Journal of Mathematical Imaging and Vision 13, 79–100, 2000

Motor Algebra for 3D Kinematics: The Case of the Hand-Eye Calibration

EDUARDO BAYRO-CORROCHANO
Centro de Investigacion en Matemáticas, Apartado Postal 402, 36000-Guanajuato, Gto., Mexico
edb@fractal.cimat.mx

KOSTAS DANIILIDIS
GRASP Laboratory, Pennsylvania University
kd@penn.grasp.edu

GERALD SOMMER
Computer Science Institute, Christian Albrechts University, Preusserstrasse 1-9, 24105 Kiel, Germany
gs@informatik.uni-kiel.de

Abstract. In this paper we apply the Clifford geometric algebra for solving problems of visually guided robotics. In particular, using the algebra of motors we model the 3D rigid motion transformation of points, lines and planes useful for computer vision and robotics. The effectiveness of the Clifford algebra representation is illustrated by the example of the hand-eye calibration. It is shown that the problem of the hand-eye calibration is equivalent to the estimation of motion of lines. The authors developed a new linear algorithm which estimates simultaneously translation and rotation as components of rigid motion.

Keywords: computer vision, kinematics, visual robotics, Clifford algebra, geometric algebra, rotors, motors, screws, hand-eye calibration

1. Introduction

The goal of this paper is basically to set up the necessary mathematics for dealing with the 3D kinematics useful as a general frame for the fields of computer vision and robotics. For that the authors choose the algebra of motors in the geometric algebra framework.

In the literature we can find various ways of representing the 3D kinematics. They will be briefly reviewed emphasizing the screw transformations which we follow in this paper. The foundations of the screw theory can be traced back to the contributions of Chasles and Poinsot in the early 1830s, see e.g. [2], as well as the dual quaternions introduced by Clifford in his seminal paper *Preliminary sketch* of *bi-quaternions* [9] and later on the work of Study [38] who utilized the dual numbers to represent the relative position of two skew lines in space. In an amenable article Rooney [34] compared the different representations of the general spatial screw. The twist or infinitesimal generator of the Euclidean group is the Lie algebra matrix approach to describe rigid 3D motions, see Murray et al. [30]. It is worth to mention the work of Chevalier [7] who presented a geometrical formulation of the dual quaternions in the Lie algebra framework.

In the area of robotics for the treatment of the manipulator kinematics Gu and Luh [14] used the dual number transformations and Pennock and Yang employing dual matrices [31] presented closed-form solutions for various types of robot manipulators. MacCarthy [29] analyzed multilinks and similar as Gu and Luh [14] he computed the dual form of the Jacobian of a manipulator again using dual orthogonal matrices. Funda and Paul [13] carried out a computational analysis of screw transformations in robotics. They showed that the dual quaternions represent simultaneously rotation and translation transformations for dealing with the kinematics of robot chains more efficiently than any other approach. Kim and Kumar [23] using the dual quaternion formalism as a line transformation operator solved the inverse kinematics of a 6 degree of freedom robot manipulator. Aspragathos and Dimitros [1] confirmed that the homogeneous transformation is the approach commonly used in robotics tasks and the approaches of dual quaternions and Lie algebra were overseen so far, although their advantages with respect to the reduction of the number of representation parameters result in practical benefits.

In the field of computer vision we can find similar representation formalisms in various types of applications like motion estimation, pose and 3D structure recognition. In most of the methods the rotation and translation transformations were represented separately using either matrices or quaternions, see the survey of Sabata and Aggarwal [35]. The disadvantage of representing separately these components is that for solving the problems nonlinear methods are often required. In the case of the so called hand-eye calibration problem which we treat in this paper as an example for the application of the motor algebra several authors considered for the computation the rotation axis and angle [36, 39], the use of quaternions [8] and a canonical matrix representation [25]. Chen [6] using the matrix screw theory found the key invariant of the screw between two 3D axes which means the rotation angle and the translation along the screw axis remain constant. In other applications the authors applied successfully dual quaternions like Walker et al. [41] for estimating 3D location, and twists and exponential maps like Bregler and Malik [5] for tracking the kinematic chains of moving objects or persons.

