4.1 Derivation

Let L be a line in $\mathcal{P}^3$ and let $\{A_\mu\}$, $\{B_\mu\}$, $\{C_\mu\}$ and $\{D_\mu\}$ define four cameras A, B, C and D, respectively (see figure 4.7). The projection of L onto the image planes of these four cameras is

$$L \xrightarrow{A} L_A = L \cdot L^a_i \, L^i_a = \lambda^a_i \, L^i_a \tag{4.71a}$$

$$L \xrightarrow{B} L_B = L \cdot L^b_i \, L^i_b = \lambda^b_i \, L^i_b \tag{4.71b}$$

$$L \xrightarrow{C} L_C = L \cdot L^c_i \, L^i_c = \lambda^c_i \, L^i_c \tag{4.71c}$$

$$L \xrightarrow{D} L_D = L \cdot L^d_i \, L^i_d = \lambda^d_i \, L^i_d \tag{4.71d}$$

The intial line L can be recovered from these projections by intersecting any two of the planes $(L_A \wedge A_4)$, $(L_B \wedge B_4)$, $(L_C \wedge C_4)$ and $(L_D \wedge D_4)$. For example,

$$L \simeq (L_A \wedge A_4) \vee (L_B \wedge B_4) \simeq (L_C \wedge C_4) \vee (L_D \wedge D_4) \tag{4.72}$$

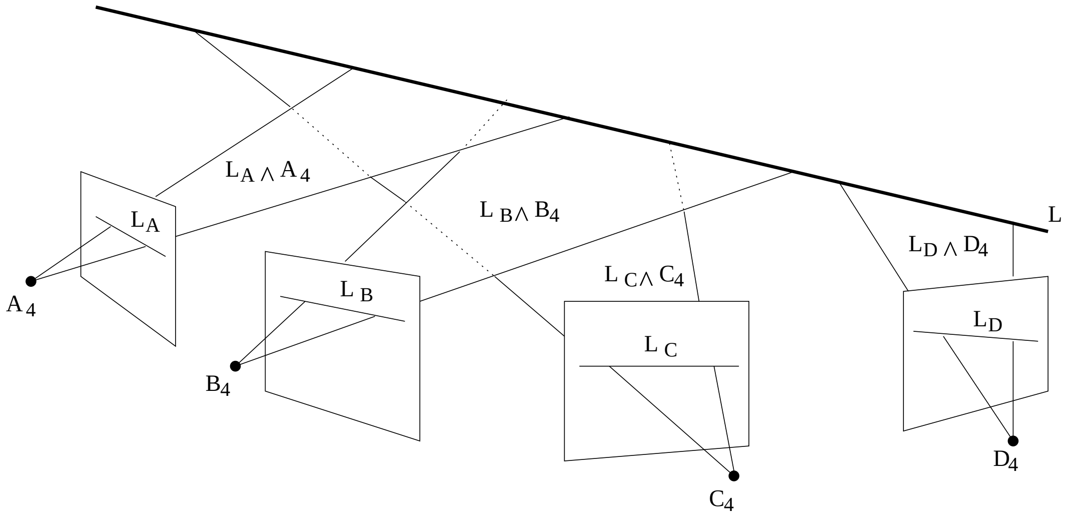


Figure 4.7: Line projected onto four image planes.

Therefore,

$$\begin{aligned} 0 &= \Big[\!\!\Big[\Big((L_A \wedge A_4) \vee (L_B \wedge B_4)\Big)\Big((L_C \wedge C_4) \vee (L_D \wedge D_4)\Big)\Big]\!\!\Big] \\ &= \lambda_i^a \lambda_j^b \lambda_k^c \lambda_l^d \;\Big[\!\!\Big[\quad \Big((L_a^i \wedge A_4) \vee (L_b^j \wedge B_4)\Big) \\ &\qquad\qquad\qquad \Big((L_c^k \wedge C_4) \vee (L_d^l \wedge D_4)\Big)\Big]\!\!\Big] \\ &= \lambda_i^a \lambda_j^b \lambda_k^c \lambda_l^d \;\Big[\!\!\Big[\Big([\![A^i]\!] \vee [\![B^j]\!]\Big)\Big([\![C^k]\!] \vee [\![D^l]\!]\Big)\Big]\!\!\Big] \\ &= \lambda_i^a \lambda_j^b \lambda_k^c \lambda_l^d \;[\![A^i B^j C^k D^l]\!] \end{aligned} \tag{4.73}$$

Therefore, a *quadfocal tensor* may be defined as

$$Q^{ijkl} = [\![A^i B^j C^k D^l]\!] \tag{4.74}$$

If the quadfocal tensor is contracted with the homogeneous line coordinates of the projections of one line onto the four camera image planes, the result is zero. In this way the quadfocal tensor encodes the relative orientation of the four camera image planes. However, note that contracting the quadfocal tensor with the line coordinates of the projection of one line onto only three image planes gives a zero vector. This follows directly from the following considerations. Let $L = A_1 \wedge A_2$ be a line and $P = A_1 \wedge A_2 \wedge A_3$ be a plane in $\mathcal{P}^3$. That is, line L lies in plane P. Then we find

$$L \cdot [\![P]\!] = A_1 \, [\![A_2 P]\!] - A_2 \, [\![A_1 P]\!] = 0. \tag{4.75}$$

For the quadfocal tensor we have, for example,

$$\begin{aligned}\lambda_i^a\lambda_j^b\lambda_k^c Q^{ijkl} &= \lambda_i^a\lambda_j^b\lambda_k^c [\![A^iB^jC^k]\!]\cdot D^l \\ &\simeq \Big(L\cdot(\lambda_k^c C^k)\Big)\cdot D^l\,, \qquad L \simeq \lambda_i^a\lambda_j^b[\![A^iB^j]\!] \\ &= 0.\end{aligned} \tag{4.76}$$

L is the line whose images on image planes A, B and C have coordinates $\{\lambda_i^a\}$, $\{\lambda_j^b\}$ and $\{\lambda_k^c\}$, respectively. Hence, L lies on plane $[\![\lambda_k^c C^k]\!]$, and thus it follows from equation (4.75) that $L\cdot\,(\lambda_k^c C^k) = 0$. This also shows that the quadfocal tensor does not add any new information to what can be known from the trifocal tensor, since the quadfocal tensor simply relates any three image planes out of a group of four.

The form for Q given in equation (4.74) can be shown to be equivalent to the form given by Heyden in [32]. From equation (4.74) it is also immediately clear that changing the order of the reciprocal vectors at most changes the overall sign of Q.

4.4.2 Transferring Lines

If the image of a line is known on two image planes, then the quadfocal tensor can be used to find its image on the other two image planes. This can be achieved through a somewhat indirect route. Let L be a line projected onto image planes A and B with coordinates $\{\lambda_i^a\}$ and $\{\lambda_j^b\}$, respectively. Then we know that

$$L \simeq \lambda_i^a\lambda_j^b[\![A^iB^j]\!] \tag{4.77}$$

Therefore, we can define three points $\{X_L^k\}$ that lie on L as

$$\begin{aligned} X_L^k &\equiv \lambda_i^a\lambda_j^b([\![A^iB^j]\!] \vee [\![C^k]\!]) \\ &= \lambda_i^a\lambda_j^b[\![A^iB^jC^k]\!] \end{aligned} \tag{4.78}$$

The projections of the $\{X_L^k\}$ onto image plane D, denoted by $\{X_{L_d}^k\}$ are given by

$$\begin{aligned} X_{L_d}^k &\equiv X_L^k\cdot D^l\,D_l \\ &= \lambda_i^a\lambda_j^b[\![A^iB^jC^k]\!]\cdot D^l\,D_l \\ &= \lambda_i^a\lambda_j^b[\![A^iB^jC^kD^l]\!]D_l \\ &= \lambda_i^a\lambda_j^b Q^{ijkl}D_l \end{aligned} \tag{4.79}$$

From the points $\{X^k_{L_d}\}$ the projection of line L onto image plane D can be recovered.

4.4.3 Rank of Q

The form for the quadfocal tensor as given in equation (4.74) may be expanded in a number of ways. For example,

$$\begin{aligned} Q^{i_1 jkl} &= (A_{i_2} \wedge A_{i_3} \wedge A_4) \cdot (B^j \wedge C^k \wedge D^l) \\ &= \quad \varepsilon^{j}_{ba} \Big[K^{k}_{i_3}{}^{\mathsf{c}} K^{l}_{i_2}{}^{\mathsf{d}} - K^{l}_{i_3}{}^{\mathsf{d}} K^{k}_{i_2}{}^{\mathsf{c}} \Big] \\ &\quad - \varepsilon^{k}_{ca} \Big[K^{j}_{i_3}{}^{\mathsf{b}} K^{l}_{i_2}{}^{\mathsf{d}} - K^{l}_{i_3}{}^{\mathsf{d}} K^{j}_{i_2}{}^{\mathsf{b}} \Big] \\ &\quad + \varepsilon^{l}_{da} \Big[K^{j}_{i_3}{}^{\mathsf{b}} K^{k}_{i_2}{}^{\mathsf{c}} - K^{k}_{i_3}{}^{\mathsf{c}} K^{j}_{i_2}{}^{\mathsf{b}} \Big] \end{aligned} \tag{4.80}$$

In terms of the standard cross product this may be written as

$$Q^{\bullet jkl} = U^j_b (K^{k}_{\bullet}{}^{\mathsf{c}} \times K^{l}_{\bullet}{}^{\mathsf{d}}) - U^k_c (K^{j}_{\bullet}{}^{\mathsf{b}} \times K^{l}_{\bullet}{}^{\mathsf{d}}) + U^l_d (K^{j}_{\bullet}{}^{\mathsf{b}} \times K^{k}_{\bullet}{}^{\mathsf{c}}) \tag{4.81}$$

From equation (4.80) it becomes clear that, as for the trifocal tensor, the transformation $A_i \mapsto s(A_i + t_i A_4)$ leaves Q unchanged up to an overall scale.

Let $P = B_4 \wedge C_4 \wedge D_4$. As for the trifocal tensor case, define a basis $\{A'_i\}$ for image plane A by

$$A'_i = (A_i \wedge A_4) \vee P \tag{4.82}$$

All the $\{A'_i\}$ lie on plane P, that is they lie on the plane formed by B_4, C_4 and D_4. Therefore, $K^{b\prime}_{j_i} = A'_i \cdot B^j$, $K^{c\prime} = A'_i \cdot C^k$ and $K^{d\prime} = A'_i \cdot D^l$ are of rank 2. As was shown previously, this is the minimum rank the camera matrices can have. Hence, forming Q with the $\{A'_i\}$ should yield its rank. However, it is not immediately obvious from equation (4.80) what the rank of Q is when substituting the $\{A'_i\}$ for the $\{A_i\}$. A more yielding decomposition of Q is achieved by expanding equation (4.82).

$$\begin{aligned} A'_i &= (A_i \wedge A_4) \vee P \\ &\simeq [\![A_i A_4]\!] \cdot (B_4 \wedge C_4 \wedge D_4) \\ &= [\![A_i A_4 B_4 C_4]\!] D_4 - [\![A_i A_4 B_4 D_4]\!] C_4 + [\![A_i A_4 C_4 D_4]\!] B_4 \\ &= \alpha^1_i B_4 + \alpha^2_i C_4 + \alpha^3_i D_4 \end{aligned} \tag{4.83}$$

where the $\{\alpha_i^j\}$ are defined accordingly. Furthermore,

$$A'_{i_1} \wedge A'_{i_2} = \lambda^1_{i_3}\, C_4 \wedge D_4 \,+ \lambda^2_{i_3}\, D_4 \wedge B_4 \,+ \lambda^3_{i_3}\, B_4 \wedge C_4 \tag{4.84}$$

with $\lambda^{j_3}_{i_3} = \alpha^{j_1}_{i_1}\alpha^{j_2}_{i_2} - \alpha^{j_1}_{i_2}\alpha^{j_2}_{i_1}$. Equation (4.80) may also be written as

$$\begin{aligned} Q^{i_1 jkl} &= (A_{i_2} \wedge A_{i_3} \wedge A_4) \cdot (B^j \wedge C^k \wedge D^l) \\ &= \quad U^j_b \Big[(A'_{i_2} \wedge A'_{i_3}) \cdot (C^k \wedge D^l) \Big] \\ &\quad - U^k_c \Big[(A'_{i_2} \wedge A'_{i_3}) \cdot (B^j \wedge D^l) \Big] \\ &\quad + U^l_d \Big[(A'_{i_2} \wedge A'_{i_3}) \cdot (B^j \wedge C^k) \Big] \end{aligned} \tag{4.85}$$

From equation (4.84) it then follows

$$\begin{aligned} (A'_{i_2} \wedge A'_{i_3}) \cdot (C^k \wedge D^l) &= \quad \lambda^1_{i_1}\, D_4 \cdot C^k\ C_4 \cdot D^l \\ &\quad - \lambda^2_{i_1}\, D_4 \cdot C^k\ B_4 \cdot D^l \\ &\quad - \lambda^3_{i_1}\, B_4 \cdot C^k\ C_4 \cdot D^l \end{aligned} \tag{4.86a}$$

$$\begin{aligned} (A'_{i_2} \wedge A'_{i_3}) \cdot (B^j \wedge D^l) &= \quad \lambda^1_{i_1}\, D_4 \cdot B^j\ C_4 \cdot D^l \\ &\quad - \lambda^2_{i_1}\, D_4 \cdot B^j\ B_4 \cdot D^l \\ &\quad + \lambda^3_{i_1}\, C_4 \cdot B^j\ B_4 \cdot D^l \end{aligned} \tag{4.86b}$$

$$\begin{aligned} (A'_{i_2} \wedge A'_{i_3}) \cdot (B^j \wedge C^k) &= - \lambda^1_{i_1}\, C_4 \cdot B^j\ D_4 \cdot C^k \\ &\quad - \lambda^2_{i_1}\, D_4 \cdot B^j\ B_4 \cdot C^k \\ &\quad + \lambda^3_{i_1}\, C_4 \cdot B^j\ B_4 \cdot C^k \end{aligned} \tag{4.86c}$$

Each of these three equations has a linear combination of three rank 1, 3-valence tensors on its right hand side. Furthermore, none of the rank 1, 3-valence tensors from one equation is repeated in any of the others. Therefore, substituting equations (4.86) into equation (4.85) gives a decomposition of Q in terms of 9 rank 1 tensors. Since this is a minimal decomposition, Q is of rank 9.

4.4.4 Degrees of Freedom of Q

Substituting equations (4.86) back into equation (4.85) gives

$$\begin{aligned} Q^{ijkl} = &\quad \varepsilon^j_{ba} \Big[\lambda^1_i \varepsilon^k_{cd} \varepsilon^l_{dc} - \lambda^2_i \varepsilon^k_{cd} \varepsilon^l_{db} + \lambda^3_i \varepsilon^k_{cb} \varepsilon^l_{dc} \Big] \\ &- \varepsilon^k_{ca} \Big[\lambda^1_i \varepsilon^j_{bd} \varepsilon^l_{dc} - \lambda^2_i \varepsilon^j_{bd} \varepsilon^l_{db} + \lambda^3_i \varepsilon^j_{bc} \varepsilon^l_{db} \Big] \\ &+ \varepsilon^l_{da} \Big[\lambda^1_i \varepsilon^j_{bc} \varepsilon^k_{cd} - \lambda^2_i \varepsilon^j_{bd} \varepsilon^k_{cb} + \lambda^3_i \varepsilon^j_{bc} \varepsilon^k_{cb} \Big] \end{aligned} \tag{4.87}$$

This decomposition of Q has $9\times3+3\times3-1=35$ DOF. The general formula for the DOF of Q gives $4\times11-15=29$ DOF. Therefore the parameterisation of Q in equation (4.87) is overdetermined. However, it will still give a self-consistent Q.

4.4.5 Constraints on Q

The constraints on Q can again be found very easily through geometric considerations. Let the points $\{X_Q^{ijk}\}$ be defined as

$$X_Q^{ijk} \equiv [\![A^i B^j C^k]\!] \tag{4.88}$$

A point X_Q^{ijk} can be interpreted as the intersection of line $[\![A^i B^j]\!]$ with plane $[\![C^k]\!]$. Therefore,

$$X_Q^{ijk_a} \wedge X_Q^{ijk_b} \wedge X_Q^{ijk_c} = 0 \tag{4.89}$$

because the three intersection points $X_Q^{ijk_a}$, $X_Q^{ijk_b}$ and $X_Q^{ijk_c}$ lie along line $[\![A^i B^j]\!]$. Hence, also their projections onto an image plane have to lie along a line. Thus, projecting the intersection points onto an image plane D we have

$$\begin{aligned} 0 &= (X_Q^{ijk_a}\cdot D^{l_a})\,(X_Q^{ijk_b}\cdot D^{l_b})\,(X_Q^{ijk_c}\cdot D^{l_c}) \\ &\quad (D_{l_a}\wedge D_{l_b}\wedge D_{l_c}) \\ \Longleftrightarrow\ 0 &= Q^{ijk_a l_a}\,Q^{ijk_b l_b}\,Q^{ijk_c l_c}[\![D_{l_a}D_{l_b}D_{l_c}D_4]\!]_d \\ &= \epsilon_{l_a l_b l_c}\,Q^{ijk_a l_a}\,Q^{ijk_b l_b}\,Q^{ijk_c l_c} \\ &= \det(Q^{ijkl})_{kl} \end{aligned} \tag{4.90}$$

Similarly, this type of constraint may be shown for every pair of indices. We therefore get the following constraints on Q.

$$\begin{aligned} &\det(Q^{ijkl})_{ij}=0;\quad \det(Q^{ijkl})_{ik}=0;\quad \det(Q^{ijkl})_{il}=0 \\ &\det(Q^{ijkl})_{jk}=0;\quad \det(Q^{ijkl})_{jl}=0;\quad \det(Q^{ijkl})_{kl}=0 \end{aligned} \tag{4.91}$$

4.4.6 Relation between Q and T

We can find the relation between Q and T via the method employed to find the relation between T and F. For example,

$$\begin{aligned} 0 &= [\![A^1 A^2 A^3]\!] \wedge [\![B^j C^k D^l]\!] \wedge [\![B^j C^k]\!] \\ &= \textstyle\sum_{i_1} \Big([\![A^{i_1} B^j C^k D^l]\!] [\![A^{i_2} A^{i_3} B^j C^k]\!] \Big) \\ &= Q^{ijkl} T_i^{\;jk} \end{aligned} \tag{4.92}$$

Similarly, equations for the other possible trifocal tensors can be found. Because of the trifocal tensor symmetry detailed in equation (4.63) all trifocal tensors may be evaluated from the following set of equations.

$$\begin{array}{lll} Q^{ijkl}\, T_i^{\;jk\,\mathsf{ABC}} = 0; & Q^{ijkl}\, T_i^{\;jl\,\mathsf{ABD}} = 0; & Q^{ijkl}\, T_i^{\;kl\,\mathsf{ACD}} = 0 \\ Q^{ijkl}\, T_j^{\;ik\,\mathsf{BAC}} = 0; & Q^{ijkl}\, T_j^{\;il\,\mathsf{BAD}} = 0; & Q^{ijkl}\, T_j^{\;kl\,\mathsf{BCD}} = 0 \\ Q^{ijkl}\, T_k^{\;ij\,\mathsf{CAB}} = 0; & Q^{ijkl}\, T_k^{\;il\,\mathsf{CAD}} = 0; & Q^{ijkl}\, T_k^{\;jl\,\mathsf{CBD}} = 0 \\ Q^{ijkl}\, T_l^{\;ij\,\mathsf{DAB}} = 0; & Q^{ijkl}\, T_l^{\;ik\,\mathsf{DAC}} = 0; & Q^{ijkl}\, T_l^{\;jk\,\mathsf{DBC}} = 0 \end{array} \tag{4.93}$$

Note that the trifocal tensors found in this way will not be of consistent scale. To fix the scale we start by defining intersection points

$$\begin{aligned} X^{jkl}_{BCD} &\equiv \Big[A_4 \wedge [\![B^j C^k]\!] \Big] \vee [\![C^k D^l]\!] \\ &\simeq \varepsilon^k_{ca} [\![B^j C^k D^l]\!] \end{aligned} \tag{4.94}$$

Projecting these points onto image plane A gives

$$\begin{aligned} X^{jkl}_{BCD_a} &\equiv X^{jkl}_{BCD} \cdot A^i A_i \\ &\simeq \varepsilon^k_{ca} [\![B^j C^k D^l]\!] \cdot A^i A_i \\ &\simeq \varepsilon^k_{ca} [\![A^i B^j C^k D^l]\!] A_i \\ &= \varepsilon^k_{ca} Q^{ijkl} A_i \end{aligned} \tag{4.95}$$

But we could have also arrived at an expression for $X^{jkl}_{BCD_a}$ via

$$\begin{aligned} X^{jkl}_{BCD_a} &\simeq \Big([\![B^j C^k]\!] \cdot L^a_{i_a} \Big) \Big([\![C^k D^l]\!] \cdot L^a_{i_b} \Big) \Big[A_4 \wedge L^{i_a}_a \Big] \vee L^{i_b}_a \\ &\simeq \Big(T_{i_1}^{\;jk\,\mathsf{ABC}} T_{i_2}^{\;kl\,\mathsf{ACD}} - T_{i_2}^{\;jk\,\mathsf{ABC}} T_{i_1}^{\;kl\,\mathsf{ACD}} \Big) A_{i_3} \end{aligned} \tag{4.96}$$

Equating this with equation (4.95) gives

$$T_{i_1}^{jk\,\mathsf{ABC}} T_{i_2}^{kl\,\mathsf{ACD}} - T_{i_2}^{jk\,\mathsf{ABC}} T_{i_1}^{kl\,\mathsf{ACD}} \simeq \varepsilon_{ca}^{k} Q^{i_3jkl} \tag{4.97}$$

This equation may be expressed more concisely in terms of the standard cross product.

$$T_{\bullet}^{jk\,\mathsf{ABC}} \times T_{\bullet}^{kl\,\mathsf{ACD}} \simeq \varepsilon_{ca}^{k} Q^{\bullet jkl} \tag{4.98}$$

Furthermore, from the intersection points

$$X_{CBD}^{kjl} \equiv \left[A_4 \wedge [\![C^k B^j]\!]\right] \vee [\![B^j D^l]\!]$$

and their projections onto image plane A we get

$$T_{\bullet}^{jk\,\mathsf{ABC}} \times T_{\bullet}^{jl\,\mathsf{ABD}} \simeq \varepsilon_{ba}^{j} Q^{\bullet jkl} \tag{4.99}$$

We can now find the correct scales for T^{ABC} by demanding that

$$\frac{T_{i_1}^{jk\,\mathsf{ABC}} T_{i_2}^{kl\,\mathsf{ACD}} - T_{i_2}^{jk\,\mathsf{ABC}} T_{i_1}^{kl\,\mathsf{ACD}}}{Q^{i_3jkl}} = \phi \tag{4.100}$$

for all j while keeping i_1, k and l constant, where ϕ is some scalar. Furthermore, we know that

$$\frac{T_{i_1}^{jk\,\mathsf{ABC}} T_{i_2}^{jl\,\mathsf{ABD}} - T_{i_2}^{jk\,\mathsf{ABC}} T_{i_1}^{jl\,\mathsf{ABD}}}{Q^{i_3jkl}} = \phi \tag{4.101}$$

for all k while keeping i_1, k and l constant, where ϕ is some different scalar. Equations (4.100) and (4.101) together fix the scales of T^{ABC} completely. Note that we do not have to know the epipoles ε_{ca}^{k} and ε_{ba}^{j}.

