PERGAMON

Pattern Recognition 35 (2002) 169–186

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A geometric approach for the analysis and computation of the intrinsic camera parameters

Eduardo Bayro-Corrochano[a,*], Bodo Rosenhahn[b]

[a]Centro de Investigación en Matemáticas, AC, Apartado Postal 402, 36000 - Guanajuato, Gto, Mexico
[b]Computer Science Institute, Christian Albrechts University, Preusserstrasse 1-9, 24105 Kiel, Germany

Received 17 March 2000; accepted 1 November 2000

Abstract

The authors of this paper adopted the projected characteristics of the absolute conic in terms of the Pascal's theorem to propose an entirely new camera calibration method based on purely geometric thoughts. The use of this theorem in the geometric algebra framework allows us to compute a projective invariant using the conics of only two images which expressed using brackets helps us to set enough equations to solve the calibration problem. The method requires restricted controlled camera movements. Our method is less sensitive to noise as the Kruppa's-equation-based methods. Experiments with simulated and real images confirm that the performance of the algorithm is reliable.

Keywords: Computer vision; Projective geometry; Clifford algebra; Geometric algebra; Calibration; Kruppa's equations; Essential and fundamental matrices

1. Introduction

The computation of the intrinsic camera parameters is one of the most important issues in computer vision. The traditional way to compute the intrinsic parameters is using a known calibration object. One of the most important method is based on the absolute conic and it requires as input only information about the point correspondences [1]. As extension a recent approach utilizes the absolute quadric [2,3]. Other important group of self-calibration methods reduces the complexity if the camera motion is known in advance, for example, translational [4], rotational about known angles [5,6], actively using the vanishing point [7] or for the case of planar camera motion image triples are used [8].

In this paper, we reuse the idea of the absolute conic in the context of the Pascal's theorem and surprisingly we get equations different to the Kruppa's ones [1,9]. Although the equations are different, they rely on the same principle of the invariance of the mapped absolute conic. The consequences are that we can decouple equations so that we require only two image views as opposite to the Kruppa's methods that require at least three image views [9]. However, as input information the method requires the translational motion direction of the camera in addition to the point correspondences. A rotation about only one axis through a known angle is also needed, that means either $\mathbf{R}_x$ or $\mathbf{R}_y$ or $\mathbf{R}_z$. Note that we do not need the entire essential matrix. The paper will show that although the algorithm requires the camera translation in advance and a rotation about one axis it has the following clear advantages: it is derived on purely geometric observations, it does not suffer a local minima in the computation of the intrinsic parameters and it does not require any initialization at all. We do hope that this original method derived from a purely geometric thoughts throws new lights to the problem of camera

* Correspondence address. Unidad Guadalajara, CINVESTAV, Computer Science Department, Apartado Postal 31-438, Apartado la Luna, 44550 Guadalajara, Jal, Mexico. Tel.: + 52-3-684-1580; fax: + 52-3-684-1708.
E-mail address: edb@gdl.cinvestav.mx (E. Bayro-Corrochano).
URLs: http://www.gdl.cinvestav.mx/~edb

0031-3203/01/$20.00
PII: S0031-3203(00)00182-5

calibration and will help in the understanding and improvement of such calibration methods which use controlled camera movements.

The paper is organized as follows: Section 2 gives a brief introduction to geometric algebra and Section 3 presents the basics of computer vision in the geometric algebra framework. Section 4 explains the conics and the Pascal theorem. Section 5 reformulates the well-known Kruppa's equations for computer vision in terms of algebra of incidence. Section 6 presents a new method for computing the intrinsic camera parameters based on Pascal's theorem. Section 7 is devoted to the experimental analysis and Section 8 to the conclusion part.

2. Geometric algebra: an outline

Geometric algebra (GA) is a coordinate-free approach to geometry based on the algebras of Grassmann [10] and Clifford [11]. The algebra is defined on a space whose elements are called *multivectors*; a multivector is a linear combination of objects of different type, e.g., scalars and vectors. It has an associative and fully invertible product called the *geometric* or *Clifford* product. The existence of such a product and the calculus associated with the geometric algebra give the system tremendous power. The geometric approach to Clifford algebra adopted in this paper was pioneered in the 1960s by David Hestenes who has, since then, worked on developing his version of Clifford algebra — which will be referred to as *geometric algebra* — into a unifying language for mathematics and physics [12]. Geometric algebra provides a very natural language for projective geometry and has all the necessary equipment to express very elegantly the ideas of incidence algebra involving the duality principle and the meet and join operations. For a more complete treatment of the projective geometry see Ref. [13] and for a brief summary see Ref. [14]. Some preliminary applications of geometric algebra in the field of computer vision have already been given [14–18].

We will begin in the following subsections with basic definitions of geometric algebra necessary for projective geometry. In the whole paper, we will denote with lower-case scalars, upper-case matrices, slant-bold lower-case vectors in 3-D and slant-bold upper-case for vectors in 4-D.

2.1. The geometric product and multivectors

Let $\mathcal{G}_n$ denote the geometric algebra of n-dimensions — this is a graded linear space. As well as vector addition and scalar multiplication we have a non-commutative product which is associative and distributive over addition — this is the geometric or Clifford product. A further distinguishing feature of the algebra is that any vector squares to give a scalar. The geometric product of

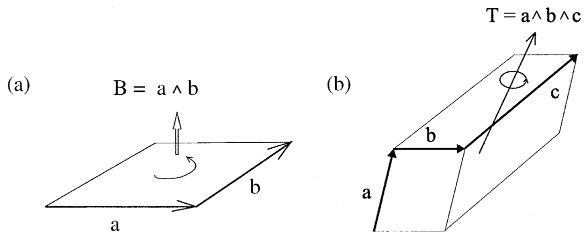


Fig. 1. (a) The directed area, or bivector, $\mathbf{a}\wedge\mathbf{b}$. (b) The oriented volume, or trivector, $\mathbf{a}\wedge\mathbf{b}\wedge\mathbf{c}$.

two vectors $\mathbf{a}$ and $\mathbf{b}$ is written $\mathbf{ab}$ and can be expressed as a sum of its symmetric and antisymmetric parts

$$\mathbf{ab} = \mathbf{a}\cdot\mathbf{b} + \mathbf{a}\wedge\mathbf{b}, \tag{1}$$

where the inner product $\mathbf{a}\cdot\mathbf{b}$ and the outer product $\mathbf{a}\wedge\mathbf{b}$ are defined by

$$\mathbf{a}\cdot\mathbf{b} = \tfrac{1}{2}(\mathbf{ab}+\mathbf{ba}), \quad \mathbf{a}\wedge\mathbf{b} = \tfrac{1}{2}(\mathbf{ab}-\mathbf{ba}). \tag{2}$$

The inner product of two vectors is the standard *scalar* or *dot* product and produces a scalar. The outer or wedge product of two vectors is a new quantity we call a *bivector*. We think of a bivector as a directed area in the plane containing $\mathbf{a}$ and $\mathbf{b}$, formed by sweeping $\mathbf{a}$ along $\mathbf{b}$ (see Fig. 1).

Thus, $\mathbf{b}\wedge\mathbf{a}$ will have the opposite orientation making the wedge product anti-commutative as given in Eq. (2). The outer product is immediately generalizable to higher dimensions — for example, $(\mathbf{a}\wedge\mathbf{b})\wedge\mathbf{c}$, a *trivector*, is interpreted as the oriented volume formed by sweeping the area $\mathbf{a}\wedge\mathbf{b}$ along vector $\mathbf{c}$ (see Fig. 1). The outer product of k vectors is a k-vector or k-blade, and such a quantity is said to have *grade k*. A *multivector* is made up of a linear combination of objects of different grade, i.e., scalar plus bivector, etc. GA provides a means of manipulating multivectors which allows us to keep track of different grade objects simultaneously — much as one does with complex number operations. For a general multivector X, the notation $\langle X\rangle$ will mean *take the scalar part of X*. The highest grade element in a space is called the *pseudoscalar*. The unit pseudoscalar is denoted by I and is crucial when discussing duality.

We now end this introductory section by giving a very brief review of the geometric algebra approach to linear algebra. A more detailed review is found in Ref. [12].

Consider a linear function f which maps vectors to vectors in the same space. We can extend f to act linearly on multivectors via the *outermorphism*, $\underline{f}$, defining the action of $\underline{f}$ on blades by

$$\underline{f}(\mathbf{a}_1\wedge\mathbf{a}_2\wedge\cdots\wedge\mathbf{a}_r) = f(\mathbf{a}_1)\wedge f(\mathbf{a}_2)\wedge\cdots\wedge f(\mathbf{a}_r). \tag{3}$$

We use the term outermorphism because $\underline{f}$ preserves the grade of any r-vector it acts on. We therefore know that the pseudoscalar of the space must be mapped onto some multiple of itself. The scale factor in this mapping is the *determinant* of $\underline{f}$:

$$\underline{f}(I) = \det(\underline{f})I. \quad (4)$$

This is much simpler than many definitions of the determinant enabling one to establish most properties of determinants with little effort.

3. The geometric algebra for computer vision

This section aims to give the formulation of the projective geometry and algebra of incidence required for the treatment of problems in computer vision in a framework with a strong geometric representation character and amenable algebraic manipulation facilities. Next, we will model the camera and the visual space in the geometric algebra.

3.1. The 3-D geometric algebra of the camera

It is important for real applications to regard the signature of the modelled space to facilitate the computations. In the case of the modelling of the image plane using homogeneous coordinates, we adopt $\mathcal{G}_{3,0,0}$ of the ordinary space, E^3, which has the standard Euclidean signature. The basis for the 3-D space has $2^3 = 8$ elements given by

$$\underbrace{1}_{scalar}, \underbrace{\{\sigma_1,\sigma_2,\sigma_3\}}_{vectors}, \underbrace{\{\sigma_1\sigma_2,\sigma_2\sigma_3,\sigma_3\sigma_1\}}_{bivectors}, \underbrace{\{\sigma_1\sigma_2\sigma_3\} \equiv I}_{trivector}. \quad (5)$$

The highest grade element is a trivector called the *pseudoscalar*. It can easily be verified that the pseudoscalar $\sigma_1\sigma_2\sigma_3$ squares to -1 and commutes with all multivectors in the 3-D space. We therefore give it the symbol I. In a space of 3-D, we can construct a trivector $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$, but no 4-vectors exists since there is no possibility of sweeping the volume element $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ over a fourth dimension. Multiplication of the three basis vectors $\{\sigma_i\}$ by I results in the three basis bivectors; $\sigma_2\sigma_3 = I\sigma_1$, $\sigma_3\sigma_1 = I\sigma_2$, $\sigma_1\sigma_2 = I\sigma_3$. These simple bivectors rotate vectors in their own plane by 90°, e.g., $(\sigma_1\sigma_2)\sigma_2 = \sigma_1$, $(\sigma_2\sigma_3)\sigma_2 = -\sigma_3$, etc. Identifying the $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ of the quaternion algebra with $\sigma_2\sigma_3$, $-\sigma_3\sigma_1$ and $\sigma_1\sigma_2$, we see that the famous Hamilton relations are recovered $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. The quaternion algebra is therefore seen to be an even subalgebra of the 3-space or $\mathcal{G}^+_{3,0,0}$.

3.2. The 4-D geometric algebra of the visual space

Since we selected $\mathcal{G}_{3,0,0}$ for modelling the image plane we are forced to adopt the same signature for the 4-D visual space. This can be achieved using the 4-D geometric algebra $\mathcal{G}_{1,3,0}$ which we associate with the projective space P^3. This is spanned with the following basis:

$$\underbrace{1}_{scalar}, \underbrace{\gamma_k}_{4\ vectors}, \underbrace{\gamma_2\gamma_3,\gamma_3\gamma_1,\gamma_1\gamma_2,\gamma_4\gamma_1,\gamma_4\gamma_2,\gamma_4\gamma_3}_{6\ bivectors}, \underbrace{I\gamma_k}_{4\ pseudovectors}, \underbrace{I}_{pseudoscalar}, \quad (6)$$

where $\gamma_4^2 = +1$, $\gamma_k^2 = -1$ for $k = 1,2,3$. The pseudoscalar is $I = \gamma_1\gamma_2\gamma_3\gamma_4$ with

$$I^2 = (\gamma_1\gamma_2\gamma_3\gamma_4)(\gamma_1\gamma_2\gamma_3\gamma_4) = -(\gamma_3\gamma_4)(\gamma_3\gamma_4) = -1. \quad (7)$$

The fourth basis vector γ_4 can be seen also as selected direction or *projective split* [14] in 4-D. The basis element γ_4 helps to associate multivectors of the 4-D space with multivectors of the 3-D space. The role and use of the projective split for a variety of problems involving the algebra of incidence can be found in Ref. [14].

3.3. Algebra of incidence

Here we will outline the approach pioneered by Hestenes for using geometric algebra to discuss the algebra of incidence. The basic projective geometry operations of meet and join will be shown to be easily expressible in terms of standard operations within the geometric algebra. For a more extended discussion we refer the reader to Ref. [13].