Summarizing, in these mathematical approaches so far we can clearly identify two key aspects the use of dual numbers and the representation of screw transformations in terms of matrices or quaternions. In regard to these aspects we should choose an algebraic system that on the one hand allows representations in terms of dual numbers and on the other hand offers an alternative to the matrix representations which often have redundant entries.

In this paper we present an isomorphic approach to screws called motors [5, 9] which offers computational advantages with respect to dual quaternions. These are based on the fact that motors represent spinors which can be defined in spaces of any dimension and in the case of kinematic chains they obey the group properties. The term motor was introduced by Clifford [9], but he died before he could show us its embedding in the special Clifford algebra. It is also important to mention that the engineering literature does not show a substantial progress so far in the use of line geometry for computer vision and robotics. In this paper we stress the role of the line or bivector algebra for the modelling of the motion of points, lines and planes. This can be certainly used as a common framework advantageous in a variety of computer vision and kinematics tasks.

The particular case of study to illustrate our mathematical approach is the hand-eye calibration, simplified as a problem of motion of lines. In the above mentioned publications many approaches came up with a decoupled computation of the motion parameters of the hand-eye calibration problem. Exploring this problem in the geometric algebra framework we end up with an approach that simultaneously computes translation and rotation without resorting to nonlinear minimization algorithms. This application illustrates the important case that some nonlinear problems treated in higher dimensional geometric algebras become linear ones. The virtue of computing in the geometric algebra framework is thus strongly confirmed by the hand-eye calibration problem.

The paper is organized as follows. In section two an outline of the geometric algebra is given. The third section presents the Euclidean 3D geometric algebra, rotors and their properties and the modelling of the 3D motion of points, lines and planes using this algebraic system. The fourth section introduces the 4D motor algebra and shows its use for the modelling of the 3D motion of points, lines and planes. Section five describes the hand-eye problem and its solution. We formulate the solution of the hand-eye calibration in the geometric algebra framework. Then it is shown that this formulation leads to the first hand-eye calibration algorithm that simultaneously estimates the translation and rotation without applying nonlinear minimization. Finally, section six is dedicated to the experimental results and section seven to the conclusions.

2. Geometric Algebra: An Outline

Geometric algebra is a coordinate-free approach to geometry based on the algebras of Grassmann [15] and Clifford [10]. The geometric approach to Clifford algebra adopted in this paper was pioneered in the 1960's by

David Hestenes [16] 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 [17, 18]. Hestenes also presented a treatment of the projective geometry using Clifford algebra [19]. Some preliminary applications of geometric algebra in the field of computer vision and neural computing have already been given [3, 4, 24, 37]. In this paper the authors present the mathematics of 3D kinematics as framework for computer vision and robotics and extend previous results [11] from the perspective of the geometric algebra. We will begin in next subsections with basic definitions of geometric algebra necessary for 3D kinematics. In the whole paper we will denote scalars with lower case letters, matrices with upper case letters, and we will use bold lower case for both vectors in 3D and bivector parts of spinors. Spinors and dual quaternions in 4D are denoted by bold upper case letters.

2.1. The Geometric Product and Multivectors

The geometric algebra is defined on a space whose elements are called *multivectors*; a general multivector is a linear combination of objects of different type, e.g. scalars and vectors. In addition to vector addition and scalar multiplication geometric algebra has a non-commutative product which is associative and distributive over addition—this is the geometric or Clifford product. The existence of such a product and the calculus associated with the geometric algebra give the system tremendous power. A further distinguishing feature of this graded algebra is that any multivector squares to a scalar. The geometric product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written $\boldsymbol{ab}$ and can be expressed as a sum of its symmetric and antisymmetric parts