Similarly, all the other trifocal tensors can be found. These in turn can be used to find the fundamental matrices and the epipoles.

4.5 Reconstruction and the Trifocal Tensor

In the following we will investigate a computational aspect of the trifocal tensor. In particular we are interested in the 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 [26]. 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 [39] that the trifocal tensor in the epipolar basis has only 7 non-zero components[3]. 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

[3] 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.

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:

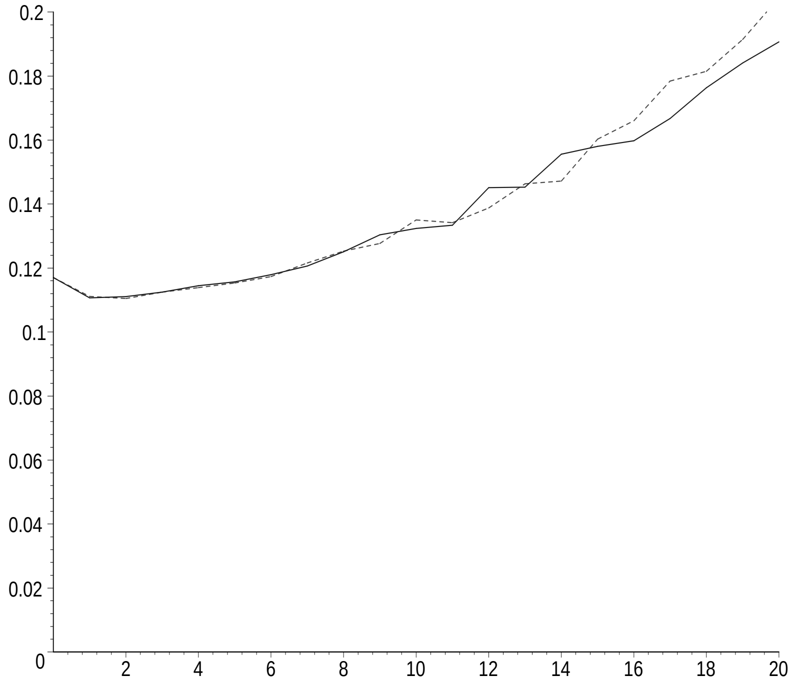


Figure 4.8: Mean distance between original points and reconstructed points in arbitrary units as a function of mean Gaussian error in pixels introduced by the cameras. The solid line shows the values using the MinFact algorithm, and the dashed line the values for the 7pt algorithm.

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 [26].

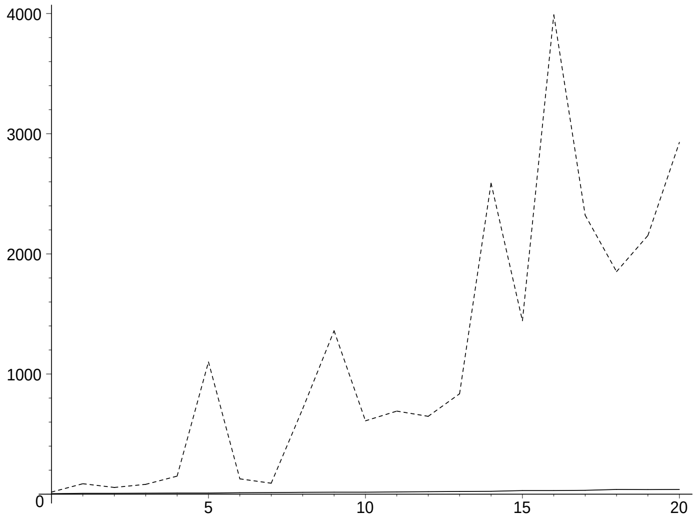


Figure 4.9: Mean difference between elements of calculated and true tensors in percent. Tensors are scaled previously, so that they are equal in one element. Solid line shows values for trifocal tensor calculated with 7pt algorithm, and dashed line shows values for trifocal tensor calculated with MinFact algorithm.

4. The points are reconstructed using a version of what is called "Method 3" in [56] and [57] adapted for three views. This uses a SVD to solve for the homogeneous reconstructed point algebraically using a set of camera matrices. In [56] and [57] 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 4.8 shows the mean distance between the original points and the

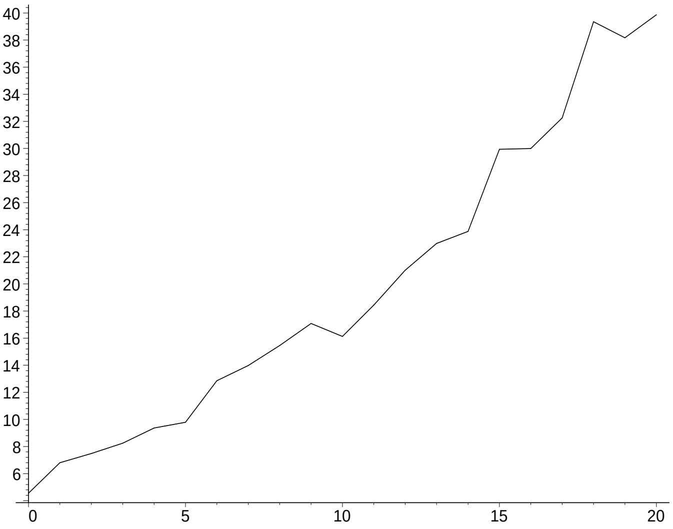


Figure 4.10: Mean difference between elements of true trifocal tensor and trifocal tensor calculated with 7pt algorithm in percent. Tensors are scaled previously, so that they are equal in one element. (This figure shows the solid line of figure 4.9 enlarged.)

reconstructed points in 3D-space in some arbitrary units[4], 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 4.9 and 4.10 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 4.9 compares the trifocal tensors found with the 7pt and the MinFact algorithms. 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 (shown enlarged in figure 4.10).

The data presented here seems to indicate that a tensor that obeys the deter-

[4] 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.

minant 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 4.9) does not seem to have a big impact on the final recomputation quality.

4.6 Conclusions

Table 4.1 summarises the expressions for the different tensors, their degrees of freedom, their rank and their main constraints. In particular note the similarities between the expressions for the tensors.

We have demonstrated in this chapter how Geometric Algebra can be used to give a unified formalism for multiple view tensors. Almost all properties of the tensors could be arrived at from geometric considerations alone. In this way the Geometric Algebra approach is much more intuitive than traditional tensor methods. We have gained this additional insight into the workings of multiple view tensors because Projective Geometry in terms of Geometric Algebra allows us to describe directly the geometry on which multiple view tensors are based. Therefore, we can understand their "inner workings" and inter-relations. The best examples of this are probably the derivations of the constraints on T and Q which followed from the fact that the intersection points of a line with three planes all have to lie along a line. It is hard to imagine a more trivial fact.

A similar analysis of multiple view tensors was presented by Heyden in [32]. However, we believe our treatment of the subject is more intuitive due to its geometric nature. In particular the "quadratic p-relations" used by Heyden were here replaced by the geometric fact that the intersection point of a line with a plane lies on that line.

We hope that our unified treatment of multiple view tensors has not just demonstrated the power of Geometric Algebra, but will also give a useful new tool to researchers in the field of Computer Vision.

Fundamental Matrix	Trifocal Tensor
$F_{i_1 j_1} = \llbracket A^{i_2} A^{i_3} B^{j_2} B^{j_3} \rrbracket$	$T_{i_1}{}^{jk} = \llbracket A^{i_2} A^{i_3} B^{j} C^{k} \rrbracket$
$F_{i j_1} = \varepsilon^{j_3}_{ba} K^{b}_{{j_2}_i} - \varepsilon^{j}_{ba} K^{b}_{{j_3}_i}$	$T_{i}{}^{jk} = \varepsilon^{j}_{ba} K^{c}_{k_i} - \varepsilon^{k}_{ca} K^{b}_{j_i}$
$F_{ij} = L^{a}_{i} \cdot \underbrace{(B_j \wedge B_4)}_{\text{line}}$	$T_{i}{}^{jk} = L^{a}_{i} \cdot \underbrace{\llbracket B^{j} C^{k} \rrbracket}_{\text{line}}$
$\det F = 0$	$\det(T_i{}^{jk})_{jk} = 0$ for each i
7 DOF	18 DOF
rank 2	rank 4

Quadfocal Tensor
$Q^{ijkl} = \llbracket A^{i} B^{j} C^{k} D^{l} \rrbracket$
$Q^{i_1 jkl} = \varepsilon^{j}_{ba} \Big[K^{c}_{k_{i_3}} K^{d}_{l_{i_2}} - K^{d}_{l_{i_3}} K^{c}_{k_{i_2}} \Big]$ $- \varepsilon^{k}_{ca} \Big[K^{b}_{j_{i_3}} K^{d}_{l_{i_2}} - K^{d}_{l_{i_3}} K^{b}_{j_{i_2}} \Big]$ $+ \varepsilon^{l}_{da} \Big[K^{b}_{j_{i_3}} K^{c}_{k_{i_2}} - K^{c}_{k_{i_3}} K^{b}_{j_{i_2}} \Big]$
$Q^{ijkl} = A^{i} \cdot \underbrace{\llbracket B^{j} C^{k} D^{l} \rrbracket}_{\text{point}}$
$\det(Q^{ijkl})_{xy} = 0$ where x and y are any pair of $\{ijkl\}$
29 DOF
rank 9

Table 4.1: Comparison of Multiple View Tensors

Chapter 5

3D-Reconstruction

Now is the winter of our discontent
Made glorious summer by this son of York;
And all the clouds that loured upon our house
In the deep bosom of the ocean burried.

"Richard III", William Shakespeare.

Although I am no son of York, I hope that this chapter will bury any doubt the reader may have about the usefulness of GA (or the previous chapters), and bring this thesis to a "glorious" conclusion.

In the following we will consider a system of two pinhole cameras looking at a scene in the world. Our goal is to create a 3D-reconstruction of the world scene from the images taken by the pinhole cameras. We will show that a (affine) 3D-reconstruction is possible if we know a number of point matches between the images and also some parallel world lines.

5.1 Introduction

3D-reconstruction is currently an active field in Computer Vision, not least because of its many applications. It is applicable wherever the "real world" has to be understood by a computer. This may be with regard to control movement (robots), to survey a scene for later interpretation (medicine), or to create and mix artifical with real environments (special effects).

Research on 3D-reconstruction can roughly be separated into three areas:

1. *Reconstruction with calibrated cameras.* [40, 38, 37, 34, 19, 46, 63] In this case, a set of images is taken of a scene with one or more calibrated

cameras. However, the camera positions are unknown. To perform a 3D-reconstruction we therefore first have to reconstruct the camera positions. To do this it is assumed that point matches between all the points are known.

2. *Reconstruction from sequences of images.* [66, 68, 67, 71, 12, 10, 43] Here a series of monocular, binocular or trinocular images is taken. To perform a reconstruction it is then assumed that point matches between the views in space and over time are known, and that the relative camera geometry and their internal parameters do not change. A popular method in this area is the use of the Kruppa equations [36, 13].
3. *Reconstruction from static views.* [3, 4] A set of images of a scene taken with unknown cameras, from unknown positions is given. We still assume that we have point matches over the images. However, note that we cannot assume anymore that the internal parameters of the cameras that took the images are the same.

The least information about a scene is given in point 3. In fact, there is so little information that a correct 3D-reconstruction is *impossible*, as we have shown in chapter 3. Therefore, some additional information is needed. Such information could be the knowledge of lengths, angles or parallel lines.

Our approach to 3D-reconstruction falls into the area of *Reconstruction from static views.* We have two images taken with unknown cameras from unknown positions and assume that apart from the point matches we also know the projections of a number of sets of parallel world lines. The latter are used to find vanishing points but also to constrain the reconstruction. This information allows us to perform an *affine* reconstruction of the scene. That is, we find the rotation, translation and the internal parameters of the second camera *relative* to the first. If we assume furthermore, that we have three mutually orthogonal sets of parallel lines, we can also find the internal calibration of the first camera and thus obtain a Euclidean 3D-reconstruction.

In the following discussion of our reconstruction algorithm we use the same notation as in chapter 3. We will also assume that the reader is familiar with our description of reciprocal frames, pinhole cameras, camera matrices and the basic form of the fundamental matrix. Of course, all this assumes some familiarity with Geometric Algebra (GA).

5.2 Image Plane Bases

Our general setup is that we have two pinhole cameras described by frames $\{A_\mu\}$ and $\{B_\mu\}$, respectively. The frame $\{A_\mu\}$ is also regarded as the world frame which we use for our reconstruction. Hence, we obtain an affine reconstruction.

The basic form of our calculation is as follows. We start with the image points obtained from real cameras, i.e. in $\mathcal{E}^3$. These image points are then embedded in $\mathcal{P}^3$. All our calculations are then performed in $\mathcal{P}^3$ and the resultant reconstruction is projected back into $\mathcal{E}^3$. This method forces us to take note of two important concepts.

1. **Correct Basis.** The power of GA in this field derives from the fact that we are not working purely with coordinates, but with the *underlying geometric basis.* Therefore, we have to make sure that the basis we are working with is actually appropriate for our problem.

2. **Scale Invariance.** The projection of homogeneous vectors into $\mathcal{E}^3$ is independent of the overall scale of the homogeneous vector. Calculations in $\mathcal{P}^3$ may depend on such an overall scale, though. We have to make sure that all our calculations are *invariant* under a scaling of the homogeneous vectors, because such a scaling cannot and should not have any influence on our final result. Furthermore, since we are initially embedding vectors from $\mathcal{E}^3$ in $\mathcal{P}^3$, we are not given any particular scale. Any expression that is invariant under a scaling of its component homogeneous vectors will be called *scale invariant.*

As mentioned above, the frames $\{A_\mu\}$ and $\{B_\mu\}$ define two pinhole cameras. Since $\{A_\mu\}$ also serves as our world frame in $\mathcal{P}^3$ we can choose that A_4, the optical centre of camera A, sits at the origin. A_1, A_2 and A_3 define the image plane of camera A. If we want to be true to our previously stated concepts, we need to give some thought as to how we should choose the $\{A_i\}$.

Note here that we use latin indices to count from 1 to 3 and greek indices to count from 1 to 4. We also make use of the Einstein summation convention, i.e. if a superscript index is repeated as a subscript within a product, a summation over the range of the index is implied. Hence, $\alpha^i A_i \equiv \sum_{i=1}^{3} \alpha^i A_i$.

The images we obtain from real cameras are 2-dimensional. Therefore, the image point coordinates we get are of the form $\{x, y\}$, which give the displacement in a horizontal and vertical direction[1] in the image coordinate frame. However, in $\mathcal{P}^3$ an image plane is defined by three vectors. Therefore, a point on a plane in $\mathcal{P}^3$ is defined by *three* coordinates. A standard way given in the literature to extend the 2D image point coordinates obtained from a real camera to $\mathcal{P}^3$ is by writing the vector $\{x, y\}$ as $\{x, y, 1\}$. This is a well

[1] Note that although we call these directions horizontal and vertical, they may not be at a 90 degree angle to each other in general.

founded and very practical choice, and if we just worked with matrices and tensors we would not need to do anything else. However, since we want to tap into the power of GA, we need to understand what kind of basis is *implicitly assumed* when we write our image point coordinates in the form $\{x, y, 1\}$.

The best way to proceed, is first to describe a 2D-image point in a 3D basis and then to embed this point in $\mathcal{P}^3$. An image point $\{x, y\}$ gives the horizontal and vertical displacements in the 2D-image plane coordinate frame. Let the basis corresponding to this 2D frame in $\mathcal{E}^3$ be $\{\boldsymbol{a}_1, \boldsymbol{a}_2\}$. If we define a third vecor $\boldsymbol{a}_3$ to point to the origin of the 2D frame in $\mathcal{E}^3$, then an image point with coordinates $\{x, y\}$ can be expressed as follows in $\mathcal{E}^3$.

$$\boldsymbol{x}_a = x\,\boldsymbol{a}_1 + y\,\boldsymbol{a}_2 + 1\,\boldsymbol{a}_3 = \hat{\alpha}^i\,\boldsymbol{a}_i, \tag{5.1}$$

with $\{\hat{\alpha}^i\} \equiv \{x, y, 1\}$. The $\{\hat{\alpha}^i\}$ are the image point coordinates corresponding to image point $\{x, y\}$ in $\mathcal{E}^3$. Now we embed the point $\boldsymbol{x}_a$ in $\mathcal{P}^3$.

$$\boldsymbol{x}_a \xrightarrow{\mathcal{P}^3} X_a = \boldsymbol{x}_a + e_4 = \hat{\alpha}^i A_i, \tag{5.2}$$

where we defined $A_1 \equiv \boldsymbol{a}_1$, $A_2 \equiv \boldsymbol{a}_2$ and $A_3 \equiv \boldsymbol{a}_3 + e_4$. That is, A_1 and A_2 are *direction vectors*, or points at infinity, because they have no e_4 component[2]. However, they still lie on image plane A. More precisely, they lie on the intersection line of image plane A with the plane at infinity. Note that A_1 and A_2 do not project back to $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, respectively. For example,

$$A_1 \xrightarrow{\mathcal{E}^3} \frac{A_1\cdot e^i}{A_1\cdot e^4} e_i = \frac{\boldsymbol{a}_1}{0} \longrightarrow \infty \tag{5.3}$$

Nevertheless, $\{A_i\}$ is still the projective image plane basis we are looking for, as can be seen when we project X_a down to Euclidean space.

$$X_a \xrightarrow{\mathcal{E}^3} \boldsymbol{x}_a = \frac{X_a\cdot e^i}{X_a\cdot e^4} e_i = \frac{\hat{\alpha}^i\,\boldsymbol{a}_i}{\hat{\alpha}^3} = x\,\boldsymbol{a}_1 + y\,\boldsymbol{a}_2 + 1\,\boldsymbol{a}_3 \tag{5.4}$$

What is important here is that neither $\hat{\alpha}^1$ nor $\hat{\alpha}^2$ appear in the denominator. This shows that by writing our image point coordinates in the form $\{x, y, 1\}$ we have implicitly assumed this type of basis. We will call this type of frame a **normalised homogeneous camera frame**. The camera frames we will use in the following are all normalised homogeneous camera frames.

[2] This shows very nicely that a Euclidean vector interpreted as a homogeneous vector is a direction.

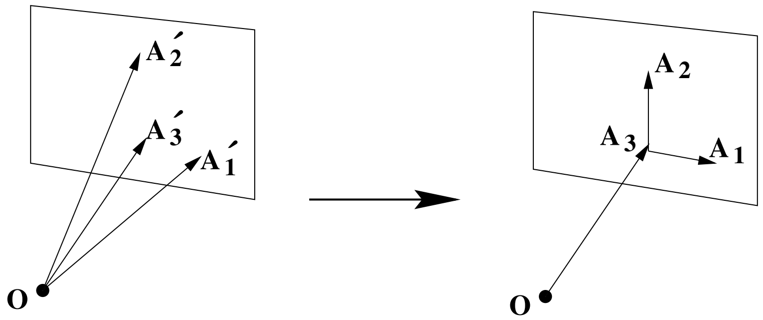


Figure 5.1: Transformation from general basis to a particular basis in which image points have coordinates of the type $\{x, y, 1\}$.

It might seem a bit odd that we have devoted so much space to the development of normalised homogeneous camera frames. However, this has far reaching implication later on and is *essential* to understand our derivation.

In $\mathcal{P}^3$ a point on the image plane of camera A can be written as $X_a = \alpha^i A_i$ in general. We can normalise the coordinates without changing the projection of X_a into $\mathcal{E}^3$. That is, $X_a \simeq \bar{\alpha}^i A_i$ with $\bar{\alpha}^i \equiv \alpha^i/\alpha^3$. The symbol $\simeq$ means equality up to a scalar factor. In this case we clearly have $\{\bar{\alpha}^i\} = \{\hat{\alpha}^i\}$.

A general point in $\mathcal{P}^3$ can be written as $X = \alpha^\mu A_\mu$ in the A-frame. We can normalise the coordinates of X_a in the same way as before to obtain $X \simeq \bar{\alpha}^\mu A_\mu$ with $\bar{\alpha}^\mu \equiv \alpha^\mu/\alpha^3$. If we project this point down to $\mathcal{E}^3$ we get[3]

$$\begin{aligned} X \xrightarrow{\mathcal{E}^3} \boldsymbol{x} &= \frac{X\cdot e^i}{X\cdot e^4}\, e_i = \frac{(\alpha^\mu A_\mu)\cdot e^i}{(\alpha^\mu A_\mu)\cdot e^4}\, e_i \\ &= \frac{\alpha^i}{\alpha^3+\alpha^4}\, \boldsymbol{a}_i = \frac{\bar{\alpha}^i}{1+\bar{\alpha}^4}\, \boldsymbol{a}_i \\ &= \hat{\alpha}^i\, \boldsymbol{a}_i\ ; \qquad\qquad \hat{\alpha}^i \equiv \frac{\bar{\alpha}^i}{1+\bar{\alpha}^4} \end{aligned} \tag{5.5}$$

Therefore, if $\bar{\alpha}^4 = 0$, then X is a point on the image plane of camera A. Also, if $\bar{\alpha}^4 = -1$ then X is a point at infinity. We will call $\bar{\alpha}^4$ the **projective depth** of a point in $\mathcal{P}^3$.