We have seen that in Euclidean spaces of 3-D the unit pseudoscalar squares to -1. In $\mathcal{G}_{1,3,0}$, it is easy to see that this is also the case. If γ_i, $i = 1,2,3,4$, are our basis vectors in the 4-D space, and $\gamma_j^2 = -1$ for $j = 1,2,3$, and $\gamma_4^2 = +1$, then

$$I^2 = (\gamma_1\gamma_2\gamma_3\gamma_4)(\gamma_1\gamma_2\gamma_3\gamma_4) = (\gamma_2\gamma_3\gamma_4)(\gamma_2\gamma_3\gamma_4)$$
$$= -(\gamma_3\gamma_4)(\gamma_3\gamma_4) = -1. \quad (8)$$

The sign of I^2 depends on the signature of the space. In a given space, any pseudoscalar P can be written as $P = \alpha I$, where α is a scalar. If I^{-1} is the inverse of I, so that $II^{-1} = 1$, then

$$PI^{-1} = \alpha II^{-1} = \alpha \equiv [P], \quad (9)$$

where we have defined the *bracket* of the pseudoscalar P, $[P]$, as its magnitude, arrived at by multiplication on the right by I^{-1}. The sign of the bracket does not depend on the signature of the space and as such it has been a useful quantity for the non-metrical applications of projective geometry.

To introduce the concepts of duality which are so important in projective geometry, we define the dual $\mathbf{A}^*$ of an r-vector $\mathbf{A}$ as

$$\mathbf{A}^* = \mathbf{A}I^{-1}. \quad (10)$$

We use the notation $\mathbf{A}^*$ to relate these ideas of duality to the notion of a *Hodge dual* in differential geometry. Note that in general I^{-1} may not commute with $\mathbf{A}$. From the definition of the unit pseudoscalar, we see that the dual of an r-vector is an $(n-r)$-vector (e.g., duality of lines $(r=1)$ $(n-r=3-1=2$ in the 3-D space). In an n-dimensional space, if $\mathbf{A}$ is an r-vector and $\mathbf{B}$ is an s-vector then using the fact that $\mathbf{B}I^{-1}=\mathbf{B}\cdot I^{-1}$ (since $\mathbf{B}I^{-1}$ must be of grade $(n-s)$)) and the identity

$$\mathbf{A}_r\cdot(\mathbf{B}_s\cdot\mathbf{C}_t)=(\mathbf{A}_r\wedge\mathbf{B}_s)\cdot\mathbf{C}_t \quad \text{for } r+s\leqslant t, \tag{11}$$

we can write

$$\mathbf{A}\cdot(\mathbf{B}I^{-1})=\mathbf{A}\cdot(\mathbf{B}\cdot I^{-1})=(\mathbf{A}\wedge\mathbf{B})\cdot I^{-1}=(\mathbf{A}\wedge\mathbf{B})I^{-1}. \tag{12}$$

Using the definition of the dual we therefore have

$$\mathbf{A}\cdot\mathbf{B}^*=(\mathbf{A}\wedge\mathbf{B})^*. \tag{13}$$

Eq. (13) illustrates the duality of the inner and outer products. If $r+s=n$, then $A\wedge B$ is the highest grade part of AB, i.e., the pseudoscalar part, and it then follows that

$$[\mathbf{A}\wedge\mathbf{B}]=(\mathbf{A}\wedge\mathbf{B})I^{-1}=\mathbf{A}\cdot\mathbf{B}^*. \tag{14}$$

In this case, we can express the bracket in terms of duals and as such, relate the inner and outer products to non-metrical quantities. It is via this route that the inner product, which is normally associated with a metric, is used in a non-metrical theory such as projective geometry. We note at this point that since we have reduced duality to a simple multiplication by an element of the algebra, there is no need to introduce a special operator or any concept of a different space.

In an n-dimensional geometric algebra one can define the *join* $\mathbf{J}=\mathbf{A}\bigwedge\mathbf{B}$ of an r-vector $\mathbf{A}$ and an s-vector $\mathbf{B}$ by

$$\mathbf{J}=\mathbf{A}\bigwedge\mathbf{B}. \tag{15}$$

If $\mathbf{A}$ and $\mathbf{B}$ have a common subspace, the join is not given simply by the wedge but also by the subspace that they span. In what follows, we will use $\bigwedge$ for the join only when the blades $\mathbf{A}$ and $\mathbf{B}$ have a common subspace, otherwise we will use the ordinary exterior product, $\wedge$. $\mathbf{J}$ can be interpreted as a *common dividend of lowest grade* and is defined up to a scale factor. It is easy to see that if $(r+s)\geqslant n$ then J will be the pseudoscalar for the space.

If $\mathbf{A}$ and $\mathbf{B}$ have a common factor (i.e., there exists a k-vector $\mathbf{C}$ such that $\mathbf{A}=\mathbf{A}'\mathbf{C}$ and $\mathbf{B}=\mathbf{B}'\mathbf{C}$ for some $\mathbf{A}',\mathbf{B}'$) then we can define the "intersection" or *meet* of $\mathbf{A}$ and $\mathbf{B}$ as $\mathbf{A}\vee\mathbf{B}$ [13] where

$$(\mathbf{A}\vee\mathbf{B})^*=\mathbf{A}^*\wedge\mathbf{B}^*. \tag{16}$$

That is, the dual of the meet is given by the join of the duals. In Eq. (16), we must be slightly careful to specify what space we take the dual of $(\mathbf{A}\vee\mathbf{B})$ with respect to. The dual of $(\mathbf{A}\vee\mathbf{B})$ is understood to be taken with respect to the *join* of $\mathbf{A}$ and $\mathbf{B}$. In most cases of practical interest this join will be the whole space and the meet is therefore easily computed so that we can use Eq. (13) to obtain a more useful expression for the meet as follows:

$$\mathbf{A}\vee\mathbf{B}=(\mathbf{A}^*\wedge\mathbf{B}^*)I=(\mathbf{A}^*\wedge\mathbf{B}^*)(I^{-1}I)I=(\mathbf{A}^*\cdot\mathbf{B}). \tag{17}$$

We therefore have the very simple and readily computed relation of $\mathbf{A}\vee\mathbf{B}=(\mathbf{A}^*\cdot\mathbf{B})$. The above concepts are discussed further in Ref. [13].

4. Conics and the Pascal's theorem

The role of the conics and quadrics is well known in the projective geometry [19]. This knowledge led to the solution of crucial problems in computer vision [20,21]. The Kruppa's equations which relies on the conic concept, have been used in the last decade to compute the intrinsic camera parameters [22]. In this work, we explore further the conics concept and use the Pascal's theorem to establish an equations system with clear geometric transparency. Next, we will explain the role of conics and that of Pascal's theorem in relation with a fundamental projective invariant. This section is mostly based on the interpretation of the linear algebra together with projective geometry in the Clifford algebra framework realized by Hestenes and Ziegler [13].

When we want to use projective geometry in computer vision we utilize homogeneous coordinates representations, doing that we embed the 3-D Euclidean visual space in the 3-D projective space P^3 or R^4 and the 2-D Euclidean space of the image plane in the 2-D projective space P^2 or R^3. In the geometric algebra framework, we select for P^2 the 3-D Euclidean geometric algebra $\mathcal{G}_{3,0,0}$ and for P^3 the 4-D geometric algebra $\mathcal{G}_{1,3,0}$. The reader should see Ref. [23] for more details relating the geometry of n cameras. Any geometric object of P^3 will be linear projective mapped to P^2 via a projective transformation, for example, the projective mapping of a quadric at infinity in the projective space P^3 results in a conic in the projective plane P^2.

Let us firstly consider a *pencil of lines* lying on the plane. Any pencil of lines is well defined by a bivector addition of two of its lines: $\mathbf{l}=\mathbf{l}_a+s\mathbf{l}_b$ with $s\in R\cup\{-\infty,+\infty\}$. If two pencils of lines $\mathbf{l}$ and $\mathbf{l}'=\mathbf{l}_a'+s'\mathbf{l}_b'$ can be related one-to-one so that $\mathbf{l}=\mathbf{l}'$ for $s=s'$ we can say that they are in projective correspondence. Using this idea, we will show that the set of intersecting points of lines in correspondence build a conic. Since the intersecting points $\mathbf{x}$ of the line pencils $\mathbf{l}$ and $\mathbf{l}'$ fulfill for $s=s'$ the following constraints:

$$\mathbf{x}\wedge\mathbf{l}=\mathbf{x}\wedge\mathbf{l}_a+s\,\mathbf{x}\wedge\mathbf{l}_b=0,$$
$$\mathbf{x}\wedge\mathbf{l}'=\mathbf{x}\wedge\mathbf{l}_a'+s\,\mathbf{x}\wedge\mathbf{l}_b'=0. \tag{18}$$

The elimination of the scalar s yields a second-order geometric product equation in $\mathbf{x}$

$$(\mathbf{x}\wedge\mathbf{l}_a)\,(\mathbf{x}\wedge\mathbf{l}_b')-(\mathbf{x}\wedge\mathbf{l}_b)\,(\mathbf{x}\wedge\mathbf{l}_a')=0. \tag{19}$$

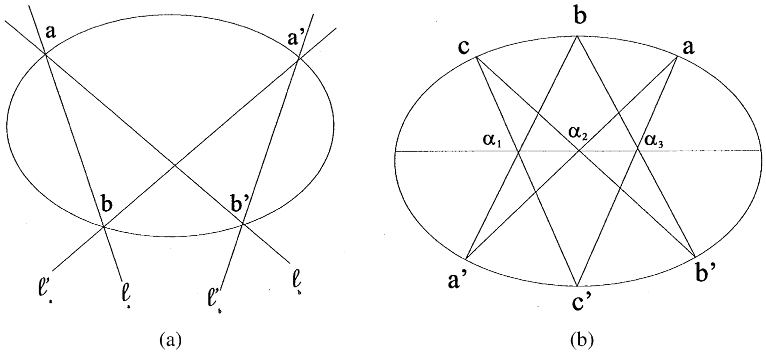


Fig. 2. (a) Two projective pencils generate a conic. (b) Pascal's theorem.

We can also get in another way the parameterized conic equation, simply computing the intersecting point $\mathbf{x}$ taking the meet of the line pencils as follows:

$$\mathbf{x} = (\mathbf{l}_a + s\mathbf{l}_b) \vee (\mathbf{l}'_a + s\mathbf{l}'_b)$$
$$= \mathbf{l}_a \vee \mathbf{l}'_a + s(\mathbf{l}_a \vee \mathbf{l}'_b + \mathbf{l}_b \vee \mathbf{l}'_a) + s^2\mathbf{l}_b \vee \mathbf{l}'_b. \tag{20}$$

Let us for now define the involved lines in terms of wedge of points $\mathbf{l}_a = \mathbf{a} \wedge \mathbf{b}$, $\mathbf{l}_b = \mathbf{a} \wedge \mathbf{b}'$, $\mathbf{l}'_a = \mathbf{a}' \wedge \mathbf{b}$ and $\mathbf{l}'_b = \mathbf{a}' \wedge \mathbf{b}'$ such that $\mathbf{l}_a \vee \mathbf{l}'_a = \mathbf{b}$ and $\mathbf{l}_b \vee \mathbf{l}'_b = \mathbf{b}'$ (see Fig. 2a). Calling $\mathbf{b}'' = \mathbf{l}_a \vee \mathbf{l}'_b + \mathbf{l}_b \vee \mathbf{l}'_a = \mathbf{d} + \mathbf{d}'$ in the last equation we get

$$\mathbf{x} = \mathbf{b} + s\mathbf{b}'' + s^2\mathbf{b}', \tag{21}$$

which represents a non-degenerated conic for $\mathbf{b} \wedge \mathbf{b}'' \wedge \mathbf{b}' = \mathbf{b} \wedge (\mathbf{d} + \mathbf{d}') \wedge \mathbf{b}' \neq 0$. Now, using this equation let us compute back the generating line pencils. Define $\mathbf{l}_1 = \mathbf{b}'' \wedge \mathbf{b}'$, $\mathbf{l}_2 = \mathbf{b}' \wedge \mathbf{b}$ and $\mathbf{l}_3 = \mathbf{b} \wedge \mathbf{b}''$ and compute its two projective pencils using Eq. (21)

$$\mathbf{b} \wedge \mathbf{x} = s\mathbf{b} \wedge \mathbf{b}'' + s^2\mathbf{b} \wedge \mathbf{b}' = s(\mathbf{l}_3 - s\mathbf{l}_2),$$
$$\mathbf{b}' \wedge \mathbf{x} = \mathbf{b}' \wedge \mathbf{b} + s\mathbf{b}' \wedge \mathbf{b}'' = \mathbf{l}_2 - s\mathbf{l}_1. \tag{22}$$

Considering the points $\mathbf{a}$, $\mathbf{a}'$, $\mathbf{b}$ and $\mathbf{b}'$ and some other point $\mathbf{y}$ lying in the conic depicted in the Fig. 2a and Eq. (18) for $s = \rho s'$ slightly different to s' we get the bracket expression

$$[\mathbf{yab}][\mathbf{ya'b'}] - \rho[\mathbf{yab'}][\mathbf{ya'b}] = 0,$$
$$\rho = \frac{[\mathbf{yab}][\mathbf{ya'b'}]}{[\mathbf{yab'}][\mathbf{ya'b}]} \tag{23}$$

for some $\rho \neq 0$. This equation is well known and represents a projective invariant which has been used quite a lot in real applications of computer vision. For a thorough study about the role of this invariant using brackets of points, lines, bilinearities and the trifocal tensor see Bayro and Lasenby [16]. Now evaluating ρ in terms of some other point $\mathbf{c}$ we get a conic equation fully represented in terms of brackets

$$[\mathbf{yab}][\mathbf{ya'b'}] - \frac{[\mathbf{cab}][\mathbf{ca'b'}]}{[\mathbf{cab'}][\mathbf{ca'b}]}[\mathbf{yab'}][\mathbf{ya'b}] = 0,$$

$$[\mathbf{yab}][\mathbf{ya'b'}][\mathbf{ab'c'}][\mathbf{a'bc'}]$$
$$- [\mathbf{yab'}][\mathbf{ya'b}][\mathbf{abc'}][\mathbf{a'b'c'}] = 0. \tag{24}$$

Again, we get a well-known concept which says that a conic is unique determined by the five points in general position $\mathbf{a}, \mathbf{a}'$, $\mathbf{b}$ and $\mathbf{b}'$ and $\mathbf{c}$. Now considering Fig. 2b, we can identify three collinear intersecting points $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$. Using the collinearity constraint and the lines which belong to pencils in projective correspondence we can write a very useful equation

$$\boldsymbol{\alpha}_1 \wedge \boldsymbol{\alpha}_2 \wedge \boldsymbol{\alpha}_3 = 0,$$

$$((\mathbf{a}' \wedge \mathbf{b}) \vee (\mathbf{c}' \wedge \mathbf{c})) \wedge ((\mathbf{a}' \wedge \mathbf{a}) \vee (\mathbf{b}' \wedge \mathbf{c})) \wedge$$
$$((\mathbf{c}' \wedge \mathbf{a}) \vee (\mathbf{b}' \wedge \mathbf{b})) = 0. \tag{25}$$

This expression is a geometric formulation using brackets of the Pascal's theorem. This theorem proves that the three intersecting points of the lines which connect opposite vertices of an hexagon circumscribed by a conic are collinear. Eq. (25) will be used in later section for computing the intrinsic camera parameters.