In $\mathcal{P}^3$ a general plane is defined by three homogeneous vectors that give points on that plane. We will now show how we can transform such a general basis into a normalised homogeneous camera frame. Figure 5.1 shows this transformation.

[3] Recall that $A_4 = e_4$ (the origin of $\mathcal{P}^3$) and that the $\{A_i\}$ are a normalised homogeneous camera frame.

Let the $\{A'_i\}$ be normalised homogeneous vectors, i.e. $A'_i \cdot e_4 = 1$. This can be assumed without loss of generality, because any homogeneous vector can be normalised without changing the point it corresponds to in $\mathcal{E}^3$. A point X'_a on plane $P'_a \equiv A'_1 \wedge A'_2 \wedge A'_3$ may then be given by

$$\begin{aligned} X'_a &= \acute{\alpha}^1 A'_1 + \acute{\alpha}^2 A'_2 + \acute{\alpha}^3 A'_3 \\ &= \acute{\alpha}^1 (A'_1 - A'_3) + \acute{\alpha}^2 (A'_2 - A'_3) + (\acute{\alpha}^1 + \acute{\alpha}^2 + \acute{\alpha}^3) A'_3 \\ &\equiv \alpha^1 A_1 + \alpha^2 A_2 + \alpha^3 A_3 \\ &= X_a \end{aligned} \tag{5.6}$$

where we identified $A_1 \equiv A'_1 - A'_3$, $A_2 \equiv A'_2 - A'_3$ and $A_3 \equiv A'_3$. Also $\alpha^1 \equiv \acute{\alpha}^1$, $\alpha^2 \equiv \acute{\alpha}^2$ and $\alpha^3 \equiv (\acute{\alpha}^1 + \acute{\alpha}^2 + \acute{\alpha}^3)$. A_1 and A_2 are directions now, i.e. $A_1 \cdot e_4 = A_2 \cdot e_4 = 0$, but we still have $P'_a = A_1 \wedge A_2 \wedge A_3$. That is, the $\{A_i\}$, which are a normalised homogeneous camera frame, are also a valid basis for plane P'_a. As before we now have $\{\hat{\alpha}^i\} = \{\bar{\alpha}^i\}$.

5.3 Plane Collineation

Before we can get started on the actual reconstruction algorithm, we need to derive some more mathematical objects which we will need as tools. The problem we want to solve first is the following. Let us assume we have three image point matches in cameras A and B. That is, if three points in space, $\{X_i\}$, are projected onto image planes A and B to give images $\{X_i^a\}$ and $\{X_i^b\}$ respectively, then we know that the pairs $\{X_i^a, X_i^b\}$ are images of the same point in space. If the three points in space do not lie along a line, they define a plane. This plane induces a collineation, which means that we can transfer image points from camera A to camera B through that plane. For example, let X_4^a be the image point on image plane A which we want to transfer to camera B through the plane. First we have to find the intersection point of line $A_4 \wedge X_4^a$ with the plane[4], and then we project this intersection point onto image plane B (see figure 5.2). This transformation can also be represented by a 3×3 matrix, which is called a collineation matrix. Our goal is to find the collineation induced by the plane $P \equiv X_1 \wedge X_2 \wedge X_3$ by knowing the projections of the points $\{X_i\}$ onto image planes A and B, and the fundamental matrix for the two cameras. Since we know the fundamental matrix we can also calculate the epipoles. The epipoles on the two image planes are always projections of

[4] Recall that A_4 is the optical centre of camera A.

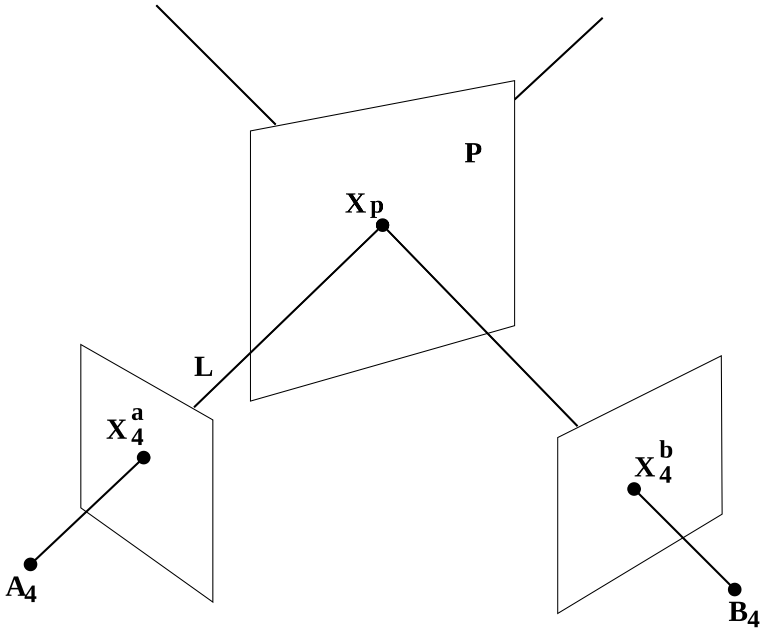


Figure 5.2: Schematic representation of a plane collineation. Image point X_4^a is projected to X_4^b under the P-collineation.

a single point in space and thus give us the projections of a fourth point from any plane in space. That is, we have in fact the projections of four points that lie on some plane P. Hence, we can find the collineation matrix directly through a matrix diagonalisation.

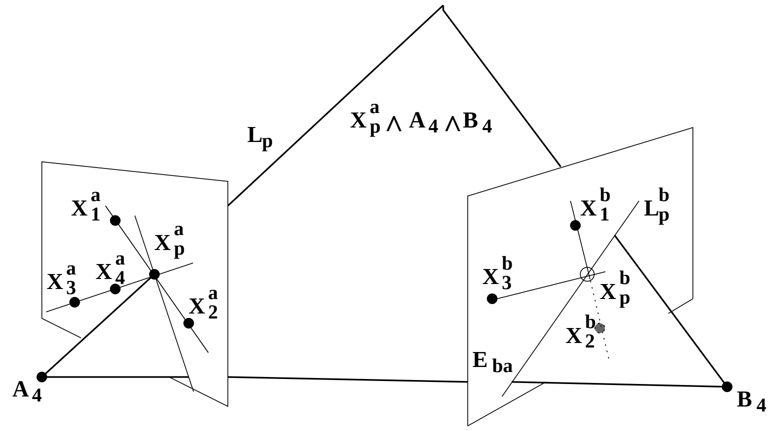
However, it is interesting to see what this means geometrically. Faugeras gives a geometrical interpretation[5] in [13]. We will follow his construction method to obtain a 3 × 3 × 3 collineation tensor.

5.3.1 Calculating the Collineation Tensor M

We start by defining three points $X_i = \alpha_i^{\mu} A_{\mu}$. The projections of these three points onto image planes A and B are $X_i^a = \bar{\alpha}_i^j A_j$ and $X_i^b = \bar{\beta}_i^j B_j$, respectively. We know the coordinates $\{\bar{\alpha}_i^j\}$ and $\{\bar{\beta}_i^j\}$, and we know that the pairs $\{\bar{\alpha}_i^j, \bar{\beta}_i^k\}$ are images of the same point in space. Furthermore, we have the fundamental matrix for the two cameras. We find the collineation induced by the plane $P = X_1 \wedge X_2 \wedge X_3$ geometrically, through a two step construction.

[5] In [13] this method is called the **Point-Plane** procedure.

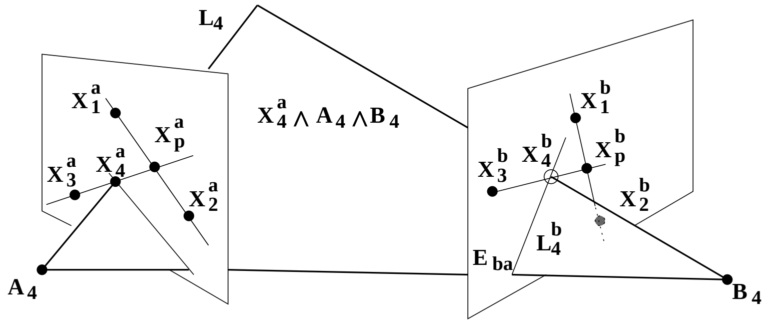
Step 1:



Let $X_4^a = \alpha_4^i A_i$ be the image point we want to project onto image plane B under the P-collineation. Now consider the intersection point X_p^a of lines $X_3^a \wedge X_4^a$ and $X_1^a \wedge X_2^a$. The intersection point of line $L_p \equiv A_4 \wedge X_p^a$ with an arbitrary plane in $\mathcal{P}^3$ obviously lies on L_p. Denote the projection of L_p onto image plane B by L_p^b. Obviously X_p^a can only be projected to some point on L_p^b, independent of the collineation. We also know that X_p^a has to project to some point on the line $X_1^b \wedge X_2^b$ under the specific P-collineation. Hence, X_p^b is the intersection point of lines L_p^b and $X_1^b \wedge X_2^b$. We can also write this as

$$X_p^b = (X_p^a \wedge A_4 \wedge B_4) \vee (X_1^b \wedge X_2^b) \tag{5.7}$$

Step 2:



Now that we have calculated the point X_p^b, we can project X_4^a under the P-collineation in an analogous way. We form a line $L_4 = A_4 \wedge X_4^a$ which we project onto image plane B. X_4^b, the projection of X_4^a under the P-collineation, is then the intersection point of L_4^b and line $X_3^b \wedge X_p^b$. This can also be expressed as

$$X_4^b = (X_4^a \wedge A_4 \wedge B_4) \vee (X_3^b \wedge X_p^b) \tag{5.8}$$

Now we will perform this two step calculation, starting with X_p^a.

$$\begin{aligned} X_p^a &= (A_4 \wedge X_3^a \wedge X_4^a) \vee (X_1^a \wedge X_2^a) \\ &= \llbracket A_4 X_3^a X_1^a X_4^a \rrbracket X_2^a + \llbracket A_4 X_2^a X_3^a X_4^a \rrbracket X_1^a \\ &= \phi_{pa}^1 X_1^a + \phi_{pa}^2 X_2^a \end{aligned} \tag{5.9}$$

with $\phi_{pa}^1 \equiv \llbracket A_4 X_2^a X_3^a X_4^a \rrbracket$ and $\phi_{pa}^2 \equiv \llbracket A_4 X_3^a X_1^a X_4^a \rrbracket$. X_p^b is then found by

$$\begin{aligned} X_p^b &= (X_p^a \wedge A_4 \wedge B_4) \vee (X_1^b \wedge X_2^b) \\ &= -\llbracket X_p^a X_1^b A_4 B_4 \rrbracket X_2^b + \llbracket X_p^a X_2^b A_4 B_4 \rrbracket X_1^b \\ &= -\phi_{pa}^2 \llbracket X_2^a X_1^b A_4 B_4 \rrbracket X_2^b + \phi_{pa}^1 \llbracket X_1^a X_2^b A_4 B_4 \rrbracket X_1^b \\ &= \phi_{pb}^1 X_1^b + \phi_{pb}^2 X_2^b \end{aligned} \tag{5.10}$$

with $\phi_{pb}^1 \equiv \phi_{pa}^1 \llbracket X_1^a X_2^b A_4 B_4 \rrbracket$ and $\phi_{pb}^2 \equiv -\phi_{pa}^2 \llbracket X_2^a X_1^b A_4 B_4 \rrbracket$. The step from line 2 to line 3 in the previous calculation follows because $\llbracket X_i^a X_i^b A_4 B_4 \rrbracket = 0$. (Recall that the pairs $\{X_i^a, X_i^b\}$ are projections of the same point in space.) We are now in a position to calculate X_4^b.

$$\begin{aligned} X_4^b &= (X_4^a \wedge A_4 \wedge B_4) \vee (X_3^b \wedge X_p^b) \\ &= -\llbracket X_4^a X_3^b A_4 B_4 \rrbracket X_p^b + \llbracket X_4^a X_p^b A_4 B_4 \rrbracket X_3^b \\ &= -\phi_{pb}^1 \llbracket X_4^a X_3^b A_4 B_4 \rrbracket X_1^b - \phi_{pb}^2 \llbracket X_4^a X_3^b A_4 B_4 \rrbracket X_2^b \\ &\quad + \phi_{pb}^1 \llbracket X_4^a X_1^b A_4 B_4 \rrbracket X_3^b + \phi_{pb}^2 \llbracket X_4^a X_2^b A_4 B_4 \rrbracket X_3^b \end{aligned} \tag{5.11}$$

If we write $X_4^b = \beta_4^k B_k$ then the $\{\beta_4^k\}$ are given by

$$\begin{aligned} \beta_4^k &= -\phi_{pb}^1 \llbracket X_4^a X_3^b A_4 B_4 \rrbracket \beta_1^k - \phi_{pb}^2 \llbracket X_4^a X_3^b A_4 B_4 \rrbracket \beta_2^k \\ &\quad + \Big(\phi_{pb}^1 \llbracket X_4^a X_1^b A_4 B_4 \rrbracket + \phi_{pb}^2 \llbracket X_4^a X_2^b A_4 B_4 \rrbracket \Big) \beta_3^k \end{aligned} \tag{5.12}$$

At this point we should think about whether we can use the normalised image point coordinates $\{\bar{\alpha}_j^i\}$, $\{\bar{\beta}_j^i\}$ and $\{\bar{\alpha}_4^i\}$, instead of their unnormalised counterparts. If this is not the case, we cannot use equation (5.12). Let the $\{\bar{\phi}_{pa}^i\}$

be the $\{\phi^i_{pa}\}$ calculated from normalised coordinates. They are related in the following way.

$$\phi^1_{pa} = \alpha^3_2\alpha^3_3\alpha^3_4\bar{\phi}^1_{pa} \tag{5.13a}$$

$$\phi^2_{pa} = \alpha^3_3\alpha^3_1\alpha^3_4\bar{\phi}^2_{pa} \tag{5.13b}$$

Therefore, the relation between the $\{\bar{\phi}^i_{pb}\}$ and $\{\phi^i_{pb}\}$ is

$$\phi^1_{pb} = \alpha^3_2\alpha^3_3\alpha^3_4\alpha^3_1\beta^3_2\bar{\phi}^1_{pb} \tag{5.14a}$$

$$\phi^2_{pb} = \alpha^3_3\alpha^3_1\alpha^3_4\alpha^3_2\beta^3_1\bar{\phi}^2_{pb} \tag{5.14b}$$

Hence, the $\{\phi^i_{pb}\}$ have the term $(\alpha^3_1\alpha^3_2\alpha^3_3\alpha^3_4)$ in common. Therefore, we can write equation (5.12) as

$$\begin{aligned}
\beta^k_4 &\simeq -\beta^3_2\bar{\phi}^1_{pb}\,(\alpha^i_4\beta^j_3F_{ij})\,\beta^k_1 - \beta^3_1\bar{\phi}^2_{pb}\,(\alpha^i_4\beta^j_3F_{ij})\,\beta^k_2 \\
&\quad +\Big(\beta^3_2\bar{\phi}^1_{pb}\,(\alpha^i_4\beta^j_1F_{ij}) + \beta^3_1\bar{\phi}^2_{pb}\,(\alpha^i_4\beta^j_2F_{ij})\Big)\,\beta^k_3 \\
&\simeq \bar{\phi}^1_{pb}\,(\bar{\alpha}^i_4\bar{\beta}^j_3F_{ij})\,\bar{\beta}^k_1 + \bar{\phi}^2_{pb}\,(\bar{\alpha}^i_4\bar{\beta}^j_3F_{ij})\,\bar{\beta}^k_2 \\
&\quad -\Big(\bar{\phi}^1_{pb}\,(\bar{\alpha}^i_4\bar{\beta}^j_1F_{ij}) + \bar{\phi}^2_{pb}\,(\bar{\alpha}^i_4\bar{\beta}^j_2F_{ij})\Big)\,\bar{\beta}^k_3
\end{aligned} \tag{5.15}$$

where $F_{ij} \equiv [\![A_iB_jA_4B_4]\!]$ is the fundamental matrix relating cameras A and B. That is, we can find the $\{\beta^k_4\}$ up to an overall constant from the $\{\bar{\alpha}^j_i\}$, $\{\bar{\beta}^j_i\}$ and $\{\bar{\alpha}^j_4\}$. To obtain our final equation we will expand the $\{\bar{\phi}^i_{pa}\}$ and $\{\bar{\phi}^i_{pb}\}$.

$$\begin{aligned}
\bar{\phi}^{j_1}_{pa} &= [\![A_4\bar{X}^a_4\bar{X}^a_{j_2}\bar{X}^a_{j_3}]\!] \\
&= \bar{\alpha}^i_4\,(\bar{\alpha}^{k_2}_{j_2}\bar{\alpha}^{k_3}_{j_3} - \bar{\alpha}^{k_3}_{j_2}\bar{\alpha}^{k_2}_{j_3})\,[\![A_4A_iA_{k_2}A_{k_3}]\!] \\
&\simeq \bar{\alpha}^i_4\,\bar{\lambda}^{j_1}_{a\,i}
\end{aligned} \tag{5.16}$$

with

$$\bar{\lambda}^{j_1}_{a\,k_1} \equiv (\bar{\alpha}^{k_2}_{j_2}\bar{\alpha}^{k_3}_{j_3} - \bar{\alpha}^{k_3}_{j_2}\bar{\alpha}^{k_2}_{j_3}). \tag{5.17}$$

To simplify the final equation we make the following definitions.

$$F(r,s) \equiv \bar{\alpha}^i_r\bar{\beta}^j_s\,F_{ij} \tag{5.18a}$$

$$f^b_{ir} \equiv \bar{\beta}^j_r\,F_{ij} \tag{5.18b}$$

Now we can write the $\{\bar{\phi}^i_{pb}\}$ as

$$\bar{\phi}^1_{pb} = \bar{\alpha}^i_4\,\bar{\lambda}^1_{a\,i}\;F(1,2) \tag{5.19a}$$

$$\bar{\phi}^2_{pb} = -\bar{\alpha}^i_4\,\bar{\lambda}^2_{a\,i}\;F(2,1) \tag{5.19b}$$

Therefore, equation (5.15) becomes

$$\beta^k_4 \simeq \bar{\alpha}^i_4\bar{\alpha}^j_4\;M^k_{ij}, \tag{5.20}$$

with

$$\begin{aligned} M^k_{ij} \equiv \Big[\quad & \Big(F(1,2)\,\bar{\lambda}^1_{a\,i}\,\bar{\beta}^k_1 - F(2,1)\,\bar{\lambda}^2_{a\,i}\,\bar{\beta}^k_2\Big)\;f^b_{j3} \\ - \quad & \Big(F(1,2)\,\bar{\lambda}^1_{a\,i}\,f^b_{j1} - F(2,1)\,\bar{\lambda}^2_{a\,i}\,f^b_{j2}\Big)\;\bar{\beta}^k_3\Big] \end{aligned} \tag{5.21}$$

Note that since M^k_{ij} can be calculated from the normalised image point coordinates, it is scale invariant.

To our knowledge a collineation tensor which can be used instead of the collineation matrix has not been derived before. Faugeras describes the method which we followed to find M^k_{ij} [13], but he does not obtain a simple tensor to perform the collineation projection. The advantage of calculating a collineation tensor is that we do not have to perform any diagonalisations. The price we pay is the higher computational cost of equation (5.20) compared to using a collineation matrix.

5.3.2 Rank of M

Equation (5.20) seems to indicate that a collineation is a quadratic relation. However, we know that $\beta^k_4 = \alpha^i_4\Psi^k_i$ where Ψ^k_i is the collineation matrix. If we take a closer look at the components of equation (5.21) we see that M is not of full rank. Reordering the terms of equation (5.21) gives

$$\begin{aligned} M^k_{ij} \;=\; & \Big(F(1,2)\,f^b_{j3}\,\bar{\beta}^k_1 - F(1,2)\,f^b_{j1}\,\bar{\beta}^k_3\Big)\;\bar{\lambda}^1_{a\,i} \\ & -\Big(F(2,1)\,f^b_{j3}\,\bar{\beta}^k_2 - F(2,1)\,f^b_{j2}\,\bar{\beta}^k_3\Big)\;\bar{\lambda}^2_{a\,i}. \end{aligned} \tag{5.22}$$

Now, $\bar{\lambda}^r_{a\,3}$ is a linear combination of $\bar{\lambda}^r_{a\,1}$ and $\bar{\lambda}^r_{a\,2}$. The relation is

$$\bar{\lambda}^{r_1}_{a\,3} = -\frac{\bar{\alpha}^1_{r_2}}{\bar{\alpha}^3_{r_2}}\;\bar{\lambda}^{r_1}_{a\,1} - \frac{\bar{\alpha}^2_{r_2}}{\bar{\alpha}^3_{r_2}}\;\bar{\lambda}^{r_1}_{a\,2}. \tag{5.23}$$

Therefore, the three matrices in indices i, j of M^k_{ij} are of rank 2. Furthermore, we can write equation (5.20) as

$$\begin{aligned}\beta^k_4 \;&\simeq\; \bar{\alpha}^1\bar{\alpha}^1\ M^k_{11} + \bar{\alpha}^2\bar{\alpha}^2\ M^k_{22} + \bar{\alpha}^1\bar{\alpha}^2\ (M^k_{12} + M^k_{21}) \\ &\quad + \bar{\alpha}^1\ (M^k_{13} + M^k_{31}) + \bar{\alpha}^2\ (M^k_{23} + M^k_{32}) + \bar{\alpha}^3\ M^k_{33}\end{aligned} \tag{5.24}$$

since $\bar{\alpha}^3 = 1$ by definition. Thus, if we perform a set of similarity transforms on M^k_{ij} such that the components $M^k_{11}, M^k_{22}, M^k_{12}, M^k_{21}$ are zero, we can read off the components of the collineation matrix from the transformed M^k_{ij}. Such a similarity transformation on M^k_{ij} is possible because the matrices in indices i, j of M^k_{ij} are of rank 2.