5. Computing the Kruppa equations in the geometric algebra

In this section, we will formulate in two ways the Kruppa equations in the geometric algebra framework. Firstly, we derive the Kruppa equation in its polynomial form using the bracket conic equation (24). Secondly, we formulate them in terms of purely brackets. The goal of the section is to compare the bracket representation with

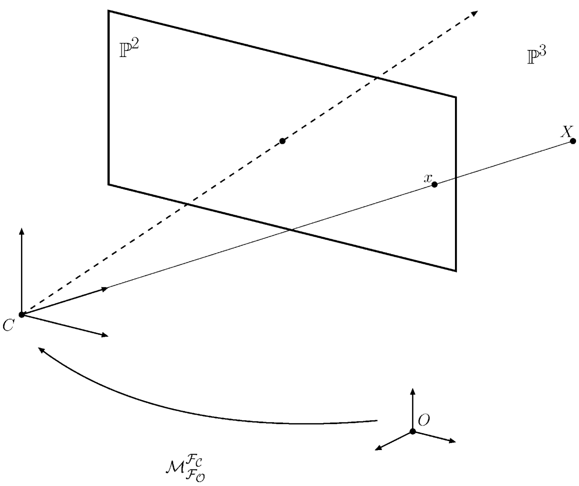


Fig. 3. The camera model: projective mapping of the visual 3-D space onto the image plane.

the standard Kruppa equations. In the first place, we will briefly explain how we model the camera transformation.

5.1. The scenario

When we use a camera to take images of the 3-D visual world the camera itself implements the projective transformation from P^3 to P^2 (see Fig. 3).

If the scene is near to the camera, this projective transformation in any object frame F_0 of P^3 is called *pinhole camera model* [24] and it is given by

$$\underbrace{\begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P_0} M_{F_0}^{F_c}, \tag{26}$$

where K is the matrix of an affine transformation and $M_{F_0}^{F_c}$ is a 3-D rigid motion which transforms the frame $\mathscr{F}_0$ to the frame $\mathscr{F}_C$ of the optical center $\mathbf{C}$ and it is given by the 4×4 matrix

$$M_{F_0}^{F_c} = \begin{bmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix}. \tag{27}$$

In this paper, we will set for the first camera $\mathscr{F}_0$ on $\mathscr{F}_C$, thus its projective transformation becomes

$$P_1 = KP_0 \begin{bmatrix} I & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} = K[I\,|\,0]. \tag{28}$$

Note that we use the notation $[I\,|\,0]$ for the resulting 3×4 matrix

$$P_0 \begin{bmatrix} I & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix}.$$

When we are considering any i-camera after a particular motion, its transformation regarding the frame F_0 set at the $\mathbf{C}$ of the first camera is given by

$$P_i = KP_0 \begin{bmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} = K[R\,|\,t]. \tag{29}$$

The scenario we will consider for the computations in the whole paper is depicted in Fig. 4. The camera model used here is the pinhole model and P_i stands for a projective transformation for the i-camera.

5.2. Standard Kruppa equations

This approach uses Eq. (24) for the conic in terms of brackets considering five points $\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}', \mathbf{c}'$ which lie on the conic:

$$[\mathbf{cab}][\mathbf{ca'b'}][\mathbf{ab'c'}][\mathbf{a'bc'}]$$
$$- [\mathbf{cab'}][\mathbf{ca'b}][\mathbf{abc'}][\mathbf{a'b'c'}] = 0,$$
$$[\mathbf{abc}][\mathbf{a'b'c}] - \frac{[\mathbf{a'b'c'}][\mathbf{abc'}]}{[\mathbf{ab'c'}][\mathbf{a'bc'}]}[\mathbf{ab'c}][\mathbf{a'bc}] = 0. \tag{30}$$

Now, a conic at infinite Ω_{inf} in P^3 can be defined employing any imaginer five points lying on the conic, e.g.,

$$\mathbf{A} = \begin{pmatrix} 1 \\ i \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} i \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{A}' = \begin{pmatrix} i \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{B}' = \begin{pmatrix} 1 \\ 0 \\ i \\ 0 \end{pmatrix}, \quad \mathbf{C}' = \begin{pmatrix} 0 \\ i \\ 1 \\ 0 \end{pmatrix}, \tag{31}$$

where $i^2 = -1$. Note that we use upper case letters to represent points of the projective space $\mathscr{P}^3$ represented in $\mathscr{G}_{1,3,0}$. These points lie on the conic, thus they fulfill the propriety $\mathbf{A}\cdot\mathbf{A} = \mathbf{B}\cdot\mathbf{B} = \mathbf{A}'\cdot\mathbf{A}' = \mathbf{B}'\cdot\mathbf{B}' = \mathbf{C}'\cdot\mathbf{C}' = 0$. Since these points are translation invariant their projections in any image plane are given by $\mathbf{a} = K[R\,|\,t]\mathbf{A} = KR\mathbf{A}$, $\mathbf{b} = K[R\,|\,t]\mathbf{B} = KR\mathbf{B}$, $\mathbf{a}' = K[R\,|\,t]\mathbf{A}' = KR\mathbf{A}'$, $\mathbf{b}' = K[R\,|\,t]\mathbf{B}' = KR\mathbf{B}'$, $\mathbf{c}' = K[R\,|\,t]\mathbf{C}' = KR\mathbf{C}'$. The rotated points $R^{\mathrm{T}}\mathbf{A}$, $R^{\mathrm{T}}\mathbf{B}$, $R^{\mathrm{T}}\mathbf{A}'$, $R^{\mathrm{T}}\mathbf{B}'$ and $R^{\mathrm{T}}\mathbf{C}'$ lie also at the conic at infinite, because they fulfill the property $\mathbf{A}(RR^{\mathrm{T}})\cdot\mathbf{A} = \mathbf{B}(RR^{\mathrm{T}})\cdot\mathbf{B} = \mathbf{A}'(RR^{\mathrm{T}})\cdot\mathbf{A}' = \mathbf{B}'(RR^{\mathrm{T}})\cdot\mathbf{B}' = \mathbf{C}'(RR^{\mathrm{T}})\cdot\mathbf{C}' = 0$. Using these rotated points the rotation R of the camera transformation in Eq. (30) is canceled, thus this equation depends only of K

$$[KRR^{\mathrm{T}}\mathbf{A}KRR^{\mathrm{T}}\mathbf{Bc}][KRR^{\mathrm{T}}\mathbf{A}'KRR^{\mathrm{T}}\mathbf{B'c}]$$
$$- \frac{[KRR^{\mathrm{T}}\mathbf{A}'KRR^{\mathrm{T}}\mathbf{B}'KRR^{\mathrm{T}}\mathbf{C}'][KRR^{\mathrm{T}}\mathbf{A}KRR^{\mathrm{T}}\mathbf{B}KRR^{\mathrm{T}}\mathbf{C}']}{[KRR^{\mathrm{T}}\mathbf{A}KRR^{\mathrm{T}}\mathbf{B}'KRR^{\mathrm{T}}\mathbf{C}'][KRR^{\mathrm{T}}\mathbf{A}'KRR^{\mathrm{T}}\mathbf{B}KRR^{\mathrm{T}}\mathbf{C}']}$$
$$\cdot [KRR^{\mathrm{T}}\mathbf{A}KRR^{\mathrm{T}}\mathbf{B'c}][KRR^{\mathrm{T}}\mathbf{A}'KRR^{\mathrm{T}}\mathbf{Bc}] = 0,$$
$$\Leftrightarrow [KAK\mathbf{Bc}][K\mathbf{A}'K\mathbf{B'c}]$$

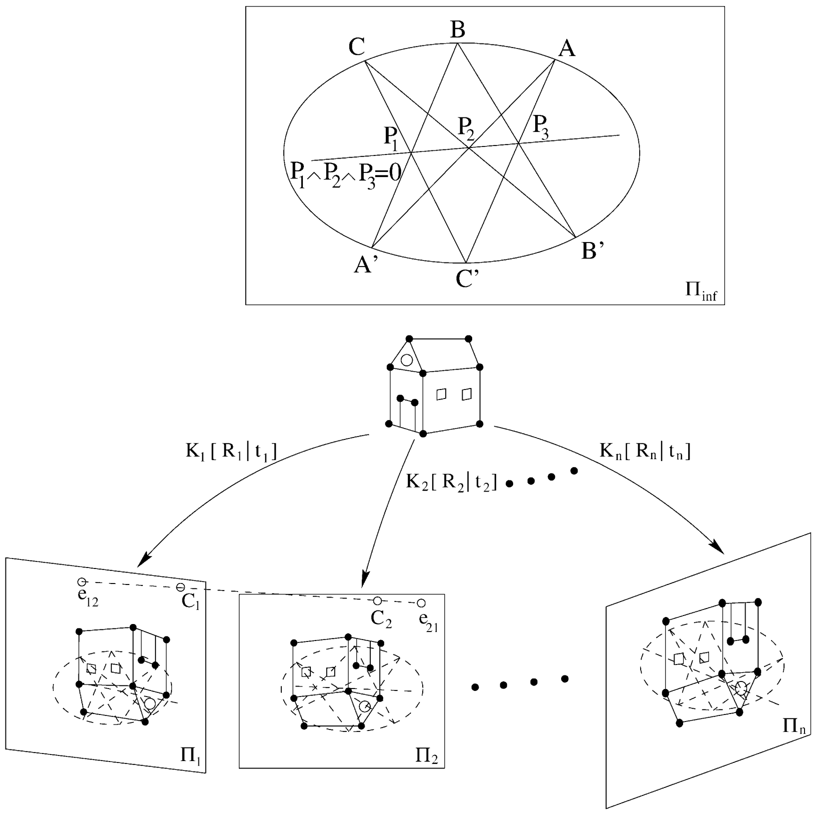


Fig. 4. The conics at infinity, the real 3-D visual space and n uncalibrated cameras.

$$-\frac{[K\mathbf{A}'K\mathbf{B}'K\mathbf{C}'][K\mathbf{A}K\mathbf{B}K\mathbf{C}']}{[K\mathbf{A}K\mathbf{B}'K\mathbf{C}'][K\mathbf{A}'K\mathbf{B}K\mathbf{C}']} \cdot [K\mathbf{A}K\mathbf{B}'\mathbf{c}][K\mathbf{A}'K\mathbf{B}\mathbf{c}] = 0. \tag{32}$$

We can further extract of the brackets the determinant of the intrinsic parameters in the multiplicative ratio of the previous equation reducing the invariant to a constant

$$\begin{aligned} Inv &= \frac{[K\mathbf{A}'K\mathbf{B}'K\mathbf{C}'][K\mathbf{A}K\mathbf{B}K\mathbf{C}']}{[K(\mathbf{A}\mathbf{B}'\mathbf{C}')][K(\mathbf{A}'\mathbf{B}\mathbf{C}')]} \\ &= \frac{[K(\mathbf{A}'\mathbf{B}'\mathbf{C}')][K(\mathbf{A}\mathbf{B}\mathbf{C}')]}{[K(\mathbf{A}\mathbf{B}'\mathbf{C}')][K(\mathbf{A}'\mathbf{B}\mathbf{C}')]} \\ &= \frac{det(K)[(\mathbf{A}'\mathbf{B}'\mathbf{C}')]det(K)[(\mathbf{A}\mathbf{B}\mathbf{C}')]}{det(K)[(\mathbf{A}\mathbf{B}'\mathbf{C}')]det(K)[(\mathbf{A}'\mathbf{B}\mathbf{C}')]} \\ &= \frac{[(\mathbf{A}'\mathbf{B}'\mathbf{C}')][(\mathbf{A}\mathbf{B}\mathbf{C}')]}{[(\mathbf{A}\mathbf{B}'\mathbf{C}')][(\mathbf{A}'\mathbf{B}\mathbf{C}')]}. \end{aligned} \tag{33}$$