5.4 The Plane at Infinity

It will be very useful to see what the collineation of the plane at infinity looks like. Without loss of generality we can set $A_4 = e_4$. Also recall that A_1 and A_2 are direction vectors, i.e. have no e_4 component, by definition. Therefore, the plane at infinity P_∞ may be given by

$$P_\infty = A_1 \wedge A_2 \wedge (A_3 - A_4) \tag{5.25}$$

Now that we have the plane at infinity we can also find an expression for the collineation matrix associated with it. We want to project a point $X^a = \alpha^i A_i$ on image plane A to image plane B under the P_∞-collineation. First we have to find the intersection point X_p of line $L = A_4 \wedge X^a$ with P_∞.

$$\begin{aligned}X_p \;&=\; (A_4 \wedge X^a) \vee P_\infty \\ &=\; [\![A_4 X^a]\!] \cdot \Big((A_1 \wedge A_2 \wedge A_3) - (A_1 \wedge A_2 \wedge A_4)\Big) \\ &=\; \sum_{i_1} \Big([\![A_4 X^a A_{i_2} A_{i_3}]\!] A_{i_1}\Big) - [\![A_4 X^a A_1 A_2]\!] A_4 \\ &=\; \alpha^j \sum_{i_1} [\![A_4 A_j A_{i_2} A_{i_3}]\!] A_{i_1} - \alpha^j [\![A_4 A_1 A_2 A_j]\!] A_4 \\ &\simeq\; \alpha^i A_i - \alpha^3 A_4\end{aligned} \tag{5.26}$$

Now we need to find the projection X_p^b of X_p onto image plane B.

$$\begin{aligned} X_p^b &= X_p \cdot B^j \, B_j \\ &\simeq \left(\alpha^i \, A_i \cdot B^j - \alpha^3 \, A_4 \cdot B^j \right) B_j \\ &= \left(\alpha^i K_i^{j\,\mathsf{b}} - \alpha^3 \varepsilon_{ba}^j \right) B_j \end{aligned} \tag{5.27}$$

where $K_i^{j\,\mathsf{b}} \equiv A_i \cdot B^j$ is the 3×3 camera matrix minor of camera B, and $\varepsilon_{ba}^j \equiv A_4 \cdot B^j$ is the epipole of camera B and also the fourth column of the full camera matrix[6]. That is,

$$K_\mu^{j\,\mathsf{b}} = [K_i^{j\,\mathsf{b}}, \varepsilon_{ba}^j] \tag{5.28}$$

From equation (5.27) it follows that we can write the collineation of P_∞ as

$$\Psi_i^{j\,\infty} \equiv [K_1^{j\,\mathsf{b}}, K_2^{j\,\mathsf{b}}, K_3^{j\,\mathsf{b}} - \varepsilon_{ba}^j] \tag{5.29}$$

where i counts the columns. Then we can write, as before

$$\beta_\infty^j \simeq \alpha^i \, \Psi_i^{j\,\infty} \tag{5.30}$$

for the projection of image point X^a under the P_∞-collineation.

What does the P_∞-collineation describe geometrically? If X^a is an image point in camera A and X^b is its projection under the P_∞-collineation, then from the construction of the collineation it follows that the lines $L_a = A_4 \wedge X^a$ and $L_b = B_4 \wedge X^b$ meet in a point on P_∞. If two lines meet in a point on the plane at infinity, they are parallel. Therefore, the P_∞-collineation tells us which two image points X^a and X^b on image planes A and B, repectively, correspond such that the lines $A_4 \wedge X^a$ and $B_4 \wedge X^b$ are parallel. Obviously, this tells us something about the relative orientation of the two cameras.

We can use our knowledge of the relation between $\Psi^{\,\infty}$ and the camera matrix to find the depths of a set of world points whose projections are known in both cameras, if we also know the projections of at least three pairs of parallel lines. We will assume for the moment that for each point pair $\{\bar{\alpha}^i, \bar{\beta}^j\}$ we also know $\bar{\beta}_\infty^j$, which is the projection of $\bar{\alpha}^i$ under the P_∞-collineation. We

[6] The full camera matrix is given by $K_\mu^{j\,\mathsf{b}} = A_\mu \cdot B^j$. See [51] for details on camera matrices and epipoles.

will show later how the $\{\bar\beta^j_\infty\}$ may be calculated. From the definition of the camera matrix we know that

$$\beta^j = \alpha^i K^{j}_{i}{}^{\mathsf{b}} + \alpha^4 \varepsilon^j_{ba}. \tag{5.31}$$

Furthermore, equation (5.30) may be rewritten as

$$\beta^j_\infty \simeq \alpha^i K^{j}_{i}{}^{\mathsf{b}} - \alpha^3 \varepsilon^j_{ba} \tag{5.32}$$

First of all note the $\simeq$ symbol, which means that we cannot use this equation directly. Secondly, we do not actually know the $\{\alpha^i\}$, $\{\beta^i\}$ and $\{\varepsilon^i_{ba}\}$ but only the normalised coordinates[7] $\{\bar\alpha^i\}$, $\{\bar\beta^i\}$ and $\{\bar\varepsilon^i_{ba}\}$, because in practical applications we find all our image point coordinates in the form $\{x, y, 1\}$. Therefore, we have to rewrite equations 5.31 and 5.32 so that they are independent of the unknown scales. Equation 5.31 becomes

$$\begin{aligned} & \bar\beta^j = \frac{\beta^j}{\beta^3} = \frac{\bar\alpha^i \bar K^{j}_{i}{}^{\mathsf{b}} + \bar\alpha^4 \bar\varepsilon^j_{ba}}{\bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} + \bar\alpha^4} \\ \Longleftrightarrow \quad & \bar\alpha^i \bar\beta^j \bar K^{3}_{i}{}^{\mathsf{b}} + \bar\alpha^4 \bar\beta^j = \bar\alpha^i \bar K^{j}_{i}{}^{\mathsf{b}} + \bar\alpha^4 \varepsilon^j_{ba}, \end{aligned} \tag{5.33}$$

and equation (5.32) becomes

$$\begin{aligned} & \bar\beta^j_\infty = \frac{\beta^j_\infty}{\beta^3_\infty} = \frac{\bar\alpha^i \bar K^{j}_{i}{}^{\mathsf{b}} - \bar\varepsilon^j_{ba}}{\bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} - 1} \\ \Longleftrightarrow \quad & \bar\alpha^i \bar\beta^j_\infty \bar K^{3}_{i}{}^{\mathsf{b}} - \bar\beta^j_\infty = \bar\alpha^i \bar K^{j}_{i}{}^{\mathsf{b}} - \bar\varepsilon^j_{ba}. \end{aligned} \tag{5.34}$$

Note that $\bar K^{j}_{i}{}^{\mathsf{b}} \equiv K^{j}_{i}{}^{\mathsf{b}} / \varepsilon^3_{ba}$. Subtracting equation (5.34) from equation (5.33) gives

$$\begin{aligned} & \bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} (\bar\beta^j - \bar\beta^j_\infty) + \bar\alpha^4 \bar\beta^j + \bar\beta^j_\infty = \bar\alpha^4 \varepsilon^j_{ba} + \bar\varepsilon^j_{ba} \\ \Longleftrightarrow \quad & \bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} (\bar\beta^j - \bar\beta^j_\infty) + \bar\beta^j_\infty - \bar\varepsilon^j_{ba} = \bar\alpha^4 (\bar\varepsilon^j_{ba} - \bar\beta^j) \\ \Longleftrightarrow \quad & \bar\alpha^4 = \bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} \, \frac{\bar\beta^j_\infty - \bar\beta^j}{\bar\beta^j - \bar\varepsilon^j_{ba}} - \frac{\bar\beta^j_\infty - \bar\varepsilon^j_{ba}}{\bar\beta^j - \bar\varepsilon^j_{ba}} \; ; \quad j \in \{1, 2\}. \end{aligned} \tag{5.35}$$

The last equation may be written more succinctly as

$$\bar\alpha^4 = \bar\alpha^i \bar K^{3}_{i}{}^{\mathsf{b}} \, \zeta^j_1 - \zeta^j_2 \; ; \quad j \in \{1, 2\}. \tag{5.36}$$

[7] Recall that $\bar\alpha^i \equiv \bar\alpha^i / \bar\alpha^3$ and similarly for the other coordinates.

with

$$\zeta_1^j \equiv \frac{\bar{\beta}_\infty^j - \bar{\beta}^j}{\bar{\beta}^j - \bar{\varepsilon}_{ba}^j} \quad ; \quad \zeta_2^j \equiv \frac{\bar{\beta}_\infty^j - \bar{\varepsilon}_{ba}^j}{\bar{\beta}^j - \bar{\varepsilon}_{ba}^j} \tag{5.37}$$

Since equation (5.36) has to give the same result for both $j = 1$ and $j = 2$ independent of $\bar{K}_i^{3\,\mathsf{b}}$, it follows[8] that $\zeta_1^1 = \zeta_1^2$ and $\zeta_2^1 = \zeta_2^2$. Therefore, we will discard the superscript of the ζs in the following.

Equation 5.36 by itself is still not useful, since we neither know $\bar{\alpha}^4$ nor $\bar{K}_i^{3\,\mathsf{b}}$. However, if we had some constraints on the depth components $\bar{\alpha}^4$ for a number of points we could find $\bar{K}_i^{3\,\mathsf{b}}$. Once $\bar{K}_i^{3\,\mathsf{b}}$ is known for a particular camera setup, we can use it to calculate the (affine) depths for any point matches. Before we show how $\bar{K}_i^{3\,\mathsf{b}}$ can be evaluated, we will take a closer look at how to find the $\{\bar{\beta}_\infty^j\}$.

5.5 Vanishing Points and P_∞

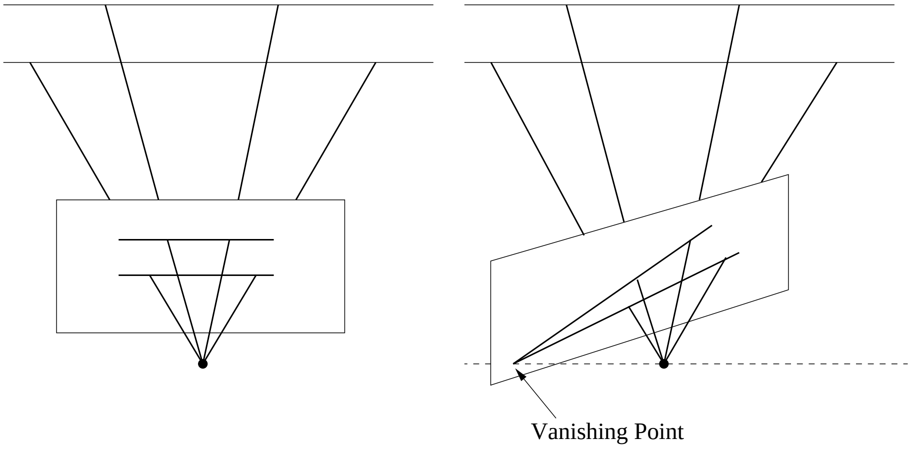


Figure 5.3: The figure demonstrates that the projections of two parallel world lines onto an image plane, are only parallel if the image plane is parallel to the world lines. The intersection point of the projections of two parallel world lines is called a *vanishing point*.

We mentioned earlier that the $\{\beta_\infty^j\}$ are the projections of the $\{\alpha^i\}$ onto image plane B under the P_∞-collineation. We can find the P_∞-collineation matrix $\Psi^{\,\infty}$, or tensor $M^{\,\infty}$, from the projection pairs of three points on P_∞ and the fundamental matrix.

[8] The ζs are only equal for different j if the image points they are calculated from are perfect. For real data they are not.

If two parallel world lines are projected onto an image plane, their projections are only parallel if the image plane is parallel to the world lines. The intersection point of the projections of two parallel world lines is called a *vanishing point* (see figure 5.3).

Two parallel world lines meet at infinity. In projective space $\mathcal{P}^3$ this may be expressed by saying that the intersection point of two parallel world lines lies on P_∞. Points on P_∞ may also be interpreted as directions. Therefore, intersecting a line with P_∞ gives its direction. In this light, a vanishing point is the projection of the intersection point of two parallel lines. Or, in other words, it is the projection of a direction.

If we knew three vanishing points which are projections of three mutually orthogonal directions, we would know how a basis for the underlying Euclidean space $\mathcal{E}^3$ projects onto the camera used. This information can be used to find the internal camera calibration [4]. Here our initial goal is to find the *relative* camera calibration of the two cameras. We can then find an affine reconstruction. To achieve this, we do not require the vanishing points to relate to orthogonal directions. However, the more mutually orthogonal the directions related to the vanishing points are, the better the reconstruction will work.

5.5.1 Calculating Vanishing Points

Before we go any further with the actual reconstruction algorithm, let us take a look at how to calculate the vanishing points.

Vanishing Points from Parallel Line Pairs. Suppose we have two image point pairs $\{\bar{\alpha}^i_{u1}, \bar{\alpha}^i_{u2}\}$ and $\{\bar{\alpha}^i_{v1}, \bar{\alpha}^i_{v2}\}$, defining two lines on image plane A, which are projections of two parallel world lines. The vanishing point is the intersection of lines L_u and L_v where

$$L_u = \lambda^u_i L^i_a \quad ; \quad L_v = \lambda^v_i L^i_a \tag{5.38}$$

and

$$\lambda^u_{i_1} \equiv \bar{\alpha}^{i_2}_{u1}\bar{\alpha}^{i_3}_{u2} - \bar{\alpha}^{i_3}_{u1}\bar{\alpha}^{i_2}_{u2} \quad ; \quad \lambda^v_{i_1} \equiv \bar{\alpha}^{i_2}_{v1}\bar{\alpha}^{i_3}_{v2} - \bar{\alpha}^{i_3}_{v1}\bar{\alpha}^{i_2}_{v2} \tag{5.39}$$

are the homogeneous line coordinates. Also note that $L^{i_1}_a \equiv A_{i_2} \wedge A_{i_3}$ (see chapter 3). Since lines L_u and L_v lie on image plane A we know that their join is simply $J = A_1 \wedge A_2 \wedge A_3$. Therefore, $J^{-1} = A^3 \wedge A^2 \wedge A^1$. The intersection point X^a_{uv} of lines L_u and L_v is then given by

$$\begin{aligned}
X^a_{uv} &= L_u \vee L_v \\
&= \Big(L_u \cdot (A^3 \wedge A^2 \wedge A^1)\Big) \cdot L_v \\
&= \sum_{i_1, j_1} \lambda^u_{i_1}\, \lambda^v_{j_1}\, \Big((A_{i_2} \wedge A_{i_3}) \cdot (A^3 \wedge A^2 \wedge A^1)\Big) \cdot (A_{j_2} \wedge A_{j_3}) \\
&= \sum_{i_1, j_1} \lambda^u_{i_1}\, \lambda^v_{j_1}\, A^{i_1} \cdot (A_{j_2} \wedge A_{j_3}) \qquad (5.40) \\
&= \sum_{i_1} (\lambda^v_{i_2}\, \lambda^u_{i_3} - \lambda^v_{i_3}\, \lambda^u_{i_2}) A_{i_1} \\
&= \alpha^i_{uv} A_i \;; \quad \alpha^{i_1}_{uv} \equiv (\lambda^v_{i_2}\, \lambda^u_{i_3} - \lambda^v_{i_3}\, \lambda^u_{i_2})
\end{aligned}$$

First of all note that the $\{\alpha^i_{uv}\}$ give a point in $\mathcal{P}^2$. Since we defined A_1 and A_2 to be directions, the image point coordinates $\{x, y\}$ in $\mathcal{E}^2$ corresponding to the $\{\alpha^i_{uv}\}$, are found to be $\{\bar{\alpha}^1_{uv}, \bar{\alpha}^2_{uv}\}$ through the projective split, where $\bar{\alpha}^i_{uv} \equiv \alpha^i_{uv}/\alpha^3_{uv}$. Note that points which lie at infinity in $\mathcal{E}^2$ can be expressed in $\mathcal{P}^2$ by points which have a zero third component. Such points will also be called directions.

The fact that points at infinity in $\mathcal{E}^2$ are nothing special in $\mathcal{P}^2$ shows an immediate advantage of using homogeneous coordinates for the intersection points over using 2D-coordinates. Since we are looking for the intersection point of the projections of two parallel world lines, it may so happen, that the projections are also parallel, or nearly parallel. In that case, the 2D image point coordinates of the vanishing point would be very large or tend to infinity. This, however, makes them badly suited for numerical calculations. When using homogeneous coordinates, on the other hand, we do not run into any such problems.

A Closer Look at Vanishing Points. It will be instructive to see what the homogeneous intersection point coordinates look like for certain sets of lines. We rewrite the $\{\alpha^i_{uv}\}$ from above as

$$\alpha^1_{uv} = \lambda^v_2 \lambda^u_3 - \lambda^v_3 \lambda^v_2 = \bar{\alpha}^1_{v1} \bar{\alpha}^1_{v2}\, d^3_v d^1_u - \bar{\alpha}^1_{u1} \bar{\alpha}^1_{u2}\, d^3_u d^1_v \qquad (5.41a)$$

$$\alpha^2_{uv} = \lambda^v_3 \lambda^u_1 - \lambda^v_1 \lambda^v_3 = \bar{\alpha}^1_{v1} \bar{\alpha}^1_{v2}\, d^3_v d^2_u - \bar{\alpha}^1_{u1} \bar{\alpha}^1_{u2}\, d^3_u d^2_v \qquad (5.41b)$$

$$\alpha^3_{uv} = \lambda^v_1 \lambda^u_2 - \lambda^v_2 \lambda^v_1 = d^1_u d^1_v \left(\frac{d^2_u}{d^1_u} - \frac{d^2_v}{d^1_v} \right) \qquad (5.41c)$$

with

$$d_u^1 \equiv \bar{\alpha}_{u1}^1 - \bar{\alpha}_{u2}^1 \ ; \quad d_u^2 \equiv \bar{\alpha}_{u1}^2 - \bar{\alpha}_{u2}^2 \ ; \quad d_u^3 \equiv \frac{\bar{\alpha}_{u2}^2}{\bar{\alpha}_{u2}^1} - \frac{\bar{\alpha}_{u1}^2}{\bar{\alpha}_{u1}^1} \tag{5.42a}$$

$$d_v^1 \equiv \bar{\alpha}_{v1}^1 - \bar{\alpha}_{v2}^1 \ ; \quad d_v^2 \equiv \bar{\alpha}_{v1}^2 - \bar{\alpha}_{v2}^2 \ ; \quad d_v^3 \equiv \frac{\bar{\alpha}_{v2}^2}{\bar{\alpha}_{v2}^1} - \frac{\bar{\alpha}_{v1}^2}{\bar{\alpha}_{v1}^1} \tag{5.42b}$$

d_u^1 and d_u^2 define the direction of the line L_u in $\mathcal{E}^2$. Hence, $d_u^1 = 0$ if L_u is parallel to A_2, and $d_u^2 = 0$ if L_u is parallel to A_1.

$\bar{\alpha}_{u1}^2/\bar{\alpha}_{u1}^1$ gives the gradient of the line from the origin of $\mathcal{E}^2$ to the point $\{\bar{\alpha}_{u1}^1, \bar{\alpha}_{u2}^2\}$. Therefore, d_u^3 gives the difference in gradients between the lines passing through the origin of $\mathcal{E}^2$ and $\{\bar{\alpha}_{u1}^1, \bar{\alpha}_{u1}^2\}$ or $\{\bar{\alpha}_{u2}^1, \bar{\alpha}_{u2}^2\}$, respectively. Hence, $d_u^3 = 0$ if the line L_u passes through the origin of $\mathcal{E}^2$.

Now we can see the effect of some special cases of line sets $\{L_u, L_v\}$ on the homogeneous coordinates of their intersection points.

1. If L_u and L_v are parallel then $\alpha_{uv}^3 = 0$. That is, the homogeneous coordinates of their intersection point are of the form $\{x, y, 0\}$.

2. If L_u and L_v are both parallel to the A_1 direction, i.e. $\bar{\alpha}_{u1}^2 = \bar{\alpha}_{u2}^2$ and $\bar{\alpha}_{v1}^2 = \bar{\alpha}_{v2}^2$, then $d_u^2 = d_v^2 = 0$. Thus, the intersection point of L_u and L_v in homogeneous coordinates is of the form $\{x, 0, 0\}$.

3. If L_u and L_v are both parallel to the A_2 direction, i.e. $\bar{\alpha}_{u1}^1 = \bar{\alpha}_{u2}^1$ and $\bar{\alpha}_{v1}^1 = \bar{\alpha}_{v2}^1$, then $d_u^1 = d_v^1 = 0$. Then the intersection point of L_u and L_v in homogeneous coordinates is of the form $\{0, y, 0\}$.

4. If L_u and L_v both pass through the origin of $\mathcal{E}^2$, i.e. the principal point of the image plane, then the homogeneous coordinates of their intersection point are of the form $\{0, 0, w\}$.

From these special cases it becomes clear that if we knew three vanishing points on image plane A of the types 2, 3 and 4 from above, and also knew how these vanishing points project onto image plane B, we would know the 3×3 camera matrix minor $K_i^{j\,\mathsf{b}}$ of camera B. This may be seen as follows.

A Special Set of Vanishing Points. Let $\{\alpha_n^i, \beta_n^j\}$ be three sets of matching vanishing points in cameras A and B. That is, the direction that projects to $\alpha_n^i A_i$ on camera A, projects to $\beta_n^j B_j$ on camera B. Now suppose that

$\alpha_1^i = \{\alpha_1^1, 0, 0\}$, $\alpha_2^i = \{0, \alpha_2^2, 0\}$ and $\alpha_3^i = \{0, 0, \alpha_3^3\}$. Then we can write

$$\begin{aligned} & \beta_n^j = \alpha_n^i \; A_i \cdot B^j \\ \iff \quad & \beta_1^j = \alpha_1^1 \; A_1 \cdot B^j \; ; \quad \beta_2^j = \alpha_2^2 \; A_2 \cdot B^j \; ; \quad \beta_3^j = \alpha_3^3 \; A_3 \cdot B^j \\ \iff \quad & \frac{\beta_n^j}{\alpha_n^n} = A_n \cdot B^j \end{aligned} \tag{5.43}$$

Note that this calculation does not give the internal calibration of the cameras. It only tells us how the two camera frames are related. However, if we take the $\{A_\mu\}$ frame as the world frame and if we furthermore knew the correct scale of the epipole on camera B, we could use it in conjunction with $K_i^{j\,\mathsf{b}}$ to find the depths of image point matches in the A-frame. The problem with this approach, apart from the unknown scale of the epipole, is that vanishing points of the form needed are hard to obtain in real life situations.

Vanishing Points from Multiple Parallel Lines. Above we described how to find a vanishing point from the projections of two parallel world lines. In practical applications the lines will only be known with a finite precision and will also be subject to a measurment error. Therefore, we could improve on the quality of a vanishing point if sets of more than two parallel lines are known. In particular, the vanishing point quality is improved if these parallel lines are taken from varying depths within in world scene.

How can we find the vanishing point of more than two projections of parallel lines? Let the $\{L_n\}$ be a set of N projections of parallel world lines onto image plane A. We are looking for the point X_v that is closest to the intersection points of all lines. X_v is an intersection point of all lines $\{L_n\}$ if $L_n \wedge X_v = 0$ for all n. That is, finding the best vanishing point means minimising a measure ε which is given by

$$\varepsilon^2 = \sum_n \Big((L_n \wedge X_v) \cdot P_a^{-1} \Big)^2 \tag{5.44}$$

where $P_a^{-1} \equiv (A_1 \wedge A_2 \wedge A_3)^{-1}$. Note that if $L_n \wedge X_v$ is not zero, it gives a scalar multiple of $P_a = A_1 \wedge A_2 \wedge A_3$. That is, in general we have $L_n \wedge X_v = \tau_n \, P_a$, where τ_n is a scalar. The closer X_v is to line L_n, the smaller is τ_n. Since we are interested in minimising τ_n, we take the inner product of $L_n \wedge X_v$ with P_a^{-1}, which cancels the P_a and leaves us with τ_n. We then take the sum of the squares of τ_n for all n to obtain an overall quality measure.

Let $L_n = \lambda_{ni} L_a^i$ and $X_v = \alpha_v^i A_i$, then

$$\begin{aligned}(L_n \wedge X_v) \cdot P_a^{-1} &= \sum_{i_1,j} \lambda_{ni_1} \alpha_v^j \left((A_{i_2} \wedge A_{i_3}) \wedge A_j \right) \cdot P_a^{-1} \\ &= \lambda_{ni}\, \alpha_v^i \left((A_1 \wedge A_2 \wedge A_3) \cdot (A^3 \wedge A^2 \wedge A^1) \right) \\ &= \lambda_{ni}\, \alpha_v^i \end{aligned} \tag{5.45}$$

The above expression is equivalent to taking the inner product of vectors[9] $\vec{l_n} \equiv [\lambda_{n1}, \lambda_{n2}, \lambda_{n3}]$ and $\vec{v} \equiv [\alpha_v^1, \alpha_v^2, \alpha_v^3]$. If we define a matrix $\Lambda \equiv [\vec{l_1}, \vec{l_2}, \ldots, \vec{l_N}]$, then $\vec{v}$ has to minimise the length of the vector $\vec{\varepsilon}$ given by

$$\vec{\varepsilon} = \Lambda\, \vec{v}^{\,T} \tag{5.46}$$

$\vec{\varepsilon}$ is related to the measure ε^2, which we try to minimise, via

$$\varepsilon^2 = \vec{\varepsilon}^{\,2} = \vec{v}\ (\Lambda^T \Lambda)\ \vec{v}^{\,T} \tag{5.47}$$

To find the best $\vec{v}$ we can now simply perform a *Singular Value Decomposition* (SVD) on the matrix $\Lambda^2 = \Lambda^T \Lambda$. This will give us the $\vec{v}$ that minimises $\Lambda^2\, \vec{v}^{\,T}$ and thus minimises ε^2. That is, we found the best fitting vanishing point in homogeneous coordinates, in the least squares sense.

Note that in [3] vanishing points are found as 2D-image point coordinates, which means that only parallel world lines can be used that are not parallel in the image. In [4] the projections of at least three parallel world lines have to be known to calculate a vanishing point. The implementation of our algorithm switches automatically between finding a vanishing point from two parallel lines, and calculating it from multiple parallel lines, depending on how much information is available.

5.5.2 Ψ^{∞} or M^{∞} from Vanishing Points

Now we return to our reconstruction algorithm. We discussed vanishing points since they are projections of points on P_∞. If we know three vanishing point matches over cameras A and B and the epipoles, we can calculate the P_∞-collineation matrix Ψ^{∞}, or tensor M^{∞}. Once we have Ψ^{∞} or M^{∞} we can

[9] We use here the notation with the arrow above a letter to describe a vector in some orthonormal frame. This notation is used to distinguish these vectors from vectors in $\mathcal{E}^3$.

find the projections of some image points $\{\bar{\alpha}^i_n\}$ on image plane A, onto image plane B under the P_∞-collineation. That is,

$$\bar{\beta}^k_{n\,\infty} \simeq \bar{\alpha}^i_n \, \Psi^{k\,\infty}_{\;i} \,, \qquad \text{or} \qquad \bar{\beta}^k_{n\,\infty} \simeq \bar{\alpha}^i_n \, \bar{\alpha}^j_n \, M^{k\,\infty}_{ij} \tag{5.48}$$

We can now use the $\{\bar{\beta}^j_{n\,\infty}\}$ to find the $\{\zeta_n\}$ for equation (5.36). It is worth repeating this equation.

$$\bar{\alpha}^4_n = \bar{\alpha}^i_n \, \bar{K}^{3\,\mathsf{b}}_{i} \, \zeta_{1n} - \zeta_{2n} \tag{5.49}$$

with

$$\zeta_{1n} \equiv \frac{\bar{\beta}^j_{n\,\infty} - \bar{\beta}^j_n}{\bar{\beta}^j_n - \bar{\varepsilon}^j_{ba}} \quad ; \quad \zeta_{2n} \equiv \frac{\bar{\beta}^j_{n\,\infty} - \bar{\varepsilon}^j_{ba}}{\bar{\beta}^j_n - \bar{\varepsilon}^j_{ba}} \tag{5.50}$$

where j is either 1 or 2.

5.6 The Reconstruction Algorithm

Now that we have found $M^{\,\infty}$ and thus can calculate the $\{\zeta_n\}$ from equation (5.49), we can think about how to find the correct depth values for the image point matches $\{\bar{\alpha}^i_n, \bar{\beta}^j_n\}$.

We will perform an affine reconstruction. That is, we reconstruct in the frame of camera A. When we plot our final reconstructed points we will assume that the A-frame forms an orthonormal frame of $\mathcal{E}^3$, though. However, we do not need to assume anything about the frame of camera B, since we will find the translation, rotation and internal calibration of camera B *relative* to camera A. To find the internal calibration of camera A relative to an orthonormal frame of $\mathcal{E}^3$, we would need to know the projection of this orthonormal set of directions onto camera A [4].

We have already found sets of parallel lines to calculate vanishing points. We can reuse these sets of lines to constrain the depth values found with equation (5.49). In particular, we will regard the $\{\bar{K}^{3\,\mathsf{b}}_{i}\}$ as free parameters. If we now take the image point matches that define the projections of two parallel world lines, we can use this extra information to constrain the $\{\bar{K}^{3\,\mathsf{b}}_{i}\}$. That is, we vary the free parameters until the reconstructed points define a pair of parallel world lines again.

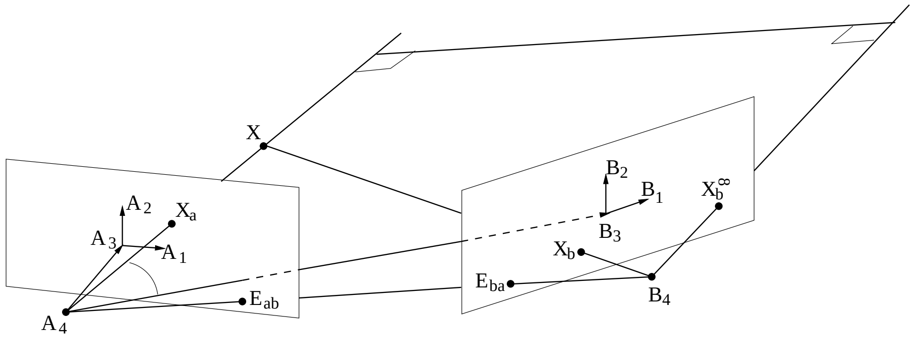


Figure 5.4: This figure shows the geometry behind equation (5.49). A point X is projected onto cameras A and B, giving images X_a and X_b, respectively. Projecting X_a onto image plane B under the P_∞-collineation gives X_b^∞. We choose A_4 to be the origin of $\mathcal{E}^3$. $K_i^{3\,\mathsf{b}}$ gives the components of A_1, A_2 and A_3 along B_3.

5.6.1 The Geometry

Before we start developing an algorithm to find the best $\{\bar{K}_i^{3\,\mathsf{b}}\}$ we should understand what varying the free parameters means geometrically. In figure 5.4 we have drawn the geometry underlying our reconstruction algorithm.

A_4 and B_4 are the optical centres of cameras A and B, respectively. We have also chosen A_4 to lie at the origin of $\mathcal{E}^3$. Recall that A_1, A_2 and B_1, B_2 are direction vectors in $\mathcal{P}^3$. We have drawn these vectors here as lying on the image planes to indicate this.

A world point X is projected onto image planes A and B giving projections X_a and X_b, respectively. X_b^∞ is the projection of X_a onto image plane B under the P_∞-collineation. Also, E_{ba} is the epipole of camera B.

Now we can see what the $\{\zeta_{1n}, \zeta_{2n}\}$ components from equation (5.49) express.

$$\zeta_{1n} \equiv \frac{\bar{\beta}^j_{n\,\infty} - \bar{\beta}^j_n}{\bar{\beta}^j_n - \bar{\varepsilon}^j_{ba}}$$

gives the ratio of the distance (in x or y direction) between X_b^∞ and X_b, and X_b and E_{ba}.

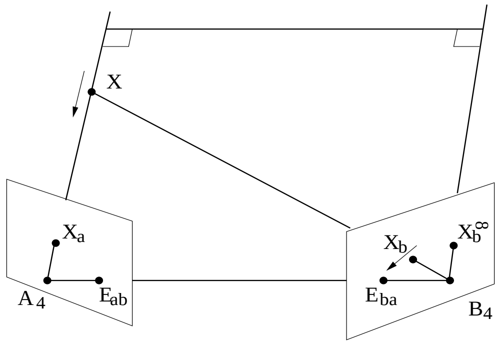
$$\zeta_{2n} \equiv \frac{\bar{\beta}^j_{n\,\infty} - \bar{\varepsilon}^j_{ba}}{\bar{\beta}^j_n - \bar{\varepsilon}^j_{ba}}$$

gives the ratio of the distance (in x or y direction) between X_b^∞ and E_{ba}, and X_b and E_{ba}.

Some Special Cases.

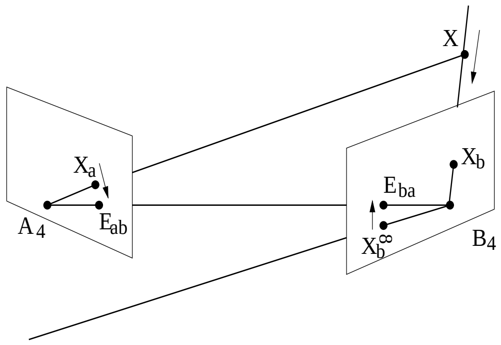
1. If $X_b = X_b^\infty$ then X is a point on P_∞, for example a vanishing point. In this case $\zeta^j_{1n} = 0$ and $\zeta^j_{2n} = 1$. Hence, equation (5.49) gives $\bar{\alpha}^4_n = -1$. From equation (5.5) it follows that this does indeed give a point on P_∞.

2.



Suppose we move a world point X along a line towards A_4 (see figure). In that case, X_a and X_b^∞ will stay constant, but X_b will move towards E_{ba}. Therefore, $\zeta^j_{1n} \to \infty$ and $\zeta^j_{2n} \to \infty$, and thus $\bar{\alpha}^4 \to \infty$ (from equation (5.49)) and $\hat{\alpha}^i \to 0$. That is, in the limit that $X = A_4$, X will be reconstructed to lie at the origin, which is A_4.

3.



Now suppose we move a world point X along a line towards B_4 (see figure). Now X_b will stay constant, whereas $X_a \to E_{ab}$ and $X_b^\infty \to E_{ba}$. Therefore, $\zeta^j_{1n} \to -1$ and $\zeta^j_{2n} \to 0$. In the limit that $X = B_4$ we have $X_a = E_{ab}$ and thus equation (5.49) gives $\bar{\varepsilon}^4_{ab} = -\bar{\varepsilon}^i_{ab}\,\bar{K}^{3\,\mathsf{b}}_{i}$. That is, we have found the position of B_4 in the A-frame.

The equation for the projective depth of the $\{\bar{\varepsilon}^i_{ab}\}$ given above can also be derived as follows.

$$\begin{aligned}
& B_4 = B_4{\cdot}A^\mu\; A_\mu = \varepsilon^\mu_{ab}\; A_\mu \\
\Rightarrow\quad & B_4{\cdot}B^j = \varepsilon^\mu_{ab}\; A_\mu{\cdot}B^j \\
\Rightarrow\quad & 0 = \varepsilon^\mu_{ab}\; K^{j\,\mathsf{b}}_{\mu}\;; \quad K^{j\,\mathsf{b}}_{\mu} \equiv A_\mu{\cdot}B^j \\
\Rightarrow\quad & \varepsilon^4_{ab} = -\varepsilon^i_{ab}\frac{K^{j\,\mathsf{b}}_{i}}{K^{j\,\mathsf{b}}_{4}}
\end{aligned} \tag{5.51}$$

What we have done here is first to express B_4 in the A-frame. Then we project B_4 onto image plane B. In the third line we use the fact that $B_4{\cdot}B^j = 0$ by definition. The resultant equation has to be valid for each $j \in \{1, 2, 3\}$. If we choose $j = 3$ we can write

$$\bar{\varepsilon}^4_{ab} = -\bar{\varepsilon}^i_{ab}\,\bar{K}^{3\,\mathsf{b}}_{i}. \tag{5.52}$$

The Meaning of the Free Parameters. Let us return to figure 5.4. Recall that $K_i^{3\,\mathsf{b}} = A_i \cdot B^3$, that is, it gives the components of the $\{A_i\}$ along B_3. Therefore, varying the $\{K_i^{3\,\mathsf{b}}\}$ means that we are moving B_3, which is the principal point on image plane B. Since X_b^∞ cannot change when we vary $K_i^{3\,\mathsf{b}}$ the relation between B_3 and B_4 is fixed. Thus, changing B_3 means changing B_4. In this respect, finding the correct $\{K_i^{3\,\mathsf{b}}\}$ means finding the correct translation of camera B relative to camera A. The relative rotation has already been fixed through finding P_∞.

However, it is the relative sizes of the $\{\bar{K}_i^{3\,\mathsf{b}}\}$ that are really important. An overall scale factor will only change the depths of all reconstructed points simultaneously. Therefore, we can fix the depth of one image point, to fix the scale of $\bar{K}_i^{3\,\mathsf{b}}$.

5.6.2 The Minimisation Function

We mentioned before that we will use our knowledge of parallel lines once again to constrain the $\{\bar{K}_i^{3\,\mathsf{b}}\}$ from equation (5.49). Let $L_u^a = X_{u1}^a \wedge X_{u2}^a$ and $L_v^a = X_{v1}^a \wedge X_{v2}^a$ be the projections of two parallel world lines onto image plane A. In general we define world points and their corresponding images on image plane A as

$$\left.\begin{array}{ll} X_{ur} \equiv \bar{\alpha}_{ur}^{\mu} A_\mu \; ; & X_{ur}^a \equiv \bar{\alpha}_{ur}^{i} A_i \\[2ex] X_{vr} \equiv \bar{\alpha}_{vr}^{\mu} A_\mu \; ; & X_{vr}^a \equiv \bar{\alpha}_{vr}^{i} A_i \end{array}\right\} r \in \{1, \ldots, n\}. \tag{5.53}$$

Furthermore, if we know the image points on image plane B corresponding to X_{u1}^a, X_{u2}^a, X_{v1}^a and X_{v2}^a, and we have found M^{∞}, then we can calculate the corresponding ζs from equation (5.50). Equation (5.49) will now allow us to find the projective depths for X_{u1}^a, X_{u2}^a, X_{v1}^a and X_{v2}^a. Therefore, we can calculate the world lines $L_u = X_{u1} \wedge X_{u2}$ and $L_v = X_{v1} \wedge X_{v2}$.

Now, we know that L_u and L_v are supposed to be parallel, which means that they have to intersect P_∞ in the same point. This will be the constraint which we will use to find the correct $\{\bar{K}_i^{3\,\mathsf{b}}\}$. Let X_u^∞ and X_v^∞ be defined as

$$X_u^\infty \equiv L_u \vee P_\infty \; ; \quad X_v^\infty \equiv L_v \vee P_\infty. \tag{5.54}$$

Lines L_u and L_v are parallel iff

$$X_u^\infty \wedge X_v^\infty = 0 \tag{5.55}$$

Instead of using this condition we could also project L_u and L_v into $\mathcal{E}^3$, and then check that they are parallel. However, projecting into $\mathcal{E}^3$ means dividing through by the projective depth, which means that our free parameters are now in the denominator of a minimisation function. Apart from creating a minimisation surface with singularities, the derivatives of such a minimisation function will be more complicated and thus cost more computing time.

Finding the minimisation parameters. We will now derive an expression for X_u^∞ in terms of the $\{\bar{\alpha}_{u1}^i, \bar{\alpha}_{u2}^i\}$.

$$\begin{aligned} X_u^\infty &= L_u \vee P_\infty = (\bar{\alpha}_{u1}^\mu\, \bar{\alpha}_{u2}^\nu\, A_\mu \wedge A_\nu) \vee P_\infty \\ &= \Big(\bar{\lambda}_{\mu_1\mu_2}^u\, A_{\mu_1} \wedge A_{\mu_2}\Big) \vee \Big(A_1 \wedge A_2 \wedge (A_3 - A_4)\Big) \\ &= \quad \bar{\lambda}_{\mu_1\mu_2}^u \Big(\; [\![A_{\mu_1} A_{\mu_2} A_1 A_2]\!]\, A_3 + [\![A_{\mu_1} A_{\mu_2} A_3 A_1]\!]\, A_2 \\ &\qquad\qquad + [\![A_{\mu_1} A_{\mu_2} A_2 A_3]\!]\, A_1\Big) \\ &\quad - \; \bar{\lambda}_{\mu_1\mu_2}^u \Big(\; [\![A_{\mu_1} A_{\mu_2} A_1 A_2]\!]\, A_4 + [\![A_{\mu_1} A_{\mu_2} A_4 A_1]\!]\, A_2 \\ &\qquad\qquad + [\![A_{\mu_1} A_{\mu_2} A_2 A_4]\!]\, A_1\Big) \\ &\simeq (\bar{\lambda}_{13}^u + \bar{\lambda}_{14}^u)\, A_1 + (\bar{\lambda}_{23}^u + \bar{\lambda}_{24}^u)\, A_2 + \bar{\lambda}_{34}^u\, (A_3 - A_4) \\ &= \chi_u^i\, A_i^\infty \end{aligned} \tag{5.56}$$

where

$$\chi_u^i \equiv (\bar{\lambda}_{i3}^u + \bar{\lambda}_{i4}^u)\;; \quad \bar{\lambda}_{\mu_1\mu_2}^u \equiv \bar{\alpha}_{u1}^{\mu_1}\, \bar{\alpha}_{u2}^{\mu_2} - \bar{\alpha}_{u1}^{\mu_2}\, \bar{\alpha}_{u2}^{\mu_1}, \tag{5.57}$$

and

$$A_1^\infty \equiv A_1\;; \quad A_2^\infty \equiv A_2\;; \quad A_3^\infty \equiv A_3 - A_4 \tag{5.58}$$

The free parameters we have are the $\{\bar{K}_i^{3\,\mathsf{b}}\}$. To make future equations somewhat clearer we will define $\varphi_i \equiv \bar{K}_i^{3\,\mathsf{b}}$. Hence, equation (5.49) will be written as

$$\bar{\alpha}_n^4 = \bar{\alpha}_n^i\, \zeta_{1n}\, \varphi_i - \zeta_{2n}. \tag{5.59}$$

Substituting equation (5.59) into equation (5.57) gives

$$\begin{aligned}
\chi^i_u &= \bar{\alpha}^i_{u1}\,\bar{\alpha}^4_{u2} - \bar{\alpha}^i_{u2}\,\bar{\alpha}^4_{u1} + \bar{\alpha}^i_{u1} - \bar{\alpha}^i_{u2} \\
&= \qquad \bar{\alpha}^i_{u1}\,(\bar{\alpha}^j_{u2}\,\zeta_{1\,u2}\,\varphi_j - \zeta_{2\,u2}) \\
&\quad - \;\; \bar{\alpha}^i_{u2}\,(\bar{\alpha}^j_{u1}\,\zeta_{1\,u1}\,\varphi_j - \zeta_{2\,u1}) \;+\; \bar{\alpha}^i_{u1} - \bar{\alpha}^i_{u2} \\
&= \Big(\zeta_{1\,u2}\,\bar{\alpha}^i_{u1}\,\bar{\alpha}^j_{u2} - \zeta_{1\,u1}\,\bar{\alpha}^i_{u2}\,\bar{\alpha}^j_{u1}\Big)\,\varphi_j \\
&\quad +\bar{\alpha}^i_{u1}\,(1-\zeta_{2\,u2}) - \bar{\alpha}^i_{u2}\,(1-\zeta_{2\,u1})
\end{aligned} \tag{5.60}$$

This equation may be written more succinctly as

$$\chi^i_u = D^{ij}_u\,\varphi_j + p^i_u \tag{5.61}$$

with

$$\begin{aligned}
&D^{ij}_u \equiv \zeta_{1\,u2}\,\bar{\alpha}^i_{u1}\,\bar{\alpha}^j_{u2} - \zeta_{1\,u1}\,\bar{\alpha}^i_{u2}\,\bar{\alpha}^j_{u1} \\
&p^i_u \equiv \bar{\alpha}^i_{u1}\,(1-\zeta_{2\,u2}) - \bar{\alpha}^i_{u2}\,(1-\zeta_{2\,u1})
\end{aligned} \tag{5.62}$$

The reason for defining the $\{D^{ij}_u\}$ and $\{p^i_u\}$ is, that they can be calculated for every parallel line pair available *before* we minimise over the $\{\varphi_j\}$. In this way we reduce the calculation time at each minimisation step.