Substituting the previous values of $\mathbf{A}, \mathbf{B}, \mathbf{A}, \mathbf{A}', \mathbf{B}', \mathbf{C}'$ in this equation we get the value of $Inv = 2$, this value will be used for further computations later on. Eq. (33) is as expected invariant to the affine transformation K. Thus, the bracket equation (23) of the projective invariant can be written as

$$[K\mathbf{A}K\mathbf{B}\mathbf{c}][K\mathbf{A}'K\mathbf{B}'\mathbf{c}] - Inv[K\mathbf{A}K\mathbf{B}'\mathbf{c}][K\mathbf{A}'K\mathbf{B}\mathbf{c}] = 0. \tag{34}$$

In terms of matrices the relation of a point $\mathbf{c}$ which lies on the image of the absolute conic $\mathscr{C}$ at infinity or a line tangent $\mathbf{l}_c$ to the dual of this conic $\mathscr{C}^*$ can be expressed as

$$0 = \mathbf{c}^{\mathrm{T}}\mathscr{C}\mathbf{c} = \mathbf{c}^{\mathrm{T}}\mathscr{C}^{\mathrm{T}}\mathbf{c} = (\mathbf{c}^{\mathrm{T}}\mathscr{C}^{\mathrm{T}})\mathscr{C}^{-1}(\mathscr{C}\mathbf{c}) = \mathbf{l}_c^{\mathrm{T}}\mathscr{C}^*\mathbf{l}_c, \tag{35}$$

where $\mathscr{C} = K^{-\mathrm{T}}K^{-1}$ and $\mathscr{C}^* \sim \mathscr{C}^{-1} = KK^{\mathrm{T}}$. In this equation, according to the duality principle, the points and lines are related according $\mathscr{C}\mathbf{c} = \mathbf{l}_c$. Thus, it is true $\mathbf{c} = KK^{\mathrm{T}}\mathbf{l}_c$. Substituting this value in Eq. (34) we get

$$\begin{aligned} &[K\mathbf{A}K\mathbf{B}\mathbf{c}][K\mathbf{A}'K\mathbf{B}'\mathbf{c}] - Inv[K\mathbf{A}K\mathbf{B}'\mathbf{c}][K\mathbf{A}'K\mathbf{B}\mathbf{c}] = 0, \\ &[K\mathbf{A}K\mathbf{B}KK^{\mathrm{T}}\mathbf{l}_c][K\mathbf{A}'K\mathbf{B}'KK^{\mathrm{T}}\mathbf{l}_c] \\ &\quad - Inv[K\mathbf{A}K\mathbf{B}'KK^{\mathrm{T}}\mathbf{l}_c][K\mathbf{A}'K\mathbf{B}KK^{\mathrm{T}}\mathbf{l}_c] = 0, \\ &[K(\mathbf{A}\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c)][K(\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c)] \\ &\quad - Inv[K(\mathbf{A}\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c)][K(\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c)] = 0, \end{aligned}$$

$$det(K)[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c]det(K)[\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c]$$
$$-\,Inv\,det(K)[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c]det(K)[\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]=0,$$
$$[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c]$$
$$-\,Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]=0. \quad (36)$$

As in the next formulas, we will utilize the so-called *epipole* for the computation, let us explain it briefly. An epipole is the null vector of the fundamental matrix F. This matrix fulfills the constraint $\mathbf{a}_i^{\mathrm{T}}F\mathbf{a}_i'=0$ for any correspondent points of a couple of cameras (see Ref. [9] for more details).

Returning to the problem we see that fortunately the matrix of the intrinsic camera parameters turns up in each bracket only once. Using k_{ij} of K and the points of the conic at infinity $\mathbf{A},\mathbf{A}',\mathbf{B},\mathbf{B}',\mathbf{C}'$ and the epipolar line $\mathbf{l}_c=\mathbf{e}\times\mathbf{y}=[p_1,p_2,p_3]^{\mathrm{T}}\times[1,\tau,0]^{\mathrm{T}}=[-p_3\tau,p_3,p_1\tau-p_2]^{\mathrm{T}}$ ($\mathbf{e}$ is the epipole and the point $\mathbf{y}=[1,\tau,0]^{\mathrm{T}}$ which lies at the line at infinity parameterized with the projective parameter τ), we can write

$$K^{\mathrm{T}}\mathbf{l}_c=\begin{pmatrix}-k_{11}p_3\tau\\ -k_{12}p_3\tau+k_{22}p_3\\ -k_{13}p_3\tau+k_{23}p_3+p_1\tau-p_2\end{pmatrix}, \quad (37)$$

which together with the value for $Inv=2$ simplifies Eq. (36) to a second-order polynomial with respect to τ as follows:

$$[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c]-Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]$$
$$=4p_1\tau p_2-2p_1^2\tau^2-2k_{22}^2p_3^2-4k_{23}p_3p_1\tau+4k_{23}p_3p_2$$
$$-2k_{13}^2p_3^2\tau^2-2k_{12}^2p_3^2\tau^2-2k_{23}^2p_3^2$$
$$-2p_2^2-2k_{11}^2p_3^2\tau^2+4k_{12}p_3^2\tau k_{22}-4k_{13}p_3\tau p_2$$
$$+4k_{13}p_3^2\tau k_{23}+4k_{13}p_3\tau^2p_1. \quad (38)$$

Expressing the polynomial in the form $P(\tau)=k_0+k_1\tau+k_2\tau^2$, we get finally the following coefficients:

$$k_0=-2k_{22}^2p_3^2+4k_{23}p_3p_2-2k_{23}^2p_3^2-2p_2^2,$$
$$k_1=4p_1p_2-4k_{23}p_3p_1+4k_{12}p_3^2k_{22}$$
$$-4k_{13}p_3p_2+4k_{13}p_3^2k_{23},$$
$$k_2=-2p_1^2-2k_{13}^2p_3^2-2k_{12}^2p_3^2-2k_{11}^2p_3^2+4k_{13}p_3p_1. \quad (39)$$

$\mathbf{l}_c$ can also be considered as an epipolar line tangent to the conic in the first camera, thus according to the homography of a point lying at the line at infinity of the second camera we can compute $\mathbf{l}_c=F[1,\tau,0]^{\mathrm{T}}$. Note that the fundamental matrix can be expressed in terms of the motion between cameras and the K of the camera, i.e., $F=K^{-\mathrm{T}}[t]_xR_{12}K^{\mathrm{T}}$ where $[t]_x$ is the tensor notation of the antisymmetric matrix representing the translation [9]. The term $E=[t]_\times R$ is called the essential matrix [25]. Using the new expression of $\mathbf{l}_c$ we can gain similarly as above new equations for the coefficients now called k_{ij}'. In this computation, F relates the second camera. Taking now these equations for the two cameras we can write the well-known Kruppa's relations

$$k_2k_1'-k_2'k_1=0,$$
$$k_0k_1'-k_0'k_1=0,$$
$$k_0k_2'-k_0'k_2=0. \quad (40)$$

We get up to a scalar factor the same Kruppa's equations presented by Luong and Faugeras [24]. The scalar factor is present in all of these equations, thus it can be canceled straightforwardly. The algebraic manipulation of this formulas was checked entirely using a Maple program.

5.3. Kruppa equations using brackets

In this section, we will formulate the Kruppa coefficients k_0,k_1,k_2 of the polynomial $P(\tau)$ in terms of brackets. This kind of representation will obviously elucidate the involved geometry. First, let us consider again the bracket $[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c]$ of Eq. (36). Since we can write

$$K^{\mathrm{T}}\mathbf{l}_c=K^{\mathrm{T}}\begin{pmatrix}-p_3\tau\\ p_3\\ p_1\tau-p_2\end{pmatrix}=K^{\mathrm{T}}\begin{pmatrix}0\\ p_3\\ -p_2\end{pmatrix}+K^{\mathrm{T}}\begin{pmatrix}-p_3\\ 0\\ p_1\end{pmatrix}\tau, \quad (41)$$

the bracket can be split in two brackets, one independent of τ and another depending of it

$$[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c]=\left[\mathbf{AB}K^{\mathrm{T}}\begin{pmatrix}0\\ p_3\\ -p_2\end{pmatrix}\right]+\left[\mathbf{AB}K^{\mathrm{T}}\begin{pmatrix}-p_3\\ 0\\ p_1\end{pmatrix}\tau\right]. \quad (42)$$

In short, $[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c]=a_1+\tau b_1$. Now using this bracket representation, Eq. (36) can be written as

$$[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c]-Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]=0,$$
$$(a_1+\tau b_1)(a_2+\tau b_2)-Inv(a_3+\tau b_3)(a_4+\tau b_4)=0,$$
$$a_1a_2+\tau b_1a_2+a_1\tau b_2+\tau^2b_1b_2$$
$$-Inv(a_3a_4+a_3a_4\tau+b_3a_4\tau+b_3b_4\tau^2)=0,$$

$$\underbrace{a_1a_2-Inv(a_3a_4)}_{k_0}$$
$$+\tau\underbrace{(a_1b_2+b_1a_2-Inv(a_3b_4+a_4b_3))}_{k_1}$$
$$+\tau^2\underbrace{(b_1b_2-Inv(b_3b_4))}_{k_2}=0. \quad (43)$$

Now, let us take a partial vector part of $K^{\mathrm{T}}\mathbf{l}_c$ and call it $K^{\mathrm{T}}\mathbf{l}_{c1}:=[-k_{11}p_3,-k_{12}p_3,-k_{13}p_3+p_1]^{\mathrm{T}}$ and the "rest"-part as $K^{\mathrm{T}}\mathbf{l}_{c2}:=[0,k_{22}p_3,k_{23}p_3-p_2]^{\mathrm{T}}$. Using both parts, we can write the coefficients of the polynomial in a bracket form as follows:

$$k_0 = [\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c2}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_{c2}], \tag{44}$$

$$k_1 = [\mathbf{AB}K^{\mathrm{T}}\mathbf{t}_t][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c2}] + [\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c1}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_{c1}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c1}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_{c2}], \tag{45}$$

$$k_2 = [\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_{c1}][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c1}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c1}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_{c1}]. \tag{46}$$

Since $\mathbf{A}, \mathbf{B}, \mathbf{A}', \mathbf{B}'$ and Inv are known, given an epipole $\mathbf{e} = (p_1, p_2, p_3)^{\mathrm{T}}$ we can finally compute the coefficients k_0, k_1 and k_2 straightforwardly. Be aware that these coefficients denoted by k_i are not the intrinsic parameters k_{ij}. The striking aspect of these equations is twofold. They are expressed in terms of brackets and they depend of the invariant real magnitude Inv. This can certainly help us to explore the involved geometry of the Kruppa equations.

Let us first analyze the k's. It should be sufficient from k_0, k_1 and k_2 to explore the involved geometry if k_0 and k_2 are expressed as follows:

$$\begin{aligned} k_0 &= a_1 a_2 - Inv(a_3 a_4) = [\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c2}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c2}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_{c2}] \\ &= \left[\begin{pmatrix} 1 & i & 0 \\ i & 1 & k_{22}p_3 \\ 0 & 0 & k_{23}p_3 - p_2 \end{pmatrix}\right]\left[\begin{pmatrix} i & 1 & 0 \\ 0 & 0 & k_{22}p_3 \\ 1 & i & k_{23}p_3 - p_2 \end{pmatrix}\right] \\ &\quad - Inv\left[\begin{pmatrix} 1 & 1 & 0 \\ i & 0 & k_{22}p_3 \\ 0 & i & k_{23}p_3 - p_2 \end{pmatrix}\right]\left[\begin{pmatrix} i & i & 0 \\ 0 & 1 & k_{22}p_3 \\ 1 & 0 & k_{23}p_3 - p_2 \end{pmatrix}\right], \end{aligned} \tag{47}$$

$$\begin{aligned} k_2 &= b_1 b_2 - Inv(b_3 b_4) = [\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_{c1}][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_{c1}] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_{c1}][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{t}_t] \\ &= \left[\begin{pmatrix} 1 & i & -k_{11}p_3 \\ i & 1 & -k_{12}p_3 \\ 0 & 0 & -k_{13}p_3 + p_1 \end{pmatrix}\right] \cdot \left[\begin{pmatrix} i & 1 & -k_{11}p_3 \\ 0 & 0 & -k_{12}p_3 \\ 1 & i & -k_{13}p_3 + p_1 \end{pmatrix}\right] \\ &\quad - Inv\left[\begin{pmatrix} 1 & 1 & -k_{11}p_3 \\ i & 0 & -k_{12}p_3 \\ 0 & i & -k_{13}p_3 + p_1 \end{pmatrix}\right] \cdot \left[\begin{pmatrix} i & i & -k_{11}p_3 \\ 0 & 1 & -k_{12}p_3 \\ 1 & 0 & -k_{13}p_3 + p_1 \end{pmatrix}\right]. \end{aligned} \tag{48}$$

Let us analyze the effect of the camera motions in these two equations. If the camera moves on a straight path parallel to the object the epipole lies at infinity, i.e., $p_3 = 0$, in this case the intrinsic parameters become zero resulting a trivial polynomial, i.e., we cannot get the coefficients k'_0, k'_1 and k'_2. On the other hand, for example trying the values $-k_{13}p_3 + p_1 = 0$ or $k_{23}p_3 - p_2 = 0$ the rest of the brackets will have the rank two and their determinant value is also zero. This simple analysis shows that analyzing the brackets for certain kinds of camera motions we can avoid certain camera motions which generate trivial Kruppa's equations. It is also interesting to see that for $k_0 = 0$ and $k_2 = 0$ we have also conic equations. So in order to avoid trivial equations we have to consider always $k_0 \neq 0$ and $k_2 \neq 0$. In other words, $K^{\mathrm{T}}\mathbf{l}_{c1}$ and $K^{\mathrm{T}}\mathbf{l}_{c2}$ should not lie on the image of the absolute conic.

Now let us consider the invariant real magnitude Inv of the Kruppa's bracket equation (36).

$$[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c] - Inv[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c] = 0,$$

$$Inv = \frac{[\mathbf{AB}K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}'K^{\mathrm{T}}\mathbf{l}_c]}{[\mathbf{AB}'K^{\mathrm{T}}\mathbf{l}_c][\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]}. \tag{49}$$

That the invariant value Inv like in Eq. (23) plays a role in the Kruppa's equations is a fact that has been over-seen so far. This can be simply explained as the fact that when we formulate the Kruppa's equation using the condition $\mathbf{c}^{\mathrm{T}}\mathscr{C}\mathbf{c} = 0$, we are actually implicitly employing the invariant given by Eqs. (23) and (49).

6. Camera calibration using the Pascal's theorem

This section presents a new technique in the geometric algebra framework for computing the intrinsic camera parameters. The previous section used the equation of Eq. (24) to compute the Kruppa's coefficients which in turn can be used to get the intrinsic camera parameters. Along this lines we will proceed here.

In Section 4, it is shown that Eq. (24) can be reformulated to express the constraint of Eq. (25) known as Pascal's theorem. Since the Pascal's equation fulfill a property of any conic, it should be also possible using this equation to compute the intrinsic camera parameters. Let us consider the three intersecting points which are collinear and fulfill

$$\underbrace{((\mathbf{a}'\wedge\mathbf{b})\vee(\mathbf{c}'\wedge\mathbf{c}))}_{\alpha_1}\wedge\underbrace{((\mathbf{a}'\wedge\mathbf{a})\vee(\mathbf{b}'\wedge\mathbf{c}))}_{\alpha_2}$$
$$\wedge\underbrace{((\mathbf{c}'\wedge\mathbf{a})\vee(\mathbf{b}'\wedge\mathbf{b}))}_{\alpha_3}=0. \tag{50}$$

In Fig. 5, at the first camera, the projected rotated points of the conic at infinity $R^{\mathrm{T}}\mathbf{A}$, $R^{\mathrm{T}}\mathbf{B}$, $R^{\mathrm{T}}\mathbf{A}'$, $R^{\mathrm{T}}\mathbf{B}'$, $R^{\mathrm{T}}\mathbf{C}'$ are $\mathbf{a}=K[R\,|\,0]R^{\mathrm{T}}\mathbf{A}=K\mathbf{A}$, $\mathbf{b}=K[R\,|\,0]R^{\mathrm{T}}\mathbf{B}=K\mathbf{B}$, $\mathbf{a}'=K[R\,|\,0]R^{\mathrm{T}}\mathbf{A}'=K\mathbf{A}'$, $\mathbf{b}'=K[R\,|\,0]R^{\mathrm{T}}\mathbf{B}'=K\mathbf{B}'$ and $\mathbf{c}'=K[R\,|\,0]\mathbf{C}'=K\mathbf{C}'$.

The point $\mathbf{c}=KK^{\mathrm{T}}\mathbf{l}_c$ depends of the intrinsic parameters and of the line $\mathbf{l}_c$ tangent to the conic computed in terms of the epipole $=[p_1,p_2,p_3]^{\mathrm{T}}$ and a point lying at the line at infinity of the first camera $\mathbf{l}_c=[p_1,p_2,p_3]^{\mathrm{T}}\times[1,\tau,0]^{\mathrm{T}}$. Now using this expression for $\mathbf{l}_c$ we can simplify Eq. (50) and get the bracket equations of the $\boldsymbol{\alpha}$'s

$$([\mathbf{a}'\mathbf{b}\mathbf{c}']\mathbf{c}-[\mathbf{a}'\mathbf{b}\mathbf{c}]\mathbf{c}')\wedge([\mathbf{a}'\mathbf{a}\mathbf{b}']\mathbf{c}-[\mathbf{a}'\mathbf{a}\mathbf{c}]\mathbf{b}')$$
$$\wedge([\mathbf{c}'\mathbf{a}\mathbf{b}']\mathbf{b}-[\mathbf{c}'\mathbf{a}\mathbf{b}]\mathbf{b}')=0$$

$$\Leftrightarrow([K\mathbf{A}'K\mathbf{B}K\mathbf{C}']KK^{\mathrm{T}}\mathbf{l}_c-[K\mathbf{A}'K\mathbf{B}KK^{\mathrm{T}}\mathbf{l}_c]K\mathbf{C}')$$
$$\wedge([K\mathbf{A}'K\mathbf{A}K\mathbf{B}']KK^{\mathrm{T}}\mathbf{l}_c-[K\mathbf{A}'K\mathbf{A}KK^{\mathrm{T}}\mathbf{l}_c]K\mathbf{B}')$$
$$\wedge([K\mathbf{C}'K\mathbf{A}K\mathbf{B}']K\mathbf{B}-[K\mathbf{C}'K\mathbf{A}K\mathbf{B}]K\mathbf{B}')=0, \tag{51}$$

$$(det(K)K([\mathbf{A}'\mathbf{B}\mathbf{C}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{C}'))$$
$$\wedge(det(K)K([\mathbf{A}'\mathbf{A}\mathbf{B}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{A}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{B}'))$$
$$\wedge(det(K)K([\mathbf{C}'\mathbf{A}\mathbf{B}']\mathbf{B}-[\mathbf{C}'\mathbf{A}\mathbf{B}]\mathbf{B}'))=0 \tag{52}$$

$$\Leftrightarrow det(K)^3K([\mathbf{A}'\mathbf{B}\mathbf{C}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{C}')$$
$$\wedge([\mathbf{A}'\mathbf{A}\mathbf{B}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{A}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{B}')$$
$$\wedge([\mathbf{C}'\mathbf{A}\mathbf{B}']\mathbf{B}-[\mathbf{C}'\mathbf{A}\mathbf{B}]\mathbf{B}')=0$$

$$\Leftrightarrow\underbrace{([\mathbf{A}'\mathbf{B}\mathbf{C}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{B}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{C}')}_{\alpha_1}$$
$$\wedge\underbrace{([\mathbf{A}'\mathbf{A}\mathbf{B}']K^{\mathrm{T}}\mathbf{l}_c-[\mathbf{A}'\mathbf{A}K^{\mathrm{T}}\mathbf{l}_c]\mathbf{B}')}_{\alpha_2}$$
$$\wedge\underbrace{([\mathbf{C}'\mathbf{A}\mathbf{B}']\mathbf{B}-[\mathbf{C}'\mathbf{A}\mathbf{B}]\mathbf{B}')}_{\alpha_3}=0. \tag{53}$$

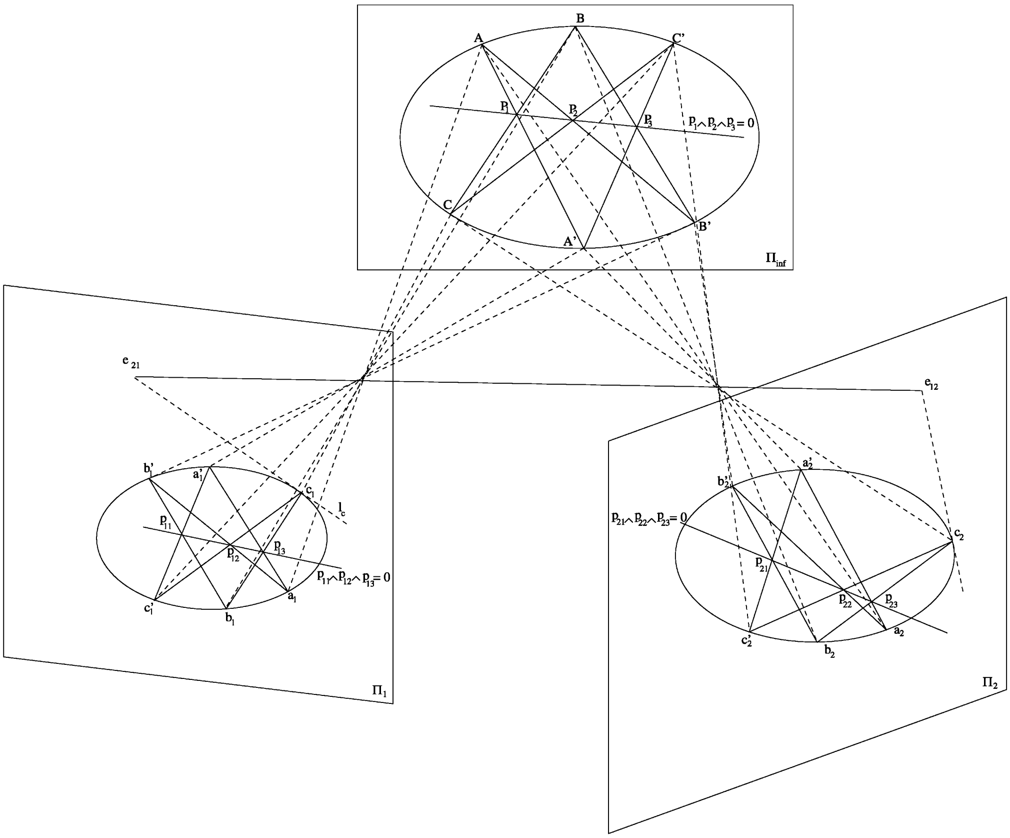


Fig. 5. Pascal's theorem at the images conics.

Note that the scalar $det(K)^3$ and K are canceled out simplifying the expression for the $\boldsymbol{\alpha}$'s. In the next section, the computation of the intrinsic parameters will be done first considering that the intrinsic parameters remain stationary under camera motions and second when these parameters change.

6.1. Computing stationary intrinsic parameters

Let us assume that the $\mathscr{F}_0$ is attached to the optical center of the first camera and consider a second camera which has a motion with respect to the first one of $[R_1 \mid t_1]$. Accordingly, the involved projective transformations are given by

$$P_1 = K[I \mid 0]^{-1}, \tag{54}$$

$$P_2 = P_1[R_1 \mid t_1]^{-1} \tag{55}$$

and their optical centres by $\mathbf{C}_1 = (0,0,0,1)^{\mathrm{T}}$ and $\mathbf{C}_2 = [R_1 \mid t_1]\mathbf{C}_1$. Thus, we can compute their epipoles as $\mathbf{e}_{21} = P_2\mathbf{C}_1$, $\mathbf{e}_{12} = P_1\mathbf{C}_2$.

Next, we will show by means of an example that the coordinates of the points $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$ are entirely independent of the intrinsic parameters. This condition is necessary for solving the problem. Let us choose, for example, a camera motion given by

$$[R_1 \mid t_1] = \begin{pmatrix} 0 & -1 & 0 & | & 2 \\ 1 & 0 & 0 & | & -1 \\ 0 & 0 & 1 & | & 3 \end{pmatrix}. \tag{56}$$

For this motion, the epipoles are $\mathbf{e}_{12} = (2k_{11} - k_{12} + 3k_{13}, -k_{22} + 3k_{23}, 3)^{\mathrm{T}}$ and $\mathbf{e}_{21} = (k_{11} + 2k_{12} - 3k_{13}, 2k_{22} - 3k_{23}, -3)^{\mathrm{T}}$. Now using the rotated conic points given by Eq. (31) and replacing this e_{12} in Eq. (53) we can make explicit the $\boldsymbol{\alpha}$'s:

$$\boldsymbol{\alpha}_1 = \begin{pmatrix} (-3+3i)k_{11}\tau \\ 3k_{11}\tau - ik_{12}\tau + ik_{22} + 2ik_{11}\tau - 3k_{12}\tau + 3k_{22} \\ ik_{11}\tau + 3k_{12}\tau - 3k_{22} + ik_{12}\tau - ik_{22} \end{pmatrix},$$

$$\boldsymbol{\alpha}_2 = \begin{pmatrix} -3ik_{11}\tau - 2k_{12}\tau + 2k_{22} - 2k_{11}\tau \\ -6i(k_{12}\tau - k_{22}) \\ -3k_{11}\tau - 4ik_{12}\tau + 4ik_{22} + 2ik_{11}\tau \end{pmatrix},$$

$$\boldsymbol{\alpha}_3 = \begin{pmatrix} 1-i \\ 1-i \\ 2 \end{pmatrix}. \tag{57}$$

Note that $\boldsymbol{\alpha}_3$ is fully independent of K.