Recall that lines L_u and L_v are parallel iff $X^\infty_u \wedge X^\infty_v = 0$. We can now write this expression in terms of the $\{\chi^i\}$.

$$\begin{aligned}
X^\infty_u \wedge X^\infty_v &= (\chi^i_u\,A^\infty_i)\wedge(\chi^j_v\,A^\infty_j) \\
&= \sum_{i_1} (\chi^{i_2}_u\,\chi^{i_3}_v - \chi^{i_3}_u\,\chi^{i_2}_v)\;A^\infty_{i_2}\wedge A^\infty_{i_3} \\
&= \Lambda^{uv}_i\,L^i_\infty\;; \quad L^{i_1}_\infty \equiv A^\infty_{i_2}\wedge A^\infty_{i_3}
\end{aligned} \tag{5.63}$$

with

$$\Lambda^{uv}_{i_1} \equiv \chi^{i_2}_u\,\chi^{i_3}_v - \chi^{i_3}_u\,\chi^{i_2}_v \tag{5.64}$$

Each of the $\{\Lambda_i^{uv}\}$ has to be zero if $X_u^\infty \wedge X_v^\infty = 0$. Therefore, from an analytical point of view, the expression we should try to minimise for each parallel line pair $\{L_u, L_v\}$ is

$$\Delta^{uv} : \varphi_j \longrightarrow \sum_{i=0}^{3} (\Lambda_i^{uv})^2. \tag{5.65}$$

Improving computational accuracy. However, for a computer with finite floating point precision, this equation poses a problem. The culprits in this case are the $\{\chi^i\}$. Recall that they give the direction of a line in homogeneous coordinates. Before they are used in equation (5.64) they should be normalised to improve the precision of the equation on a computer. We normalise the $\{\chi_u^i\}$ in the following way.

$$\hat{\chi}_u^i \equiv \frac{\chi_u^i}{\sqrt{\sum_i (\chi_u^i)^2}} \tag{5.66}$$

Therefore, the minimisation function we will use is

$$\Delta^{uv} : \varphi_j \longrightarrow \sum_{i=0}^{3} (\hat{\Lambda}_i^{uv})^2. \tag{5.67}$$

where

$$\hat{\Lambda}_{i_1}^{uv} \equiv \hat{\chi}_u^{i_2}\, \hat{\chi}_v^{i_3} - \hat{\chi}_u^{i_3}\, \hat{\chi}_v^{i_2} \tag{5.68}$$

Since the $\{\hat{\chi}_u^i\}$ are normalised it might seem possible on first sight to use as minimisation function

$$\Delta^{uv} : \varphi_j \longrightarrow \sum_{i=0}^{3} (\hat{\chi}_u^i - \hat{\chi}_v^i)^2. \tag{5.69}$$

This would be faster to calculate and also have much simpler derivatives. However, this equation is sensitive to an overall sign change of the $\{\hat{\chi}^i\}$, but we are only interested in whether two lines are parallel, not in whether the vectors that define them point in the same or in opposite directions.

The derivatives. The derivative of Δ^{uv} is computationally not a particularly expensive expression. Therefore, we can use a minimisation routine that also uses the derivatives of the minimisation function. This will make the minimisation process more efficient and robust.

The partial derivatives of the $\{\hat{\chi}^i_u\}$ are

$$\partial_{\varphi_r}\, \hat{\chi}^i_u = \frac{D^{ir}_u}{\sqrt{\sum_k (\chi^k_u)^2}} - \frac{\chi^i_u \sum_k (\chi^k_u\, D^{kr}_u)}{\sqrt[3]{\sum_k (\chi^k_u)^2}}, \tag{5.70}$$

and the partial derivatives of the $\{\hat{\Lambda}^{uv}_i\}$ are

$$\partial_{\varphi_r}\, \hat{\Lambda}^{uv}_{i_1} = (\partial_{\varphi_r}\, \hat{\chi}^{i_2}_u)\, \hat{\chi}^{i_3}_v + \hat{\chi}^{i_2}_u\, (\partial_{\varphi_r}\, \hat{\chi}^{i_3}_v) - (\partial_{\varphi_r}\, \hat{\chi}^{i_3}_u)\, \hat{\chi}^{i_2}_v - \hat{\chi}^{i_3}_u\, (\partial_{\varphi_r}\, \hat{\chi}^{i_2}_v). \tag{5.71}$$

Now we can calculate the partial derivatives of the minimisation function Δ^{uv}.

$$\partial_{\varphi_r}\, \Delta^{uv} = 2 \sum_{i=0}^{3} (\hat{\Lambda}^{uv}_i\, \partial_{\varphi_r}\, \hat{\Lambda}^{uv}_i) \tag{5.72}$$

Implementing the depth constraint. At the moment the minimisation function Δ^{uv} depends on three parameters: the $\{\varphi_j\}$. However, we mentioned earlier that we can fix the depth of one point. This will reduce the number of free parameters to two.

If we choose to fix the depth of point X_{u1} it follows from equation (5.59) that the following condition must hold.

$$\begin{aligned} 0 &= \zeta_{1\,u1}\, \bar{\alpha}^i_{u1}\, \varphi_i - \zeta_{2\,u1} - \bar{\alpha}^4_{u1} \\ &= \xi^i_{u1}\, \varphi_i - \omega_{u1} \end{aligned} \tag{5.73}$$

with

$$\xi^i_{u1} \equiv \zeta_{1\,u1}\, \bar{\alpha}^i_{u1}\ ; \quad \omega_{u1} \equiv \zeta_{2\,u1} + \bar{\alpha}^4_{u1} \tag{5.74}$$

With regard to equation (5.5) a good choice for $\bar{\alpha}^4_{u1}$ is -0.5, which means that $\hat{\alpha}^3 = 2$. However, if the point we chose to fix has a much larger depth value than the other points we are trying to reconstruct, then some points may be reconstructed to lie *behind* the optical centre. In this case, we will invariably get a bad reconstruction.

We can rewrite equation (5.73) as

$$\bar{\xi}^i_{u1}\, \varphi_i = 1\ ; \quad \bar{\xi}^i_{u1} \equiv \frac{\xi^i_{u1}}{\omega_{u1}} \tag{5.75}$$

Of course, we have to make sure that $\omega_{u1} \neq 0$, which means we have to be somewhat careful with the choice of $\bar{\alpha}^4_{u1}$.

Our minimisation routine should only search over that parameter space where equation (5.75) is satisfied. We can achieve this by reparameterising the equation.

$$\bar{\xi}^i_{u1}\,(\tau_1\,\Phi^1_i + \tau_2\,\Phi^2_i + \tau_3\,\Phi^3_i) = 1\ ; \quad \tau_3 \equiv 1 \tag{5.76}$$

with

$$\varphi_j = \tau_i\,\Phi^i_j \tag{5.77}$$

and

$$\bar{\xi}^i_{u1}\,\Phi^1_i = 0\ ; \quad \bar{\xi}^i_{u1}\,\Phi^2_i = 0\ ; \quad \bar{\xi}^i_{u1}\,\Phi^3_i = 1 \tag{5.78}$$

We have replaced the $\{\varphi_i\}$ with the $\{\tau_i\}$. Therefore, $\{\tau_1, \tau_2\}$ are now the free parameters, while τ_3 is fixed at unity.

The question now is how we can find the appropriate Φ^j_i matrix. First of all we can set the vector Φ^3_i to be the inverse of the vector $\bar{\xi}^i_{u1}$.

$$\Phi^3_i = \frac{\bar{\xi}^i_{u1}}{\sum_j (\bar{\xi}^j_{u1})^2} \quad\Longrightarrow\quad \bar{\xi}^i_{u1}\,\Phi^3_i = \frac{\sum_i (\bar{\xi}^i_{u1})^2}{\sum_j (\bar{\xi}^j_{u1})^2} = 1 \tag{5.79}$$

We find the remaining first two rows of Φ^j_i with the help of an SVD. We do this by creating a 3×3 matrix H whose three rows are all given by Φ^3_i. That is, $H \equiv [\Phi^3_i, \Phi^3_i, \Phi^3_i]$. Therefore, H is of rank 1 and has a nullity of 2. Applying an SVD to H will find a set of three orthogonal vectors, two of which span the null space of H, while the remaining one is just a scaled version of Φ^3_i. We first should find that scale and apply it to H. The null space of H is exactly the space we want our minimisation routine to search over. Hence, we set Φ^1_i and Φ^2_i to be the correctly scaled null space vectors found with the SVD. This satisfies equation (5.78), and therefore equation (5.76) will stay unchanged when varying τ_1 and τ_2.

Image Point Normalisation. Before we can calculate the collineation tensor for the P_∞-collineation we have to find the fundamental matrix (F) for the two views (see equation (5.21)). For the calculation of the F we cannot use the pixel coordinates directly, because they are typically too large to obtain good accuracy in our numerical calculations. This is also true for all other calculations performed here. Therefore, we need to scale the image point coordinates so that they are of order 1.

In [26] Hartley suggests that the scales and skews applied to the image point coordinates are found in the following way. The skew is given by the coordinates of the centroid of all image points. Then the average distance of the skewed image points from the origin is calculated. The inverse of that distance gives the scale.

This is a good method if we just wanted to calculate F. However, it turns out that for our purposes such a scaling is not suitable. In fact, we found it is important to conserve the aspect ratio of the images (separately), and to ensure that the origin of the image plane is chosen in the same way in both images.

We choose the image plane origin to be in the centre of each image plane and then scale the image points by dividing their x and y coordinate by the image resolution in the x-direction. This preserves the aspect ratio.

Calculating the Fundamental Matrix. In recent times a lot of effort has gone into the analysis of the fundamental matrix (F) and the trifocal tensor (T), in order to find constraints so that they may be calculated as accurately as possible [26, 24, 39, 48, 14, 17, 32]. However, as our experimental results will show, calculating an optimal F does not seem to be of high importance for our reconstruction algorithm. In retrospect this will justify the simple calculation method we use for F.

We calculate F with a simple SVD method by writing the components of F as a column vector $\vec{f}$. If we have N point matches $\{\bar{\alpha}^i_n, \bar{\beta}^i_n\}$ then the F we look for has to satisfy

$$\bar{\alpha}^i_n \, \bar{\beta}^j_n \, F_{ij} = 0, \tag{5.80}$$

for all n. In matrix notation this can written as

$$A \, \vec{f} = 0 \tag{5.81}$$

with

$$A \equiv \begin{pmatrix} \bar{\alpha}^1_1 \, \bar{\beta}^1_1 & \bar{\alpha}^1_1 \, \bar{\beta}^2_1 & \bar{\alpha}^1_1 \, \bar{\beta}^3_1 & \bar{\alpha}^2_1 \, \bar{\beta}^1_1 & \dots & \bar{\alpha}^3_1 \, \bar{\beta}^3_1 \\ \bar{\alpha}^1_2 \, \bar{\beta}^1_2 & \bar{\alpha}^1_2 \, \bar{\beta}^2_2 & \bar{\alpha}^1_2 \, \bar{\beta}^3_2 & \bar{\alpha}^2_2 \, \bar{\beta}^1_2 & \dots & \bar{\alpha}^3_2 \, \bar{\beta}^3_2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \bar{\alpha}^1_N \, \bar{\beta}^1_N & \bar{\alpha}^1_N \, \bar{\beta}^2_N & \bar{\alpha}^1_N \, \bar{\beta}^3_N & \bar{\alpha}^2_N \, \bar{\beta}^1_N & \dots & \bar{\alpha}^3_N \, \bar{\beta}^3_N \end{pmatrix} \tag{5.82}$$

For real data there will typically not be an F that satisfies equation (5.80) perfectly. Therefore, what we try to calculate is the F that satisfies these conditions as well as possible. This is equivalent to finding the vector that has the least influence on the range of A. This can be achieved by performing an SVD on A.

$$A = U\,D\,V^T, \tag{5.83}$$

where U and V are orthogonal matrices and D is a diagonal matrix [53]. The column vector in V that corresponds to the smallest diagonal value in D is the $\vec{f}$ that we are looking for. Note that at least nine point matches have to be known to find F in this way.

As was shown in chapter 3, F is of rank 2. An F found with the above method from real image point matches, usually does not satisfy this constraint. An indication that the image point matches are particularly bad is that there are two or more diagonal values in D of the same order of magnitude.

A linear method to enforce the rank of F is to project the initial F to the nearest F that satisfies the rank constraint. This may be done by performing an SVD on F, i.e. $F = UDV^T$, then setting the smallest diagonal value in D to zero, and recalculating $F = UDV^T$ with the changed D. However, this did not have any significant effect on our reconstructions.

5.6.3 The Minimisation Routine

We used a modified version of the *conjugate gradient* method to perform the minimisation. This modified version is called *MacOpt* and was developed by David MacKay [45]. It makes a number of improvements over the conjugate gradient method as given in [53]. We will list the most important modifications in the following.

- The initial step size in the line minimisation as given in [53] may be too big, which can result in a lot of wasted computing power. MacOpt rectifies this in two ways:

  1. the initial step size of any line minimization is inversely proportional to the gradient,
  2. the constant of proportionality is adopted during the minimisation.

- MacOpt uses the gradient information also for the line search. In this way the line minimum can be bracketed with only two gradient evaluations.

- MacOpt does not evaluate the function at all, but only uses the gradient information.
- The general purpose minimiser in [53] gives very high precision in the line minimisations, which is actually not necessary. MacOpt only brackets the minimum and then guesses where it is by linear interpolation.

To optimise the minimisation process we calculate the $\{D^{ij}\}$ and $\{p^i\}$ from equation (5.62) for each parallel line pair, *before* we start MacOpt. During the minimisation process we can then calculate the $\{\chi^i\}$ quickly with equation (5.61). Unfortunately, we cannot precalculate anything else because of the normalisation of the $\{\chi^i\}$ which we need to perform.

MacOpt assumes that the minimisation surface is fundamentally convex with no local minima. However, our surface is not of that shape. It turns out that the success rate of finding the absolute minimum can be improved if we first use the unnormalised χs to step towards the minimum, and then use the normalised χs to find the minimum with high accuracy. This is because the minimisation surface for the unnormalised χs is of a convex shape, whereas the minimisation surface for the normalised χs has a number of local minima.

The problems that may occur with MacOpt and a general discussion of the minimisation surface, in particular with relation to the reconstruction, is demonstrated by the program **MVT**. This program can be found on the enclosed CD or can be downloaded from C.Perwass' home page. For more details see appendix A.

5.7 Experimental Results

We can now outline the structure of our reconstruction algorithm.

Step 1: We find point matches and sets of projections of parallel lines over the two images.

Step 2: We calculate three vanishing points and the fundamental matrix. This allows us to find the P_∞-collineation matrix Ψ^{∞} or tensor M^{∞}.

Step 3: We select a set of parallel lines that we want to use to constrain our minimisation. Note that one pair of parallel lines may be enough. More pairs do not necessarily improve the result, since they may not be consistent due to errors.

Step 4: The image points on image plane A which define the chosen parallel lines are projected onto image plane B under the P_∞-collineation with Ψ^{∞} or M^{∞}.

Step 5: We can now find the $\{K^{3\,\mathsf{b}}_{i}\}$ by minimising equation (5.65) or equation (5.67).

Step 6: Once we have found $K^{3\,\mathsf{b}}_{i}$ we can use it in conjunction with Ψ^{∞} or M^{∞} in equation (5.59) to reconstruct any other image point matches for this camera setup. Note that this method of finding the image point depths saves us from performing an additional triangulation [25], which would be necessary if we first calculated the camera calibrations explicitly and then tried to find the image point depths.

Note that we have used M^{∞} instead of Ψ^{∞} for our calculations. This was done to avoid the problem of numerical instability when calculating Ψ^{∞} through a matrix diagonalisation.

Figure 5.5 shows the data that has to be known and calculated as input to our minimisation routine. The image points and parallel line indices are the source data. The latter index which image point pairs form parallel lines. The inputs to the minimisation routine are the image points, the parallel line indices, the epipoles and the image points projected through the plane at infinity. The fundamental matrix and the vanishing points are only intermediate calculations to find M^{∞}. M^{∞} is also only needed once to project the image points on image plane A onto image plane B under the P_∞-collineation.

There are quite a number of factors that influence the reconstruction. These are shown in figure 5.6. We can distinguish between two types of influences:

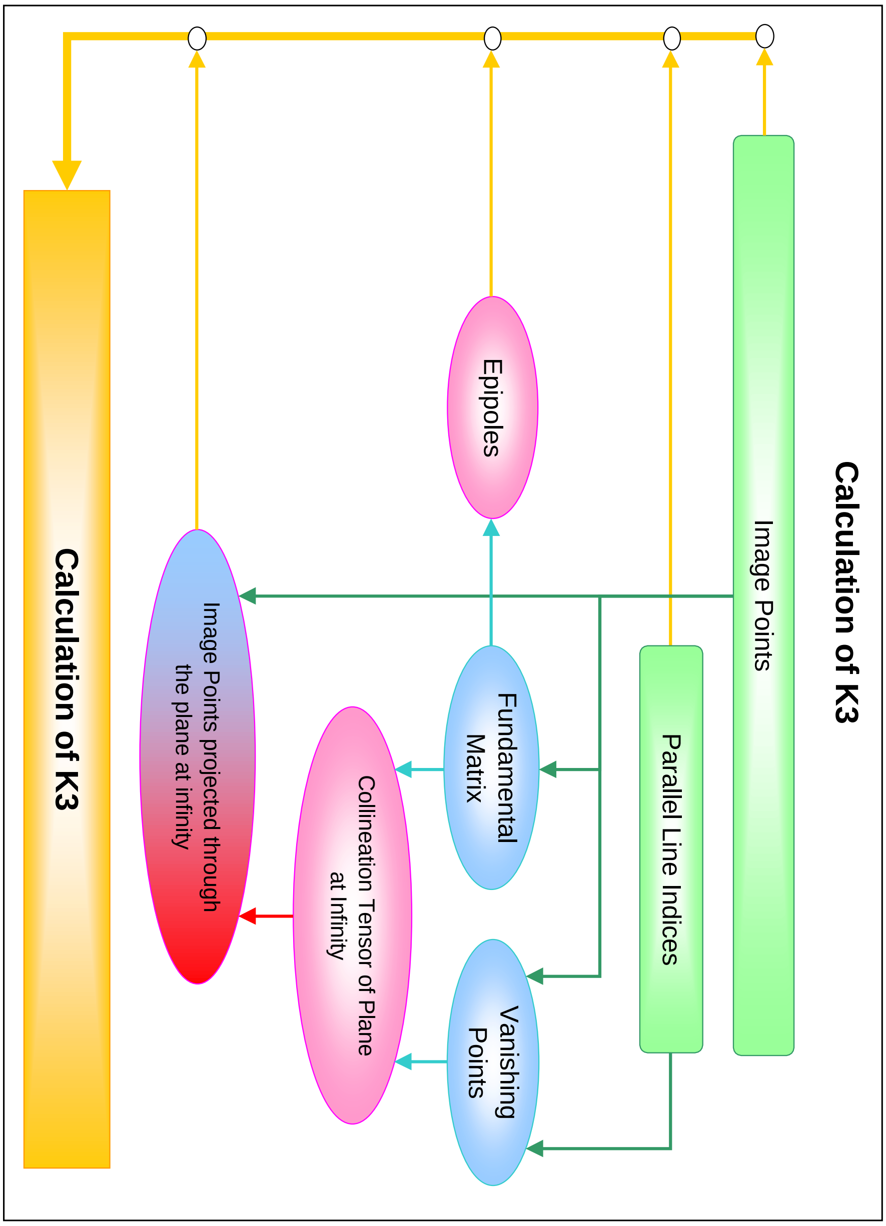


Figure 5.5: Data needed for calculation of $K^{3\,\mathsf{b}}_{j}$.

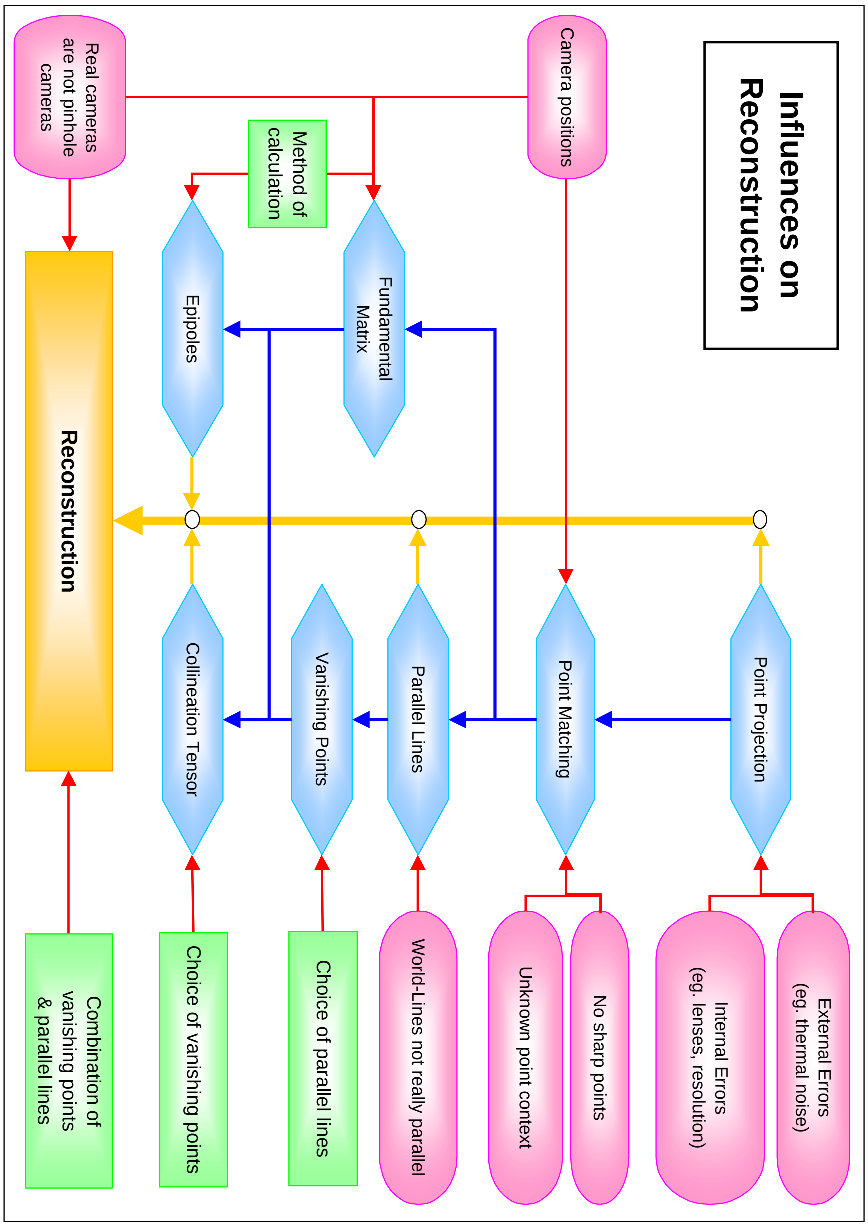

Influences on Reconstruction
Real cameras are not pinhole cameras
Camera positions
Method of calculation
Fundamental Matrix
Epipoles
Reconstruction
Point Projection
Point Matching
Parallel Lines
Vanishing Points
Collineation Tensor
External Errors (eg. thermal noise)
Internal Errors (eg. lenses, resolution)
No sharp points
Unknown point context
World-Lines not really parallel
Choice of parallel lines
Choice of vanishing points
Combination of vanishing points & parallel lines

Figure 5.6: Influences on the reconstruction. Red, rounded boxes show those influences on which we have no influence whereas green, square boxes indicate choices that can be made.

those on which we have no influence once we are given our source data (i.e. image points and parallel lines), and those which depend on how we deal with the source data. The former are indicated by red, rounded boxes and the latter by green, square ones. The pointed boxes indicate the calculations which we need to perform before we can start the minimisation routine.

If we have real data we can only try to improve the reconstruction by varying the choice of parallel lines, the choice of vanishing points and the method of calculation of the fundamental matrix and the epipoles. However, if we use synthetic data, we have a handle on all influences shown.

5.7.1 Synthetic Data

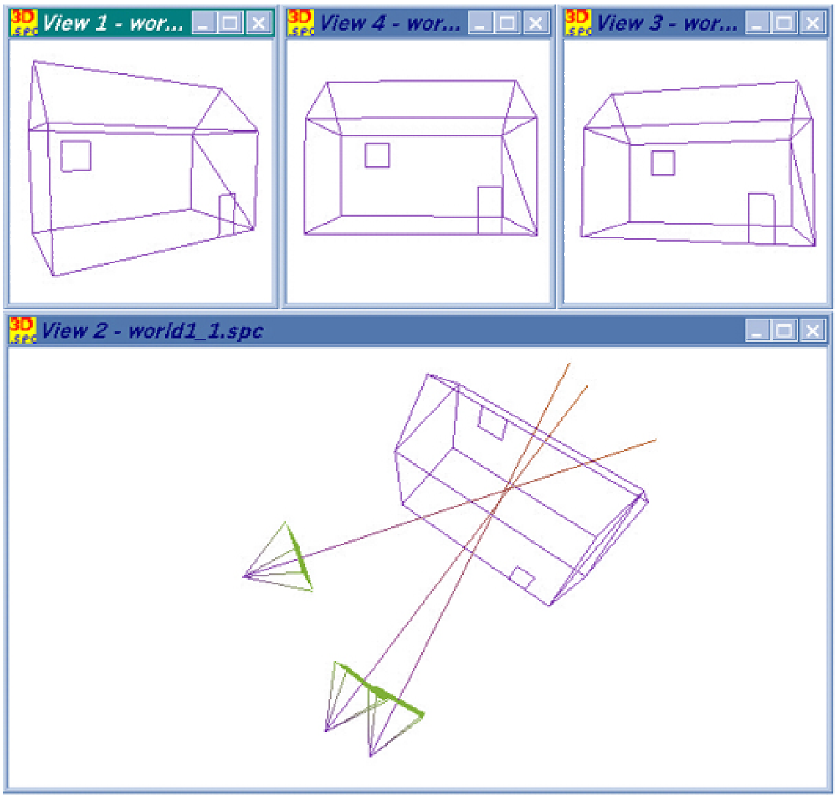


Figure 5.7: The synthetic data was created from projections of the house onto the cameras.

To test the quality of the reconstruction algorithm we use synthetic data. One big advantage of using synthetic data is that we can get a *geometric* quality measure of the reconstruction. Also if an algorithm fails with perfect synthetic data, it is clearly unlikely to work with real data.

The lower picture in figure 5.7 shows a house with three cameras. The three smaller pictures on top show the projections of the house onto the three image planes. The house consists of 18 vertices, which were all used in our calculations. We performed two trials: trial 1 uses an orthogonal set of vanishing points. Trial 2 uses two orthogonal vanishing points but the third vanishing point is found from the two lines on the roof which are vertically sloping and closest to the camera. In each trial we also tested two camera configurations: the camera to the very left and the very right, and the two cameras which are close together. The former will be called the *far cameras* and the latter the *close cameras* configuration.

Recall that we can and, in fact, have to fix the depth of one point. Since we know the true points we can set this depth to its true value. Also remember that we perform our reconstruction in the frame of one of the cameras. But we also know this frame and can therefore transform our reconstructed points to lie in the appropriate frame. The reconstruction obtained in this way can then be compared *directly* with the true object.

In our experiments we added a Gaussian error with a mean deviation between 0 and 12 pixels to the image points. The camera resolutions were 600 × 600 pixels. For each setting of the mean deviation of the induced error we calculated the $\{K_i^{3\,\mathsf{b}}\}$ 100 times, each time with different errors, to obtain a statistically meaningful result. Each calculation of the $\{K_i^{3\,\mathsf{b}}\}$ can be used to reconstruct any image point matches in the two images. Therefore, we projected the house again onto the two image planes, again introducing an error of the same mean deviation. These image points are then reconstructed and compared with the true points. This was done 20 times for each calculation of the $\{K_i^{3\,\mathsf{b}}\}$. This way we obtained a separation of the calibration and the reconstruction.

The quality measure of a reconstruction is given by the root mean squared error between the locations of the reconstructed points and the true points. That is, we take the root of the mean of the sum of the distances squared between the true and the reconstructed points. We evaluated the RMS error over the 20 reconstructions for each calibration (i.e. calculation of the $\{K_i^{3\,\mathsf{b}}\}$), and also over all calculations of the $\{K_i^{3\,\mathsf{b}}\}$ for each mean deviation of the induced error. The former will be called the "RMS/Trial" and the latter the "Total RMS".

Figure 5.8 shows the results when using an orthogonal set of vanishing points and figure 5.9 when using a non-orthogonal set, as described above. Note that the y-axis has a $\log_{10}$ scale. The length of the house is 2 units, its total height 1.5 units and its depth 1 unit. The results for the close cameras configuration are slighty displaced to the right, so that they can be distinguished from the far cameras setup.

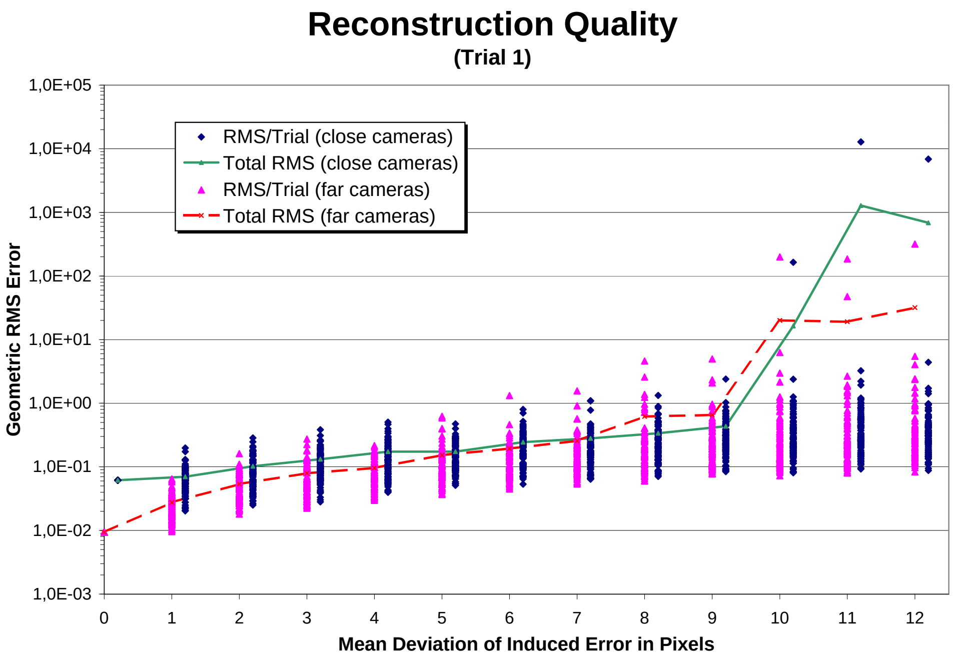


Figure 5.8: Comparison of reconstruction quality for first trial.

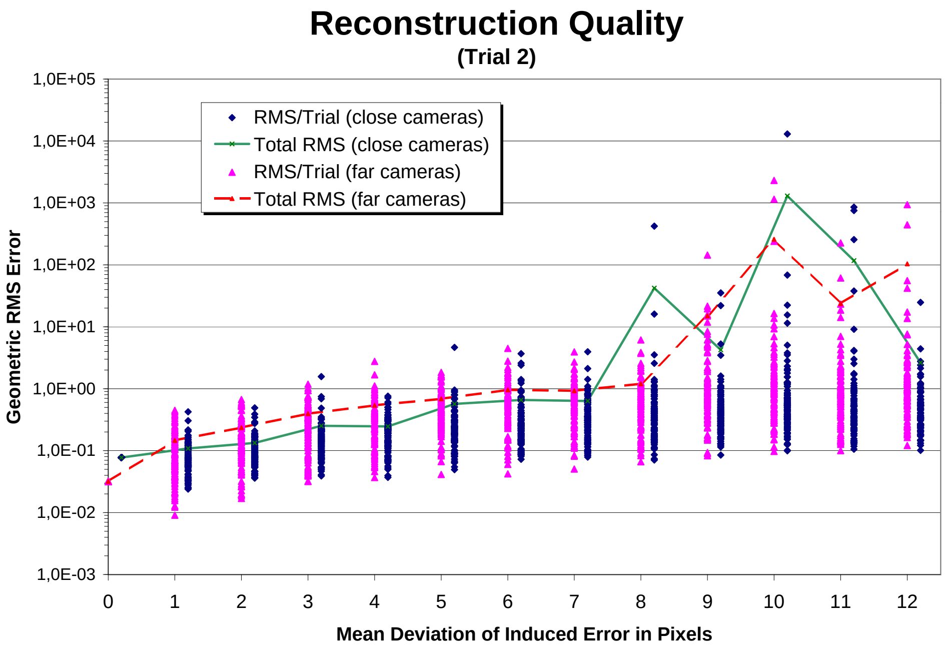


Figure 5.9: Comparison of reconstruction quality for second trial.

The first thing we can see from the graphs is that as the induced error increases over 6 pixels we start to get error configurations where the algorithm breaks down. This can be either due to the minimisation getting stuck in local minima or because the absolute minimum is at a wrong position. The latter is possible since the minimisation surface depends on M^{∞} and F.

Furthermore, it can be seen that the far cameras configuration is more immune to induced errors than the close cameras configuration. Also the non-orthogonal set of vanishing points fares worse than the orthogonal one. Curiously, in trial 2 the far cameras configuration is worse than the close cameras configuration.

In general it can be seen, though, that an error with a mean deviation of up to 5 pixels still gives acceptable reconstructions. It might seem odd, though, that if some error is introduced into the image points, the reconstruction can actually be better than with no noise at all. This is because even if no additional error is applied, there is still an error due to the digitisation in the cameras. Particular configurations of induced error can compensate for that by chance. However, the figures also show that the probability of the added error improving the reconstruction is about as high as making the reconstruction worse (relative to the total RMS). Nevertheless, this fact supplies us with an interesting idea: we might be able to improve our reconstructions from real data by *adding noise* to the image points. Since our calibration algorithm is very fast it seems feasible to employ maximum entropy methods. We will discuss this in future work.

Recall that we have calculated F with a simple method which does not enforce the rank 2 constraint on F. Nevertheless, the reconstruction quality is quite good, which seems to indicate that a highly accurate F is not very important for our algorithm. Of course, using a fully contraint F might improve the results.

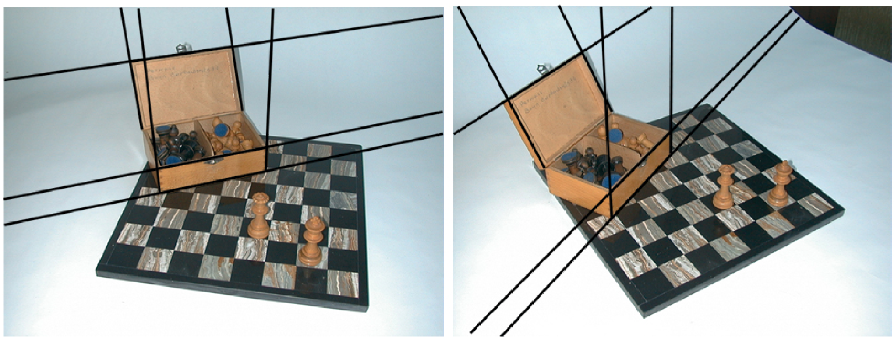


Figure 5.10: Initial images with parallel lines used for the calculation of the vanishing points and minimisation function indicated.

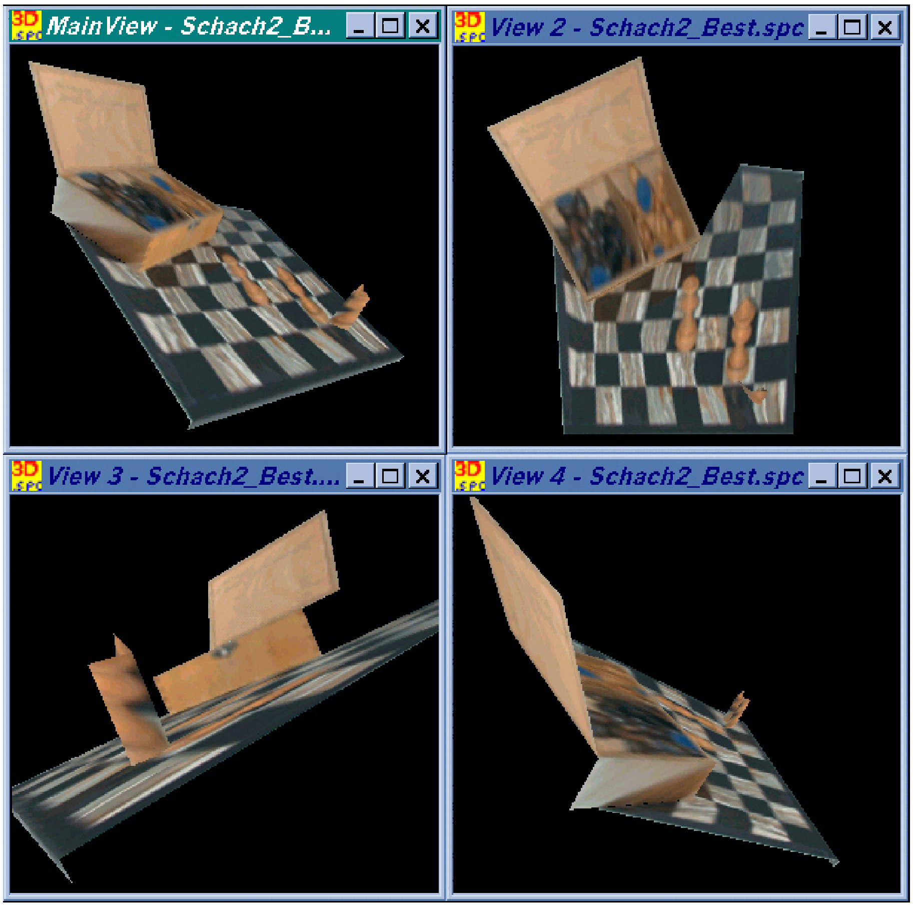

MainView - Schach2_B...
View 2 - Schach2_Best.spc
View 3 - Schach2_Best....
View 4 - Schach2_Best.spc

Figure 5.11: Reconstruction of the chessboard (Schachbrett).

5.7.2 Real Data

The real test for any reconstruction algorithm is the reconstruction of a real world scene. Figure 5.10 shows two views of a chessboard which we used for reconstruction[10]. The original images had a resolution of 1280 × 960 pixels. The lines indicate the parallel lines used to calculate the vanishing points. The two sets of parallel lines on the front of the chessbox were used in the minimisation routine. The fundamental matrix used was calculated from 13 point matches. The resultant reconstruction[11] can be seen in figure 5.11.

The different views of the reconstruction show that the chessbox was reconstructed quite well. However, the chessboard is not really square. Remember, though, that this is only an affine reconstruction drawn in an orthonormal frame. That is, we assume that the camera frame is orthonormal. Furthermore, we have only used two line pairs and one line triplet to find three vanishing points, of which only two relate to orthogonal directions in $\mathcal{E}^3$. The reconstruction might be improved by exploiting all the parallel lines available, of which there are many on a chessboard.

Also note that the front side of the chessboard is reconstructed very nicely, at a proper right angle to its top side. The chess figure, which can be seen best in the bottom left hand view of figure 5.11, is not reconstructed particularly well. This is because it is very difficult to find matching point sets for round objects.

5.8 Conclusions

We have presented here an algorithm for the affine reconstruction of 3D scenes from two static images. The information we need is firstly point matches over the two images, and secondly at least three sets of parallel lines. From this information alone we implicitly[12] find the internal calibration, rotation and translation of the second camera relative to the first one. This allows us to perform an affine reconstruction of the scene. Assuming that the three sets of parallel lines are mutually orthogonal we could also find the internal calibration of the first camera.

Our algorithm is clearly not automatic. This is because apart from the point

[10] These pictures were actually taken by C.Perwass' father, in a different country, with equipment unknown to the authors. They were then sent via email to the authors. That is, the only thing known about the pictures to the authors, are the pictures themselves.

[11] This and other reconstructions, as well as some more analysis of the reconstruction algorithm are demonstrated by the program **MVT**. See appendix A.

[12] Future work will look at how these entities can be found explicitly.

matches, combinations of vanishing points and parallel lines can be chosen freely. Also the information that certain lines in an image are actually parallel in the world, is a knowledge-based decision that humans are easily capable of, but not computers.

Advantages of our algorithm are that it is fast and that the reconstruction is robust for a particular calibration. On a PentiumII/233MHz under Windows 98 it took on average 160ms for a calibration (10000 trials). This time includes updating of dialog boxes. In an optimised program this time could probably be reduced to less than half. Robustness of the calibration depends mostly on the set of vanishing points used. The more similar the directions the vanishing points describe are, the less robust the calibration is.

We believe that apart from presenting an interesting affine reconstruction algorithm we have also shown that GA is a useful tool which allows us to gain geometric insight into a problem.

Chapter 6

Conclusions

I give you now Professor Twist,
A conscientious scientist.
Trustees exclaimed, "He never bungles!"
And sent him off to distant jungles.
Camped on a tropic riverside,
One day he missed his loving bride.
She had, the guide informed him later,
Been eaten by an alligator.
Professor Twist could not but smile.
"You mean," he said, "a crocodile."

"The Purist", Ogden Nash, 1938.

We have presented here not only some new results but have also shown how the GA approach to Computer Vision can simplify and unify the mathematical expressions and concepts. This was possible because GA gives us a "behind the scenes" look. That is, we do not just work with coordinates, as in the standard tensor method approach, but with the underlying geometry. The best example of this is probably that the complicated "p-relations" [32] follow from the geometric fact that the intersection point of a line with a plane, lies on that line (see equation (4.49), page 83).

Using GA to work in projective space is intuitive to us because we can interpret all algebraic objects as points, lines or planes and intersections thereof. However, points, lines and planes are just convenient interpretations of linear spaces, which is what GA is really all about. The power of GA is derived from the outer product and the extension of the scalar to the inner product. The outer product basically adds two linear spaces if they are linearly independent. The inner product, on the other hand, subtracts two linear spaces, if one is contained in the other. That is, apart from adding and subtracting linear dependences, as in standard vector algebra, we can also add and subtract linear spaces in GA.

A geometry is generally defined through a set of elements and an intersection operation which acts on them. In this respect, GA does not immediately seem to be a good candidate to describe a geometry. However, an intersection operation can be defined in terms of the inner and outer product, and is called the meet (see definition 3.9.1, page 59). Therefore, we have all the necessary prerequisites to describe projective, affine and Euclidean geometry with GA.

A particular feature of our description of projective geometry is that we use frames and reciprocal frames throughout. We showed that a reciprocal vector is the dual of a plane. Reciprocal vectors therefore enabled us to express intersections between planes very concisely (see equations (3.54) and (3.55), page 62).

This particular use of reciprocal vectors formed the basis of our unified treatment of multiple view tensors. In chapter 4 we showed that the fundamental matrix can be understood as the projection matrix of a line onto an image plane (see equation (4.20), page 75).

The affine 3D-reconstruction algorithm we presented in chapter 5 is a direct result of the additional geometric insight we gained on the collineation of the plane at infinity (Ψ^{∞}) by using GA. There we expressed Ψ^{∞} in terms of a camera matrix (see equation (5.29), page 115). Together with the collineation matrix or tensor this enabled us to obtain a reconstruction equation with just two free parameters. The result of this analysis was a fast and robust affine reconstruction algorithm.

In conclusion we can say that we have presented new results in geometric algebra, in projective geometry, in the analysis of multiple view tensors and in the field of (affine) 3D-reconstruction. We believe that this thesis has shown clearly the advantages gained when using GA, but apart from that has also provided new insights and tools for Computer Vision research.

6.1 Summary of Important Results

In the following we will summarise the most important or useful results.