According to the Pascal's theorem, these three points lie on the same line, therefore by replacing these points in Eq. (25) we get the following second-order polynomial in τ:

$$-40Ik_{12}^2\tau^2 - 52Ik_{11}^2\tau^2 - 16Ik_{11}\tau k_{22} + 16Ik_{11}\tau^2 k_{12} - 40Ik_{22}^2 + 80Ik_{12}\tau k_{22} = 0. \tag{58}$$

Solving this polynomial and choosing one of the solutions which is nothing else than the solution for one of the two lines tangent to the conic we get

$$\tau := \frac{16Ik_{11}k_{22} - 80Ik_{12}k_{22} + 24\sqrt{14}k_{11}k_{22}}{2(-40Ik_{12}^2 - 52Ik_{11}^2 + 16Ik_{11}k_{12})}. \tag{59}$$

Now, considering the homogeneous representation of these points

$$\boldsymbol{\alpha}_i = [\alpha_{i1}, \alpha_{i2}, \alpha_{i3}]^{\mathrm{T}} \sim \left[\frac{\alpha_{i1}}{\alpha_{i3}}, \frac{\alpha_{i2}}{\alpha_{i3}}, 1\right]^{\mathrm{T}}, \tag{60}$$

we can finally express their coordinates as follows:

$$\alpha_{11} = \frac{-(2k_{11} - 10k_{12} + 3ik_{11}\sqrt{14} + 8ik_{12} + 2k_{12}\sqrt{14} - 10ik_{11} + 2\sqrt{14}k_{11})}{2ik_{11} - 10ik_{12} - 3\sqrt{14}k_{11} - 4k_{12} - 4ik_{12}\sqrt{14} - 16k_{11} + 2ik_{11}\sqrt{14}}, \tag{61}$$

$$\alpha_{12} = \frac{2i(-2ik_{12} - 3k_{12}\sqrt{14} + 13ik_{11})}{2ik_{11} - 10ik_{12} - 3\sqrt{14}k_{11} - 4k_{12} - 4ik_{12}\sqrt{14} - 16k_{11} + 2ik_{11}\sqrt{14}}, \tag{62}$$

$$\alpha_{21} = \frac{(1-i)(2ik_{11} - 10ik_{12} - 3\sqrt{14}k_{11})}{5k_{11} - 4k_{12} + ik_{11}\sqrt{14} + 2ik_{12} + 3k_{12}\sqrt{14} - 13ik_{11} + ik_{12}\sqrt{14}}, \tag{63}$$

$$\alpha_{22} = \frac{11ik_{11} + 8ik_{12} + 3\sqrt{14}k_{11} - 6k_{12} - ik_{12}\sqrt{14} - 3k_{11} + 2ik_{11}\sqrt{14} - 3k_{12}\sqrt{14}}{5k_{11} - 4k_{12} + ik_{11}\sqrt{14} + 2ik_{12} + 3k_{12}\sqrt{14} - 13ik_{11} + ik_{12}\sqrt{14}}. \tag{64}$$

Now considering the case of exact orthogonal image axis we can set in previous equation $k_{12} = 0$ and get

$$\alpha_{11} = \frac{2i - 3\sqrt{14} + 10 + 2i\sqrt{14}}{2 + 3i\sqrt{14} + 16i + 2\sqrt{14}}, \tag{65}$$

$$\alpha_{12} = 26\,\frac{i}{2 + 3i\sqrt{14} + 16i + 2\sqrt{14}}, \tag{66}$$

$$\alpha_{21} = \frac{(1+i)(-2I + 3\sqrt{14})}{-5i + \sqrt{14} - 13}, \tag{67}$$

$$\alpha_{22} = -\frac{-11 + 3i\sqrt{14} - 3i - 2\sqrt{14}}{-5i + \sqrt{14} - 13}. \tag{68}$$

The coordinates are indeed independent of the intrinsic parameters. We test with the Maple program several times using other conic points, we get always expressions for the α's independent of the intrinsic parameters.

After this illustration by an example we will get now the coordinates using any camera motion, for that let us define $\mathbf{s} = [s_1, s_2, s_3]^{\mathrm{T}} = [I\,|\,0][R_1\,|\,t_1]C_1$. Using this value, the epipole is $e_{12} = K[I\,|\,0][R_1\,|\,t_1]C_1 = K\mathbf{s}$. Note that in this expression the intrinsic parameters are separate from the extrinsic ones. Similar as above using the general camera motion and the epipole value the coordinates for the intersecting points read

$$\alpha_{11} = -\frac{(-s_1s_2 + Is_3\sqrt{(s_1^2+s_2^2+s_3^2)} - Is_3^2 - Is_1^2 + s_1\sqrt{(s_1^2+s_2^2+s_3^2)} - Is_2s_3)}{(-Is_1s_2 - s_3\sqrt{(s_1^2+s_2^2+s_3^2)} - s_3^2 - s_1^2 + Is_1\sqrt{(s_1^2+s_2^2+s_3^2)} + s_2s_3)}, \tag{69}$$

$$\alpha_{21} = \frac{-2(s_3^2+s_1^2)}{-Is_2s_1 - s_3\sqrt{(s_1^2+s_2^2+s_3^2)} - s_3^2 - s_1^2 + Is_1\sqrt{(s_1^2+s_2^2+s_3^2)} + s_2s_3}, \tag{70}$$

$$\alpha_{12} = \frac{(-1-I)(Is_1s_2 + s_3\sqrt{(s_1^2+s_2^2+s_3^2)})}{-Is_1s_2 - s_3\sqrt{(s_1^2+s_2^2+s_3^2)} + s_1^2 + s_3^2 + s_1\sqrt{(s_1^2+s_2^2+s_3^2)} - Is_2s_3}, \tag{71}$$

$$\alpha_{22} = \frac{I(Is_1s_2 + s_3\sqrt{(s_1^2+s_2^2+s_3^2)}) + Is_1\sqrt{(s_1^2+s_2^2+s_3^2)} + s_2s_3 + Is_1^2 + Is_3^2)}{-Is_1s_2 - s_3\sqrt{(s_1^2+s_2^2+s_3^2)} + s_1^2 + s_3^2 + s_1\sqrt{(s_1^2+s_2^2+s_3^2)} - Is_2s_3}. \tag{72}$$

Note that the intrinsic parameters are totally canceled out. We have then Eqs. (69)–(72) which allow us to compute the actual values of $\alpha_{11}, \alpha_{21}, \alpha_{12}, \alpha_{22}$. Now replacing Eq. (59) of τ in the entries of α_1 and α_2 given by Eq. (57), we obtain the equations for $\alpha_{11}, \alpha_{21}, \alpha_{12}$ and α_{22} depending of the intrinsic parameters $k_{i,j}$. Equalizing these equations with their real values obtained previously, we gain four quadratic equations which depend only of four unknown intrinsic parameters. Thus, we should find another set of equations to solve the problem. The way to do that is simply considering the second camera with its epipole e_{21}. Since we are assuming that the intrinsic parameters remain constant we can consequently gain a second set of four equations depending again of the four intrinsic parameters. Both set of equations help to find the four intrinsic parameters. Since this is a system of quadratic equations we resort to an iterative procedure for finding the solution. First, we tried the Newton–Raphson [27] and the "Continuation method" [9]. These methods were not practicable enough due to their complexity. We used instead a variable in size window minima search which through the computation ensure the reduction of the quadratic error. This simple approach worked fast and reliable. The procedure can be summarized in the following steps:

(1) Use 8 or more point correspondences between two cameras and a controlled rotation about only one axis and a 3-D translation between the cameras.
(2) Calculate the actual values of the homogeneous $\boldsymbol{\alpha}_i$ by using the known camera motion and Eqs. (69)–(72).
(3) Calculate $K^{\mathrm{T}}\mathbf{l}_c = K^{\mathrm{T}}\mathbf{e}_{12} \times [1, \tau, 0]^{\mathrm{T}}$ with the epipole in the first camera, evaluated from the point correspondences. Note that in K^{T} are the unknown intrinsic parameter. To fulfill Pascal's theorem solve the equations system to get τ (Eq. (59)).
(4) Replace τ in Eq. (57) and calculate the homogeneous representation of these intersection points to get quadratic polynomials which depends on the four unknown intrinsic parameters. Note that the intrinsic parameters are not cancelled out because of the insert of the real values from the epipole. Because of the invariant properties of the $\boldsymbol{\alpha}$'s the polynomials must be equal to the evaluated values of the $\boldsymbol{\alpha}$'s in step 2. This leads to four quadratic equations.
(5) Proceed exactly as steps $2 \rightarrow 4$ using the epipole $\mathbf{e}_{21}$ in the second camera to obtain a new set of four equations depending of the intrinsic parameters.
(6) Using the iterative procedure explained above find K using the eight equations.

The interesting aspect here is that we require only two views (one-camera motion) to find a solvable equation system. Other methods gain for each camera motion only a couple of equations, thus they require at least three camera views (two-camera motions) to solve the problem [9,22].

6.2. Computing non-stationary intrinsic parameters

In this case, we will consider that due to the camera motion, the intrinsic parameters may have been changed. The procedure can be formulated along the same previous ideas with the difference of that we compute the line

l_c using the fundamental matrix and a point lying at line at infinite of the second camera as follows:

$$\mathbf{l}_c = \begin{pmatrix} p'_1 \\ p'_2 \\ p'_3 \end{pmatrix} \times \begin{pmatrix} 1 \\ \tau' \\ 0 \end{pmatrix} \sim F \begin{pmatrix} 1 \\ \tau' \\ 0 \end{pmatrix}. \tag{73}$$

The authors [9] used the same idea for computing $\mathbf{l}_c$. Now, these equations can be expressed as follows:

$$\sim F \begin{pmatrix} 1 \\ \tau' \\ 0 \end{pmatrix} = \begin{pmatrix} a & b & x \\ c & d & y \\ n & m & z \end{pmatrix} \begin{pmatrix} 1 \\ \tau' \\ 0 \end{pmatrix} = \begin{pmatrix} a + b\tau' \\ c + d\tau' \\ n + m\tau' \end{pmatrix} \sim \begin{pmatrix} \frac{a + b\tau'}{c + d\tau'} \\ 1 \\ \frac{n + m\tau'}{c + d\tau'} \end{pmatrix}. \tag{74}$$

Note that we get the mapping $\tau = (a + b\tau')/(c + d\tau')$ in the second camera.

Now similar as in the previous case, we will use an example for facilitating the understanding. We will use the same camera motion given in Eq. (56). The fundamental matrix in terms of the intrinsic parameters K of the first camera, K' of the second one and the camera motion reads

$$F = K^{-1^{\mathrm{T}}}[t]_\times RK'^{-1} = \begin{pmatrix} -3\frac{k'_{22}k_{22}}{v_2} & 0 & -\frac{(k'_{11} - 3k'_{13})k_{22}k'_{22}}{v_2} \\ 0 & -3\frac{k'_{11}k_{11}}{v_2} & -\frac{k_{11}k'_{11}(2k'_{22} - 3k'_{23})}{v_2} \\ \frac{(2k_{11} + 3k_{13})k_{22}k'_{22}}{v_2} & -\frac{(k_{22} - 3k_{23})k_{11}k'_{11}}{v_2} & 1 \end{pmatrix}, \tag{75}$$

where $v_2 = -3k'_{22}k_{22}k_{13}k'_{13} + k_{22}k'_{22}k'_{11}k_{13} + k_{22}k'_{23}k'_{11}k_{11} - 2k_{22}k'_{22}k'_{13}k_{11} + 2k_{23}k'_{22}k'_{11}k_{11} - 3k_{23}k'_{23}k'_{11}k_{11}$.

Note that the value of the line $\mathbf{l}_c = [p'_1, p'_2, p'_3]^{\mathrm{T}} \times [1, \tau', 0]^{\mathrm{T}}$ is now computed in terms of the fundamental matrix, i.e., $\mathbf{l}_c = F^{\mathrm{T}}[1, \tau', 0]^{\mathrm{T}}$, where F now depends of K (first camera) and K' (second one). Similar as above we compute the $\boldsymbol{\alpha}$'s and according the Pascal's theorem we gain a polynomial similar as Eq. (58). This reads

$$\begin{aligned}
&(-4ik_{22}^2k_{11}^2(13k'^2_{22} + 10k'^2_{11}\tau'^2 - 4k'_{22}k'_{11}\tau'))/ \\
&((3k'_{22}k_{22}k_{13}k'_{13} + k_{22}k'_{22}k'_{11}k_{13} + k_{22}k'_{23}k'_{11}k_{11} \\
&\quad - 2k_{22}k'_{22}k'_{13}k_{11} + 2k_{23}k'_{22}k'_{11}k_{11} \\
&\quad - 3k_{23}k'_{23}k'_{11}k_{11})^2) = 0.
\end{aligned} \tag{76}$$

We select one of both solutions of this second-order polynomial

$$\tau' = \frac{4k'_{22}k'_{11} + 6Ik'_{22}k'_{11}\sqrt{14}}{20(k'^2_{11})} \tag{77}$$

and substitute in the homogeneous coordinates of the $\boldsymbol{\alpha}$'s using $k_{12} = k'_{12} = 0$ we get

$$\alpha_{11} = -\frac{i(-5i - 4 + i\sqrt{14})}{5i + 2 + 2i\sqrt{14}}, \tag{78}$$

$$\alpha_{21} = \frac{-2 + 3i\sqrt{14}}{5i + 2 + 2i\sqrt{14}}, \tag{79}$$

$$\alpha_{12} = \frac{10 - 10i}{-4i - 2 + 3i\sqrt{14} - \sqrt{14}}, \tag{80}$$

$$\alpha_{22} = -\frac{8 + 6i - \sqrt{14} + 3i\sqrt{14}}{-4i - 2 + 3i\sqrt{14} - \sqrt{14}}, \tag{81}$$

where

$$\boldsymbol{\alpha}_3 = \begin{pmatrix} 1 - i \\ 1 - i \\ 2 \end{pmatrix}$$

remains fully independent of the intrinsic parameters. According to these results, the calibration procedure for computing non-stationary intrinsic parameters comprises of the following steps:

(1) Use 8 or more image point correspondences and a rotation about one axis and the 3-D translation.
(2) Compute the invariants $\boldsymbol{\alpha}'_i$ using Eqs. (85)–(88).
(3) Compute the F matrix according to Ref. [24].
(4) Considering

$$\begin{pmatrix} k_{11} & 0 & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} k'_{11} & 0 & k'_{13} \\ 0 & k'_{22} & k'_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

for the first and second camera, respectively, in equation $F = K^{-\mathrm{T}}EK'^{-1}$ and instead of $\mathbf{l}_c = \mathbf{e}_{12} \times [1, \tau, 0]$ we use $\mathbf{l}_c = F(1, \tau', 0)^{\mathrm{T}}$(parameterized point at infinity of the second camera), we can solve Eq. (76) for τ'. Then substitute τ' in the equations for α'_{ij} which are equalized with the values of the invariants α'_{ij}.
(5) Proceed exactly as steps $2 \rightarrow 4$ however in the second image using instead of $\mathbf{l}'_c = \mathbf{e}_{21} \times [1, \tau', o]$ we use $\mathbf{l}'_c = F^{\mathrm{T}}(1, \tau, 0)^{\mathrm{T}}$ to get similar result as Eq. (76) for τ. Then substitute τ in the equations for α'_{ij} which

are equalized with the values of the invariants α'_{ij}. In this way, we have gain another four quadratic equations depending only of the k_{ij}.

(6) Solve eight equations to get K.

6.3. Decoupling the essential matrix of the intrinsic camera parameters

If we consider now general motion

$$[R\,|\,t] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \tag{82}$$

the fundamental matrix reads

$$F = K^{-\mathrm{T}} E K'^{-1} \tag{83}$$

$$= \begin{pmatrix} k_{11} & 0 & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{pmatrix}^{-\mathrm{T}} \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{pmatrix} \times \begin{pmatrix} k'_{11} & 0 & k'_{13} \\ 0 & k'_{22} & k'_{23} \\ 0 & 0 & 1 \end{pmatrix}^{-1}. \tag{84}$$

Using this equation, the rotated conic points given by Eq. (31) and the tangent line $\mathbf{l}_c$ as Eq. (73), we can compute by means of Eq. (53), the coordinates of the $\boldsymbol{\alpha}$'s fully independent of the intrinsic parameters

$$\begin{aligned} \alpha_{11} = {} & i(iE_{11}E_{22}^2 + iE_{11}E_{32}^2 - iE_{12}E_{21}E_{22} - iE_{12}E_{31}E_{32} \\ & - iE_{12}\sqrt{v_3} + E_{21}E_{12}^2 + E_{21}E_{32}^2 - E_{22}E_{11}E_{12} \\ & - E_{22}E_{31}E_{32} - E_{22}\sqrt{v_3} - E_{31}E_{12}^2 - E_{31}E_{22}^2 \\ & + E_{32}E_{11}E_{12} + E_{32}E_{21}E_{22} + E_{32}\sqrt{v_3})/ \\ & (iE_{11}E_{22}^2 + iE_{11}E_{32}^2 - iE_{12}E_{21}E_{22} - iE_{12}E_{31}E_{32} \\ & - iE_{12}\sqrt{v_3} - E_{31}E_{12}^2 - E_{31}E_{22}^2 + E_{32}E_{11}E_{12} \\ & + E_{32}E_{21}E_{22} + E_{32}\sqrt{v_3} - E_{21}E_{12}^2 - E_{21}E_{32}^2 \\ & + E_{22}E_{11}E_{12} + E_{22}E_{31}E_{32} + E_{22}\sqrt{v_3}), \end{aligned} \tag{85}$$

$$\begin{aligned} \alpha_{12} = {} & 2(-E_{21}E_{12}^2 - E_{21}E_{32}^2 + E_{22}E_{11}E_{12} \\ & + E_{22}E_{31}E_{32} + E_{22}\sqrt{v_3})/(iE_{11}E_{22}^2 + iE_{11}E_{32}^2 \\ & - iE_{12}E_{21}E_{22} - iE_{12}E_{31}E_{32} - iE_{12}\sqrt{v_3} \\ & - E_{31}E_{12}^2 - E_{31}E_{22}^2 + E_{32}E_{11}E_{12} \\ & + E_{32}E_{21}E_{22} + E_{32}\sqrt{v_3} - E_{21}E_{12}^2 - E_{21}E_{32}^2 \\ & + E_{22}E_{11}E_{12} + E_{22}E_{31}E_{32} + E_{22}\sqrt{v_3}), \end{aligned} \tag{86}$$

$$\begin{aligned} \alpha_{21} = {} & (1-i)(E_{11}E_{22}^2 + E_{11}E_{32}^2 - E_{12}E_{21}E_{22} \\ & - E_{12}E_{31}E_{32} - E_{12}\sqrt{v_3})/(-iE_{11}E_{22}^2 \\ & - iE_{11}E_{32}^2 + iE_{12}E_{21}E_{22} + iE_{12}E_{31}E_{32} \\ & + iE_{12}\sqrt{v_3} - E_{21}E_{12}^2 - E_{21}E_{32}^2 + E_{22}E_{11}E_{12} \\ & + E_{22}E_{31}E_{32} + E_{22}\sqrt{v_3} - iE_{31}E_{12}^2 - iE_{31}E_{22}^2 \\ & + iE_{32}E_{11}E_{12} + iE_{32}E_{21}E_{22} + iE_{32}\sqrt{v_3}), \end{aligned} \tag{87}$$

$$\begin{aligned} \alpha_{22} = {} & -(E_{11}E_{22}^2 + E_{11}E_{32}^2 - E_{12}E_{21}E_{22} \\ & - E_{12}E_{31}E_{32} - E_{12}\sqrt{v_3} - iE_{31}E_{12}^2 - iE_{31}E_{22}^2 \\ & + iE_{32}E_{11}E_{12} + iE_{32}E_{21}E_{22} + iE_{32}\sqrt{v_3} \\ & - E_{21}E_{12}^2 - E_{21}E_{32}^2 + E_{22}E_{11}E_{12} + E_{22}E_{31}E_{32} \\ & + E_{22}\sqrt{v_3})/(-iE_{11}E_{22}^2 - iE_{11}E_{32}^2 + iE_{12}E_{21}E_{22} \\ & + iE_{12}E_{31}E_{32} + iE_{12}\sqrt{v_3} - E_{21}E_{12}^2 - E_{21}E_{32}^2 \\ & + E_{22}E_{11}E_{12} + E_{22}E_{31}E_{32} + E_{22}\sqrt{v_3} \\ & - iE_{31}E_{12}^2 - iE_{31}E_{22}^2 + iE_{32}E_{11}E_{12} \\ & + iE_{32}E_{21}E_{22} + iE_{32}\sqrt{v_3}), \end{aligned} \tag{88}$$

where

$$\begin{aligned} v_3 = {} & 2E_{11}E_{12}E_{21}E_{22} + 2E_{11}E_{12}E_{31}E_{32} \\ & + 2E_{21}E_{22}E_{31}E_{32} - E_{12}^2E_{31}^2 - E_{12}^2E_{21}^2 \\ & - E_{22}^2E_{31}^2 - E_{22}^2E_{11}^2 - E_{32}^2E_{21}^2 - E_{32}^2E_{11}^2. \end{aligned} \tag{89}$$

Other authors use other way to separate the essential matrix [9], after they have computed F and the intrinsic camera parameters K, they use a direct factorization

$$E = [t]_x R = K^{\mathrm{T}} F K. \tag{90}$$

The computation of the rotation R and translation t from E is a classical problem [25,26].

Finally, we will give some critical remarks about our procedure comparing with a standard method. If we know at least a rotation about one axis and a 3-D translation and the eight or more correspondences, the Pascal's method can compute K. Using the expression $E = K^{\mathrm{T}}FK$ and at least a rotation about one axis and a 3-D translation and eight or more correspondences, we could not compute the intrinsic parameters. That is because we got quadratic expressions of the k_{ij} terms. In the case of non-stationer intrinsic parameters $E = K^{\mathrm{T}}FK'$, similarly we could not get sensible results due to the quadratic expressions in the terms of k_{ij}.

7. Experimental analysis

In this section, we test the Pascal-theorem-based method using simulated and real images. We explore

Table 1
Intrinsic parameters dependency of the type of camera motion

Type	1. Motion	2. Motion	3. Motion	k_{ij}	Min. num.
1	$R_y, t_y = 0$	$R_y, t_y = 0$	$R_y, t_y = 0$	k_{11}, k_{13}, —, k_{23}	
2	$R_y, t_x = 0$	$R_y, t_x = 0$	$R_y, t_x = 0$	—, k_{13}, k_{22}, k_{23}	
3	$R_x, t_x = 0$	$R_x, t_x = 0$	$R_x, t_x = 0$	—, k_{13}, k_{22}, k_{23}	
4	$R_xR_y, t_y = 0$	$R_xR_y, t_y = 0$	$R_xR_y, t_y = 0$	—, k_{13}, k_{22}, k_{23}	
5	$R_x, R_y, R_z, t_z = 0$			— , — , — , —	
6	R_x			$k_{11}, k_{13}, k_{22}, k_{23}$	1
7	R_y			$k_{11}, k_{13}, k_{22}, k_{23}$	1
8	R_z			$k_{11}, k_{13}, k_{22}, k_{23}$	1
9	R_xR_y			$k_{11}, k_{13}, k_{22}, k_{23}$	1
10	R_xR_z			$k_{11}, k_{13}, k_{22}, k_{23}$	1
11	R_yR_z			$k_{11}, k_{13}, k_{22}, k_{23}$	1
12	$R_zR_yR_x$			$k_{11}, k_{13}, k_{22}, k_{23}$	1
13	$R_z, t_x = 0$	$R_z, t_y = 0$		$k_{11}, k_{13}, k_{22}, k_{23}$	2
14	$R_x, t_x = 0$	$R_x, t_y = 0$		$k_{11}, k_{13}, k_{22}, k_{23}$	2
15	$R_y, t_x = 0$	$R_y, t_y = 0$		$k_{11}, k_{13}, k_{22}, k_{23}$	2
16	$R_xR_y, t_y = 0$	$R_xR_y, t_x = 0$		$k_{11}, k_{13}, k_{22}, k_{23}$	2

firstly the effect of different kinds of camera motion in the computing of the intrinsic camera parameters and then the accuracy of the computation by increasing noise.

7.1. Experiments with simulated images

Using a Maple simulation we tested the Pascal's-theorem-based method to explore the dependency of the type and the amount of necessary camera motions for solving the problem and the noise sensitivity of the method. The experiments showed that at least a rotation about only one axis and displacement along the three axes are necessary for the stabile computation of all intrinsic parameters.

7.1.1. The role of the type of camera motion for the computation of the intrinsic parameters

We generated various types of motion varying the rotation and the translation. In Table 1, the subindex of the rotation matrices indicates that the rotations are about the x-, y- or z-axis and translations along the axes are denominated by t_x, t_y and t_z . When $t_x = 0$ or $t_y = 0$, we cannot solve the problem as in cases 1–4. The case 5 shows that when the translation along the z-axis is zero the camera epipoles lye at infinite making impossible the computation.

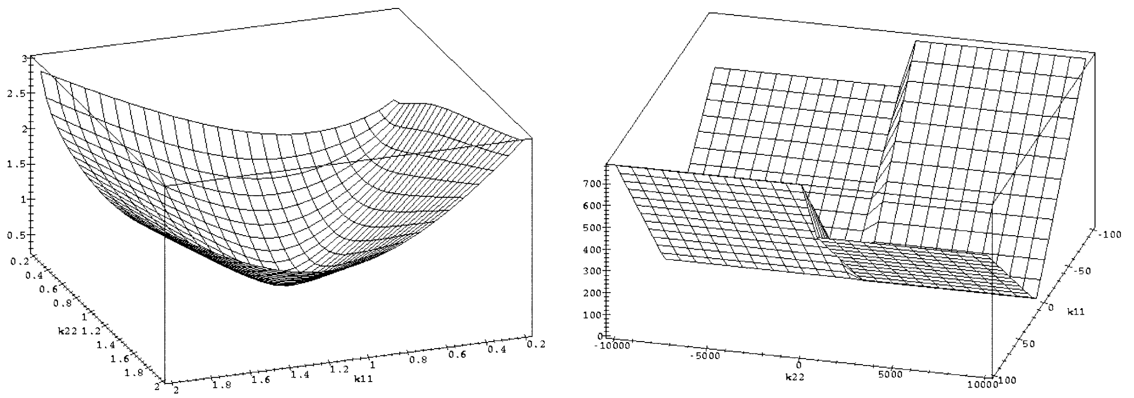
The experiments show that at least a rotation about one axis and displacement along the three axes are necessary for the computation of all parameters as in cases 6–8. We see also that for cases of 9–12 rotations about other axes leads to same result. Cases 13–16 simply corresponds to separated axis translations, i.e., two-camera motions are needed. When all of the intrinsic parameter can be computed like in for cases 6–16 we have a clear minima in the quadratic error function as presented in Fig. 6a. However, when it is indefinite like in cases 1–4, we do not have a clear minima, the value lies indefinite along a minima line (see Fig. 6b).

These experiments indicate that we can use controlled motion to simplify the estimation of the motion necessary to be known for computing the intrinsic parameters.

7.1.2. Noise sensitivity analysis

In order to test the performance of our approach using a Maple simulation, we carried out a similar motions given in items 7 and 12 of Table 1. The coordinates of the points were corrupted with zero mean Gaussian noise with a standard deviation multiple of a pixel in a range 0 –3.0 pixel. In the simulation, 12 point correspondences were used. For the tests, we used exact arithmetic of the Maple program instead of floating point arithmetic of the C language. For the first experiment, the camera was firstly rotated about the y-axis and it had a small translation along the three axes. The true intern camera parameters were $k_{11} = k_{22} = 500$ and $k_{13} = k_{23} = 256$. We used an image 512×512 pixels.