6.1.1 Geometric Algebra

Definition 2.3.1 (p.20) *Let $\mu, \nu \in P_{\mathcal{V}^n}$, then the* ***outer product*** *of reduced basis blades $e_\mu, e_\nu \in \mathcal{G}^n$ is written $e_\mu {\wedge} e_\nu$ and defined as*

$$e_\mu {\wedge} e_\nu \equiv \begin{cases} e_\mu e_\nu & \textit{if } \mu \cap \nu = \emptyset \\ 0 & \textit{otherwise} \end{cases}$$

Definition 2.3.4 (p.23) *Let $\mu, \nu \in P_{\mathcal{V}^n}$, then the* ***inner product*** *of reduced basis blade $e_\mu, e_\nu \in \mathcal{G}^n$ is written $e_\mu {\cdot} e_\nu$ and defined as*

$$e_\mu {\cdot} e_\nu \equiv \begin{cases} e_\mu e_\nu & \textit{if } \mu \cap \nu \neq \emptyset \; \textbf{\textit{and}} \\ & \mu \cap \nu = \begin{cases} \mu & 0 < |\mu| \leq |\nu| \\ \nu & 0 < |\nu| < |\mu| \end{cases} \\ 0 & \textit{otherwise} \end{cases}$$

Theorem 2.3.1 (p.21) *Let $\mu \in P_{\mathcal{V}^n}$, $r \in N_{\mathcal{V}^n}$ and hence $e_\mu, e_r \in \mathcal{G}^n$. Then the outer product between e_μ and e_r can be written as*

$$e_\mu {\wedge} e_r = \begin{cases} e_\mu \overline{\times} e_r & \textit{if } |\mu| \textit{ is even or zero} \\ e_\mu \underline{\times} e_r & \textit{if } |\mu| \textit{ is odd} \end{cases}$$

Theorem 2.3.2 (p.23) *Let $\mu \in P_{\mathcal{V}^n}$, $r \in N_{\mathcal{V}^n}$ and hence $e_\mu, e_r \in \mathcal{G}^n$. Then the inner product between e_μ and e_r can be written as*

$$e_\mu {\cdot} e_r = \begin{cases} e_\mu \underline{\times} e_r & \textit{if } |\mu| \textit{ is even or zero} \\ e_\mu \overline{\times} e_r & \textit{if } |\mu| \textit{ is odd} \end{cases}$$

Theorem 2.4.2 (p.28) *The cyclic reordering property. Let $A, B, C \in \mathcal{G}^n$ be three general multivectors. Then*

$$\langle ABC \rangle = \langle CAB \rangle$$

Lemma 2.5.2 (p.34) *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{V}^n$ and $A_{\langle k \rangle} \in \mathcal{G}^n$. Furthermore, assume that $A_{\langle k \rangle} {\wedge} \boldsymbol{a} \neq 0$ and $A_{\langle k \rangle} {\cdot} \boldsymbol{b} \neq 0$. Then*

$$(A_{\langle k \rangle} {\wedge} \boldsymbol{a}) {\cdot} \boldsymbol{b} = A_{\langle k \rangle} (\boldsymbol{a} {\cdot} \boldsymbol{b}) - (A_{\langle k \rangle} {\cdot} \boldsymbol{b}) {\wedge} \boldsymbol{a}$$

Equations 2.64 (p.35)

$$\begin{aligned} A_{\langle k \rangle} \veebar (\boldsymbol{a} {\wedge} \boldsymbol{b}) &= (A_{\langle k \rangle} {\wedge} \boldsymbol{a}) {\cdot} \boldsymbol{b} - (A_{\langle k \rangle} {\wedge} \boldsymbol{b}) {\cdot} \boldsymbol{a} \\ A_{\langle k \rangle} \barwedge (\boldsymbol{a} {\wedge} \boldsymbol{b}) &= (A_{\langle k \rangle} {\cdot} \boldsymbol{a}) {\cdot} \boldsymbol{b} + (A_{\langle k \rangle} {\wedge} \boldsymbol{a}) {\wedge} \boldsymbol{b} \end{aligned}$$

Theorem 2.5.1 (p.36) *Let the $\{\boldsymbol{a}_i\}$ be a basis of $\mathcal{V}^n$, and let a blade $A_{\langle k \rangle} \in \mathcal{G}^n$ be defined as $A_{\langle k \rangle} = \boldsymbol{a}_1 {\wedge} \boldsymbol{a}_2 {\wedge} \ldots {\wedge} \boldsymbol{a}_k$. Also let $\boldsymbol{b} \in \mathcal{V}^n$. Then,*

$$A_{\langle k \rangle} {\cdot} \boldsymbol{b} = \sum_{i=0}^{k-1} (-1)^i (\boldsymbol{a}_{k-i} {\cdot} \boldsymbol{b}) \; [A_{\langle k \rangle} \backslash \boldsymbol{a}_{k-i}]$$

6.1.2 Projective Geometry

Embedding a vector from $\mathcal{E}^3$ in $\mathcal{P}^3$. **Equation (3.3)** (p.44)

$$\boldsymbol{x} \xrightarrow{\mathcal{P}^3} X = \alpha^i \, e_i + e_4 = \alpha^\mu \, e_\mu,$$

Projecting from $\mathcal{P}^3$ into $\mathcal{E}^3$. **Equation (3.4)** (p.44)

$$X \xrightarrow{\mathcal{E}^3} \boldsymbol{x} = \frac{X {\cdot} e^i}{X {\cdot} e^4} \, e_i$$

The dual bracket. **Equation (3.14)** (p.47)

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

Reciprocal A-frame. **Equation (3.32)** (p.56)

$$A^{\mu_1} = \rho_a^{-1} \; [\![A_{\mu_2} A_{\mu_3} A_{\mu_4}]\!]$$

Reciprocal A-line frame. **Equation (3.36)** (p.57)

$$\begin{aligned} L_{\nu}^{a} = \rho_{a}^{-1} \Big\{ & [\![A_1 A_4]\!], [\![A_2 A_4]\!], [\![A_3 A_4]\!], \\ & [\![A_3 A_1]\!], [\![A_2 A_3]\!], [\![A_1 A_2]\!] \Big\} \end{aligned}$$

The meet operation. **Equation (3.47)** (p.59)

$$A \vee B = [\![[\![A]\!][\![B]\!]]\!] = [\![A]\!]\cdot B = A^{*}\cdot B$$

Equation (3.48) (p.60)

$$[\![A]\!] \vee [\![B]\!] = [\![AB]\!]$$

Equation (3.50) (p.60)

$$X \xrightarrow{A} X' = (X\wedge A_4) \vee (A_1\wedge A_2\wedge A_3) \simeq (\, X\cdot A^{i}\,)A_{i}$$

Equation (3.52) (p.61)

$$L \xrightarrow{A} L' = (L\wedge A_4) \vee (A_1\wedge A_2\wedge A_3) \simeq (\, L\cdot L_{i}^{a}\,)L_{a}^{i}$$

Camera matrix. **Equation (3.58)** (p.63)

$$K_{\mu}^{j\,\mathsf{b}} \equiv A_{\mu}\cdot B^{j}$$

6.1.3 Multiple View Tensors

Fundamental Matrix	Trifocal Tensor
$F_{i_1 j_1} = \llbracket A^{i_2} A^{i_3} B^{j_2} B^{j_3} \rrbracket$	$T_{i_1}{}^{jk} = \llbracket A^{i_2} A^{i_3} B^{j} C^{k} \rrbracket$
$F_{i j_1} = \varepsilon^{j_3}_{ba} K^{j_2}_{i}{}^{\mathsf{b}} - \varepsilon^{j}_{ba} K^{j_3}_{i}{}^{\mathsf{b}}$	$T_i{}^{jk} = \varepsilon^{j}_{ba} K^{k}_{i}{}^{\mathsf{c}} - \varepsilon^{k}_{ca} K^{j}_{i}{}^{\mathsf{b}}$
$F_{ij} = L^a_i \cdot \underbrace{(B_j \wedge B_4)}_{\text{line}}$	$T_i{}^{jk} = L^a_i \cdot \underbrace{\llbracket B^j C^k \rrbracket}_{\text{line}}$
$\det F = 0$	$\det(T_i{}^{jk})_{jk} = 0$ for each i
7 DOF	18 DOF
rank 2	rank 4

Quadfocal Tensor
$Q^{ijkl} = \llbracket A^i B^j C^k D^l \rrbracket$
$\begin{aligned} Q^{i_1 jkl} = \quad & \varepsilon^{j}_{ba} \Big[K^{k}_{i_3}{}^{\mathsf{c}} K^{l}_{i_2}{}^{\mathsf{d}} - K^{l}_{i_3}{}^{\mathsf{d}} K^{k}_{i_2}{}^{\mathsf{c}} \Big] \\ - \quad & \varepsilon^{k}_{ca} \Big[K^{j}_{i_3}{}^{\mathsf{b}} K^{l}_{i_2}{}^{\mathsf{d}} - K^{l}_{i_3}{}^{\mathsf{d}} K^{j}_{i_2}{}^{\mathsf{b}} \Big] \\ + \quad & \varepsilon^{l}_{da} \Big[K^{j}_{i_3}{}^{\mathsf{b}} K^{k}_{i_2}{}^{\mathsf{c}} - K^{k}_{i_3}{}^{\mathsf{c}} K^{j}_{i_2}{}^{\mathsf{b}} \Big] \end{aligned}$
$Q^{ijkl} = A^i \cdot \underbrace{\llbracket B^j C^k D^l \rrbracket}_{\text{point}}$
$\det(Q^{ijkl})_{xy} = 0$ where x and y are any pair of $\{ijkl\}$
29 DOF
rank 9

6.1.4 3D-Reconstruction

Collineation Tensor. **Equation (5.21)** (p.113)

$$\begin{aligned} M^{k}_{ij} \equiv \Big[\quad & \Big(F(1,2)\,\bar{\lambda}^{1}_{a\,i}\,\bar{\beta}^{k}_{1} - F(2,1)\,\bar{\lambda}^{2}_{a\,i}\,\bar{\beta}^{k}_{2}\Big)\,f^{b}_{j3} \\ - \quad & \Big(F(1,2)\,\bar{\lambda}^{1}_{a\,i}\,f^{b}_{j1} - F(2,1)\,\bar{\lambda}^{2}_{a\,i}\,f^{b}_{j2}\Big)\,\bar{\beta}^{k}_{3}\Big] \end{aligned}$$

with

$$\bar{\lambda}^{j_1}_{a\,k_1} \equiv (\bar{\alpha}^{k_2}_{j_2}\bar{\alpha}^{k_3}_{j_3} - \bar{\alpha}^{k_3}_{j_2}\bar{\alpha}^{k_2}_{j_3}).$$

$$F(r,s) \equiv \bar{\alpha}^{i}_{r}\bar{\beta}^{j}_{s}\,F_{ij}$$

$$f^{b}_{ir} \equiv \bar{\beta}^{j}_{r}\,F_{ij}$$

P_{∞}-collineation. **Equation (5.29)** (p.115)

$$\Psi^{j\ \infty}_{\ i} \equiv [K^{j\ \mathsf{b}}_{\ 1}, K^{j\ \mathsf{b}}_{\ 2}, K^{j\ \mathsf{b}}_{\ 3} - \varepsilon^{j}_{ba}]$$

Reconstruction equation. **Equation (5.36)** (p.116)

$$\bar{\alpha}^{4} = \bar{\alpha}^{i}\bar{K}^{3\ \mathsf{b}}_{i}\,\zeta^{j}_{1} - \zeta^{j}_{2}\ ; \quad j \in \{1,2\}$$

Acknowledgements. I would like to thank Joan and Anthony Lasenby for their help with my research in general, Alan MacDonald for helpful feedback on chapter 2, Adam for in–depth discussions about projective geometry in general and chapter 3 in particular, Klaus and Ole for entertaining conversations about my work and anything else as well, Uli for telling me everything about computer security that I always wanted to know but never dared to ask, Ignas for generally having a good time at B.C. and never letting me win at Squash, and, of course, my parents for their constant support.

Appendix A

The MVT Program

Enclosed with this thesis is a CD which contains the **MVT** program. The program runs under Windows 95/98/NT4/5. It demonstrates some aspects of the reconstruction algorithm. In particular it shows the relation between the minimisation surface and the reconstruction. However, some examples of how the reconstruction quality changes when using different sets of vanishing points and parallel lines, are also given.

The program allows the user to rotate and move all objects freely in 3D-space. Furthermore, the objects can be viewed in a real 3D-mode with red/green or red/blue 3D-glasses.

Before starting the program please make sure that your graphics card and monitor can display 16Bit colour and are set appropriately. Also, help and the demonstrations are only available if the Internet Explorer is installed[1]. When the program is started it should come up with a help screen which gives detailed information about everything else you need to know.

Should you not be in posession of the enclosed CD, you can also download the program from my webpage. At the time of writing this is:

`http://www.perwass.de/`

If you have any questions or comments please contact me under:

`christian@perwass.de`

[1] Apologies to all those who have uninstalled Internet Explorer out of protest.

Bibliography

[1] Albrecht Beutelspacher and Ute Rosenbaum. *Projective Geometry*. Cambridge University Press, 1998. ISBN 0-521-48277-1.

[2] David A. Brannan, Matthew F. Esplen, and Jeremy J. Gray. *Geometry*. Cambridge University Press, 1999. ISBN 0-521-59193-7.

[3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4:127–140, 1990.

[4] Roberto Cipolla and Edmond Boyer. 3d model acquisition from uncalibrated images. In *Proc. IAPR Workshop on Machine Vision Applications, Chiba, Japan*, pages 559–568, 1998. To be published.

[5] William K. Clifford. Applications of grassmann's extensive algebra. In R. Tucker, editor, *Mathematical Papers*, pages 266–276. Macmillian, London, 1882.

[6] William K. Clifford. *Mathematical Papers*. Macmillian, London, 1882.

[7] William K. Clifford. On the classification of geometric algebras. In R. Tucker, editor, *Mathematical Papers*, pages 397–401. Macmillian, London, 1882.

[8] R. W. R. Darling. *Differential Forms and Connections*. Cambridge University Press, 1994.

[9] Richard A. Dean. *Classical Abstract Algebra*. Harper & Row, 1990.

[10] Frédéric Devernay and Olivier Faugeras. From projective to euclidean reconstruction. Technical Report 2725, INRIA, Sophia Antipolis, 1995.

[11] Leo Dorst. Honing geometric algebra for its use in the computer sciences. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*. Springer-Verlag, 2000 to be published.

[12] Reyes Enciso and Thierry Vieville. Self-calibration from four views with possibly varying intrinsic parameters. *Image and Vision Computing*, 15:293–305, 1997.

[13] Olivier Faugeras. Stratification of Three Dimensional Vision: Projective, Affine and Metric Representations. *Journal of the Optical Society of America - A*, 12(3), 1995.

[14] 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.

[15] Olivier Faugeras and Bernard Mourrain. On the Geometry and Algebra of the Point and Line Correspondences between N Images. In *Proceedings ICCV'95*, pages 951–956, 1995.

[16] Olivier Faugeras and Theodore Papadopoulo. Grassmann–Cayley algebra for modelling systems of cameras and the algebraic equations of the manifold of trifocal tensors. Technical Report 3225, INRIA, Sophia Antipolis, 1997.

[17] Olivier Faugeras and Theodore Papadopoulo. A nonlinear method for estimating the projective geometry of three views. Technical Report 3221, INRIA, Sophia Antipolis, 1997.

[18] 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.

[19] Olivier D. Faugeras and Steve Maybank. Motion from point matches: Multiplicity of solutions. *International Journal of Computer Vision*, 4:225–246, 1990.

[20] Theodore Frankel. *The Geometry of Physics*. Cambridge University Press, 1997.

[21] John E. Gilbert and Margaret A. M. Murray. *Clifford algebras and Dirac operators in harmonic analysis*. Cambridge University Press, 1991.

[22] S. F. Gull, A. N. Lasenby, and C. J. L. Doran. Imaginary numbers are not real – the geometric algebra of space time. *Found. Phys.*, 23(9):1175, 1993.

[23] Alexander J. Hahn. *Quadratic Algebras, Clifford Algebras, and Arithmetic Witt Groups*. Springer–Verlag, 1994.

[24] Richard I. Hartley. In defence of the 8-point algorithm. *IEEE Int. Conf. Computer Vision*, pages 1064–1070, 1995.

[25] Richard I. Hartley. Triangulation. In R. Sara and V. Hlavac, editors, *Computer Analysis of Images and Patterns*, number 970 in Lecture Notes in Computer Science, pages 190–197. Springer, 1995.

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

[27] Richard I. Hartley. Computation of the quadrifocal tensor. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV'98*, number 1406 in Lecture Notes in Computer Science. Springer, 1998.

[28] Richard I. Hartley. Minimizing algebraic error. In *Phil. Trans. R. Soc. Lond. A*, volume 356, pages 1175–1192, 1998.

[29] D. Hestenes. *New Foundations for Classical Mechanics.* Dordrecht, 1986.

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

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

[32] Anders Heyden. A Common Framework for Multiple View Tensors. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV'98*, number 1406 in Lecture Notes in Computer Science, pages 3–19. Springer, 1998.

[33] Anders Heyden and Kalle Aström. Simplifications of multilinear forms for sequences of images. *Image Vision and Computing*, 15:749–757, 1997.

[34] Anders Heyden and Gunnar Sparr. Reconstruction from calibrated cameras - a new proof of the kruppa-demazure theorem. *Journal of Mathematical Imaging and Vision*, 10:1–20, 1999.

[35] Wilhelm Klingenberg. *Lineare Algebra und Geometrie.* Springer–Verlag, 1990.

[36] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitz.–Ber. Akad. Wiss., Wien, math. naturw. Kl. Abt. IIa.*, 122:1939–1948, 1913.

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

[38] Joan Lasenby, W. J. Fitzgerald, A.N. Lasenby, and C.J.L. Doran. New geometric methods for computer vision: An application to structure and motion estimation. *International Journal of Computer Vision*, 3(26):191–213, 1998.

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

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

[41] Pertti Lounesto. *Clifford Algebras and Spinors.* Cambridge University Press, 1997.

[42] Quang-Tuan Luong, Rachid Deriche, Olivier Faugeras, and Theo Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical Report 1894, INRIA, Sophia Antipolis, 1993.

[43] Quang-Tuan Luong and Olivier Faugeras. Self-calibration of a stereo rig from unknown camera motions and point correspondences. Technical Report 2014, INRIA, Sophia Antipolis, 1993.

[44] Alan MacDonald. Elementary Construction of the Geometric Algebra. In *Proceedings 5th International Conference on Clifford Algebras and their Applications in Mathematical Physics*, 1999. To be published.

[45] David MacKay. MacOpt - a nippy wee optimizer. http://wol.ra.phy.cam.ac.uk/mackay/c/macopt.html. Retrieved on 26 October 1999.

[46] Nassir Navab and Olivier Faugeras. The critical sets of lines for camera displacement estimation: a mixed euclidean-projective and constructive approach. Technical Report 2305, INRIA, Sophia Antipolis, 1994.

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

[48] C.B.U. Perwass and J. Lasenby. A Geometric Analysis of the Trifocal Tensor. In R. Kakarala R. Klette, G. Gimel'farb, editor, *Image and Vision Computing New Zealand, IVCNZ'98, Proceedings*, pages 157–162. The University of Auckland, 1998.

[49] 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.

[50] C.B.U. Perwass and J. Lasenby. 3D-Reconstruction from Vanishing Points. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*. Springer-Verlag, 2000 to be published.

[51] C.B.U. Perwass and J. Lasenby. A Unified Description of Multiple View Geometry. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*. Springer-Verlag, 2000 to be published.

[52] Ian R. Porteous. *Clifford Algebras and the Classical Groups.* Cambridge University Press, 1995.

[53] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in Pascal.* Cambridge University Press, 1990. ISBN 0-521-37516-9.

[54] Marcel Riesz. *Clifford Numbers and Spinors.* Kluwer Academic Publishers, 1993.

[55] B.A. Rosenfeld. *A history of non-Euclidean geometry.* Springer–Verlag, 1976.

[56] Charlie Rothwell, Gabriella Csurka, and Olivier Faugeras. A Comparison of Projective Reconstruction Methods for Pairs of Views. Technical Report 2538, INRIA, Sophia Antipolis, 1995.

[57] 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.

[58] J.G. Semple and G.T.Kneebone. *Algebraic Projective Geometry.* Oxford University Press, 1952. ISBN 0-19-8503636.

[59] 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, pages 190–206. Springer, 1997.

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

[61] Amnon Shashua and Shai Avidan. The rank 4 constraint in multiple (≥ 3) view geometry. In R. Cipolla and B. Buxton, editors, *Computer Vision – ECCV'96*, number 1065 in Lecture Notes in Computer Science, pages 196–206. Springer, 1996.

[62] Amnon Shashua and Steve J. Maybank. Degenerate n point configuration of three views: Do critical surfaces exist? TR 96–19, Hebrew University, Computer Science, 1996.

[63] Ilan Shimshoni, Ronen Basri, and Ehud Rivlin. A geometric interpretation of weak-perspective motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):252–257, 1999.

[64] Frank Sommen. An algebra of abstract vector variables. *Porugaliae Math.*, 54:287–310, 1997.

[65] Frank Sommen. The problem of defining abstract bivectors. *Result. Math.*, 31:148–160, 1997.

[66] Bill Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, 1997.

[67] Bill Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15:617–625, 1997.

[68] Bill Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, 1998.

[69] Neil L. White. Grassmann–Cayley algebra and robotics. *J. Intell. Robot. Syst.*, 11:91–107, 1994.

[70] Neil L. White. A tutorial on Grassmann–Cayley algebra. In Neil L. White, editor, *Invariant methods in discrete and computational geometry*, pages 93–106. Kluwer, Dordrecht, 1995.

[71] Cyril Zeller and Olivier Faugeras. Camera self-calibration from video sequences: the kruppa equations revisited. Technical Report 2793, INRIA, Sophia Antipolis, 1996.

Wenn er [der Mensch] sein Einssein mit den anderen fühlt, sieht er vielleicht zum ersten Mal, daß es eine Illusion ist, sein isoliertes, individuelles Ich für etwas zu halten, das er festhalten, kultivieren und bewahren soll; er wird empfinden, wie nutzlos es ist, die Antwort auf das Leben darin zu suchen, sich zu haben, anstatt er selbst zu sein und zu werden.

“Haben oder Sein”, Erich Fromm.