Table 2 shows the computed intrinsic parameters. The extreme right column of the table shows the error obtained substituting these parameters in the polynomial (76) which gives zero for the case of zero noise. The values in this column show that by increasing noise the computed intrinsic parameters have a tiny deviation from the ideal value of zero. This indicates that the procedure is relatively stable against noise. We could imagine that there is a relative flat surface around the global minimum of the polynomial. Note that there are discontinuities shown by noise 1.25. However, as opposite to other methods [9] which cannot compute these values, our



(a) (b)

Fig. 6. (a) Minima in the quadratic error function by computable parameters. (b) Absence of a minima by undefined intrinsic parameter.

Table 2
Intrinsic parameters by rotation about the y-axis, and translation along the three axes by increasing noise

Noise	k_{11}	k_{13}	k_{22}	k_{23}	Error
0	500	256	500	256	10^{-8}
0.1	505	259	509	261	0.001440
0.5	504	259.5	503.5	258	0.004897
0.75	498	254	503.5	258	0.001668
1	482	242	485	254	0.011517
1.25	473	220	440	238	0.031206
1.5	517	272	518	266	0.015
2.0	508	262.5	504	258.5	0.006114
2.5	515	268	501.9	257	0.011393
3	510	265	524	276	0.011440

Table 3
Intrinsic parameters by rotation about the three axes, and translation along the three axes by increasing noise

Noise	k_{11}	k_{13}	k_{22}	k_{23}	Error
0	500	256	500	256	10^{-8}
0.1	500	255.5	501	258	0.000659
0.5	498	254	499	255	0.001031
0.75	496	252	505	261	0.004013
1	508	263	508.5	266	0.004656
1.25	494	250	514	272	0.010244
1.5	502	255	488	240	0.007613
2.0	524	276	487	242	0.017349
2.5	490	252	540	334	0.025362
3	502	258	522	284	0.013075

method can anyway compute the intrinsic parameters. For the second experiment, the camera was firstly rotated about the three axes and it had a small translation along the three axes. The results are presented in Table 3. The accuracy of Tables 2 and 3 is tight related with that of the point correspondence. In Ref. [9], similar experiments are presented and comparing with our results, we can conclude that the procedures using the Kruppa's equations are slightly more noise sensitive as the Pascal's-theorem-based method. Note that our experiments presents computations with noise varying in a range beyond 1 pixel and show that the error is below 0.0254. As opposed to Ref. [9], our method has proved that for a range more than 1 pixel it has a steady behavior and always is able to compute the intrinsic parameters.

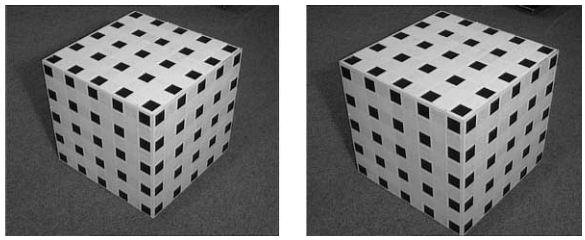


Fig. 7. Scenario.

7.2. Experiments with real images

In this section, we present experiments using real images with one general camera motion (see Fig. 7). The motion was done about the three coordinate axes. We use a calibration dice and for comparison purposes we compute the intrinsic parameters from the involved projective matrices by splitting the intrinsic parameters from the extrinsic parameters. These reference values were: first camera $k_{11} = 1200.66$, $k_{22} = 1154.77$, $k_{13} = 424.49$, $k_{23} = 264.389$ and second camera $k_{11} = 1187.82$, $k_{22} = 1141.58$, $k_{13} = 386.797$, $k_{23} = 288.492$ with mean errors of 0.688 and 0.494, respectively. Thereafter using

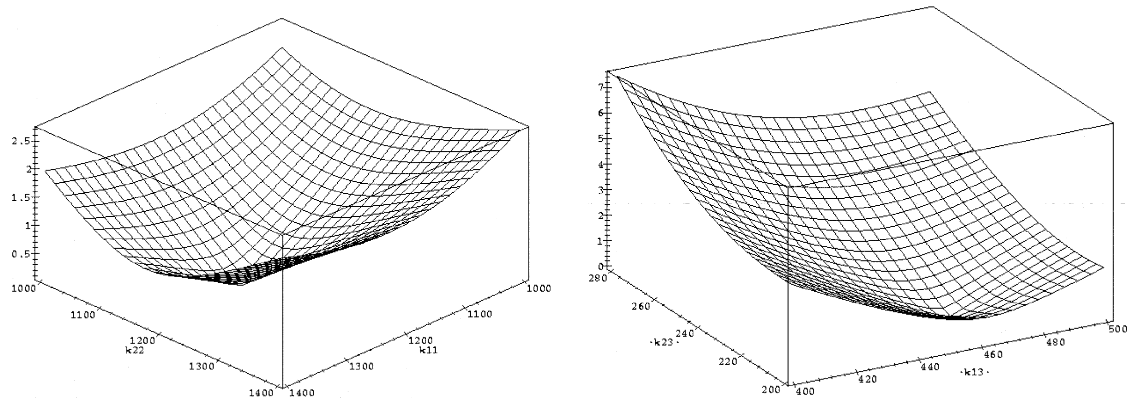


Fig. 8. Minima for k_{11} and k_{13} in their quadratic error function.

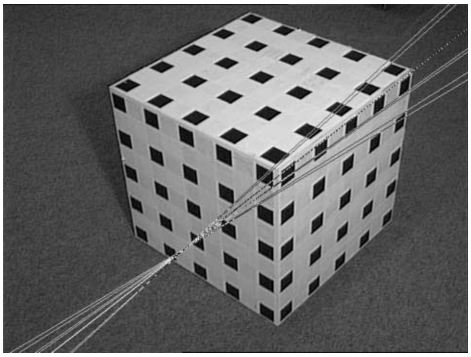


Fig. 9. Superimposed epipolar lines using the reference and the Pascal's method.

the gained $[R_1 \mid t_1]$ and $[R_2 \mid t_2]$ we compute the $[R \mid t]$ between cameras which is required for the Pascal's-theorem-based method. The fundamental matrices were computed using a non-linear method. Using the Pascal's-theorem-based method, we compute the following intrinsic parameters $k_{11} = 1244$, $k_{22} = 1167$, $k_{13} = 462$ and $k_{23} = 217$, see the minimum values in Fig. 8. These values resemble quite well to the reference ones and cause an error of $\sqrt{|eqn_1|^2 + \cdots + |eqn_8|^2}$: 0.00496045 in the error function. This square error is computed using the eight equations computed from the α's of the first and second cameras. The difference with the reference values is attributable to inherent noise in the computation and to the fact that the reference values are also not exact.

In order to visualize how good we gain the epipolar geometry, we superimposed the epipolar lines for some points using the reference method and the Pascal's method. In both, we computed the fundamental matrix in terms of their intrinsic parameters, i.e., $F = K^{-T}([t]_x R)K^{-1}$. Fig. 9 shows this comparison. It is clear that both methods give quite similar epipolar lines and interesting enough as the figure shows, the intersecting point or epipole coincides almost exact.

8. Conclusions

This paper presents a geometric approach to compute the camera intrinsic parameters in the geometric algebra framework using the Pascal's theorem. We adopted the projected characteristics of the absolute conic in terms of the Pascal's theorem to propose an entirely new camera calibration method based on purely geometric thoughts. The use of this theorem in the geometric algebra framework allows us the computing of a projective invariant using the conics of only two images. Then, this projective invariant expressed in terms of brackets helps us to set enough equations to solve the calibration problem. Our method requires to know the point correspondences, the camera translation direction and a known rotation about only one axis, not the complete essential matrix. The method throws new light for the understanding of the problem owing to the application of the Pascal's theorem and it also explains the over-seen role of the projective invariant in terms of the brackets. Using synthetic and real images, we show that the method performs efficiently without any initialization or getting trapped in local minima.

Acknowledgements

Eduardo Bayro Corrochano was supported by the Project SO-201 of the Deutsche Forschungsgemeinschaft.

References

[1] R.I. Hartley, An algorithm for self-calibration from several views, Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994, pp. 908–912.
[2] B. Triggs, Autocalibration and the absolute quadric, IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 609–614.
[3] M. Pollefeys, R. Khoch, L. Van Gool, Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, Proceedings of the Sixth International Conference on Computer Vision, 1998, pp. 90–95.
[4] L. Dron, Dynamic camera self-calibration from controlled motion sequences, IEEE Conference on Computer Vision and Pattern Recognition, New York, June 15–18, 1993, pp. 501–506.
[5] A. Basu, Active calibration: alternative strategy and analysis, IEEE Conference on Computer Vision and Pattern Recognition, New York, June 15–18, 1993, pp. 495–500.
[6] F. Du, M. Brady, Self-calibration of the intrinsic parameters of cameras for active vision systems, IEEE Conference on Computer Vision and Pattern Recognition, New York, June 15–18, 1993, pp. 477– 482.
[7] K. Daniilidis, J. Ernst, Active intrinsic calibration using vanishing points, Pattern Recognition Lett. 17 (1996) 1179–1189.
[8] M. Armstrong, A. Zisserman, R.I. Hartley, Self-calibration from image triples, Proceedings of the Fourth European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 1064, 1996, pp. 3–16.
[9] Q.T. Luong, O.D. Faugeras, Self-calibration of a moving camera from point correspondences and fundamental matrices, Int. J. Comput. Vision 22 (3) (1997) 261–289.
[10] H. Grassmann, Der ort der Hamilton'schen Quaternionen in der Ausdehnungslehre, Math. Ann. 12 (1877) 375.
[11] W.K. Clifford, Applications of Grassmann's extensive algebra, Am. J. Math. 1 (1878) 350–358.
[12] D. Hestenes, G. Sobczyk, Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics, D. Reidel, Dordrecht, 1984.
[13] D. Hestenes, R. Ziegler, Projective geometry with Clifford algebra, Acta Appl. Math. 23 (1991) 25–63.
[14] E. Bayro-Corrochano, J. Lasenby, G. Sommer, Geometric algebra: a framework for computing point and line correspondences and projective structure using n uncalibrated cameras, IEEE Proceedings of ICPR'96, Vienna, Austria, Vol. I, August, 1996, pp. 334–338.
[15] E. Bayro-Corrochano, J. Lasenby, Object modelling and motion analysis using Clifford algebra, in: Roger Mohr, Wu Chengke, Xi'an (Eds.), Proceedings of Europe-China Workshop on Geometric Modeling and Invariants for Computer Vision, China, April, 1995, pp. 143–149.
[16] E. Bayro-Corrochano, J. Lasenby, Geometric techniques for the computation of projective invariants using n uncalibrated cameras, Proceedings of the Indian Conference on Computer Vision and Image Processing, New Delhi, India, 21–23 December, 1998, pp. 95–100.
[17] J. Lasenby, A.N. Lasenby, C.J.L. Doran, W.J. Fitzgerald, New geometric methods for computer vision — an application to structure and motion estimation, Int. J. Comput. Vision 26 (3) (1998) 191–213.
[18] E. Bayro-Corrochano, V. Banarer, Geometric analysis and computing of projective structure and motion, Proceedings of the 10th International Conference on Image Analysis and Processing, Venice, Italy, 27–29 September 1999, pp. 291–296.
[19] J.G. Semple, G.T. Kneebone, Algebraic Projective Geometry, Oxford University Press, Oxford (reprinted by Oxford Science Publications) 1985.
[20] J.L. Mundy, A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA, USA, 1992.
[21] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, 2000.
[22] S.J. Maybank, O.D. Faugeras, A theory of self-calibration of a moving camera, Int. J. Comput. Vision 8 (2) (1992) 123–151.
[23] J. Laseby, E. Bayro-Corrochano, Analysis and computation of projective invariants from multiple views in the geometric algebra framework (special issue), Invariants Pattern Recognition Classification. (IJPRAI) 13 (8) (1999) 1105–1121.
[24] Q.T. Luong, O.D. Faugeras, An optimization framework for efficient self-calibration and motion determination. Proceedings of the Conference on Computer Vision and Pattern Recognition, Jerusalem, Israel, Vol. A, 1994, pp. 248–252.
[25] H.C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature 293 (1981) 133–138.
[26] R.I. Hartley, Estimation of relative camera position for uncalibrated cameras, Proceedings of the European Conference on Computer Vision, ECCV'92, 1992, pp. 579–587.
[27] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1994.

About the Author—EDUARDO JOSÉ BAYRO-CORROCHANO gained his Ph.D. in Cognitive Computer Science in 1993 from the University of Wales at Cardiff. From 1995 to 1999 he has been Researcher and Lecturer at the Institute for Computer Science, Christian Albrechts University, Kiel, Germany, working on the applications of Clifford algebra to cognitive systems.

His current research interest focuses on geometric methods for artificial perception and action systems. It includes geometric neural networks, visually guided robotics, color image processing, Lie bivector algebras for early vision and robot maneuvering. He is contributor, editor and author of books on applications of Clifford geometric algebra in computer science and engineering.

About the Author—BODO ROSENHAN gained his M.S. degree in Computer Science in 1998. Since 1999 he has been pursuing his Ph.D. at the Institute for Computer Science, Christian Albrechts University, Kiel, Germany, working on the applications of Clifford geometric algebra in computer vision.

His research interests involve Kalman filter techniques, 3-D motion estimation, and visual guided robotics.