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

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

$$\boldsymbol{a}\cdot\boldsymbol{b} = \tfrac{1}{2}(\boldsymbol{ab}+\boldsymbol{ba}) \tag{2}$$

$$\boldsymbol{a}\wedge\boldsymbol{b} = \tfrac{1}{2}(\boldsymbol{ab}-\boldsymbol{ba}). \tag{3}$$

The inner product of two vectors is the standard *scalar* or *dot* product which results in 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

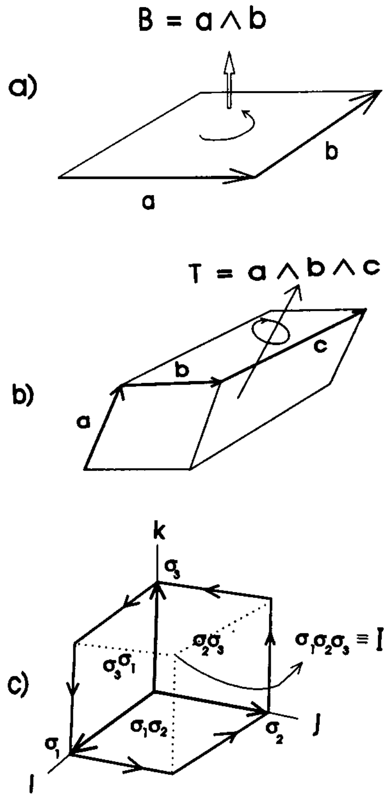


Figure 1. (a) bivector $\boldsymbol{B}$, (b) trivector $\boldsymbol{T}$, (c) 3D basis.

in the plane containing $\boldsymbol{a}$ and $\boldsymbol{b}$, formed by sweeping $\boldsymbol{a}$ along $\boldsymbol{b}$—see Fig. 1(a).

Thus, $\boldsymbol{b}\wedge\boldsymbol{a}$ will have the opposite orientation making the outer product anti-commutative as given in Eq. (3). The outer product is immediately generalizable to higher dimensions—for example, $(\boldsymbol{a}\wedge\boldsymbol{b})\wedge\boldsymbol{c}$, a *trivector*, is interpreted as the oriented volume formed by sweeping the area $\boldsymbol{a}\wedge\boldsymbol{b}$ along vector $\boldsymbol{c}$, see Fig. 1(b). The outer product of k linear independent vectors is a k-blade, and such a quantity is said to have *grade* k. A *multivector* is *homogeneous* if it contains terms of only a single grade. Thus, a k-vector is a homogeneous multivector of grade k computed as a linear combination of linear independent k-blades. The geometric algebra provides means of manipulating multivectors which allows us to keep track of different grade objects simultaneously.

Any couple of any multivectors can be multiplied using the geometric product. Consider two homogeneous multivectors $\boldsymbol{A}_r$ and $\boldsymbol{B}_s$ of grades r and s, respectively. The geometric product of $\boldsymbol{A}_r$ and $\boldsymbol{B}_s$ can be written as

$$\boldsymbol{A}_r\boldsymbol{B}_s = \langle\boldsymbol{AB}\rangle_{r+s} + \langle\boldsymbol{AB}\rangle_{r+s-2} + \cdots + \langle\boldsymbol{AB}\rangle_{|r-s|} \tag{4}$$

where $\langle \boldsymbol{AB}\rangle_t$ is used to denote the t-grade part of multivector $\boldsymbol{A}_r\boldsymbol{B}_s$, e.g. $\boldsymbol{ab} = \langle \boldsymbol{ab}\rangle_0 + \langle \boldsymbol{ab}\rangle_2 = \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{b}$. Note that $\langle \boldsymbol{AB}\rangle_0$ corresponds to a full contraction or inner product and $\langle \boldsymbol{AB}\rangle_{|r-s|}$ a generalized contraction or generalized inner product. Since the elements of $\boldsymbol{A}_r\boldsymbol{B}_s$ are of different grade $\boldsymbol{A}_r\boldsymbol{B}_s$ is thus an inhomogeneous multivector. In the following sections expressions of grade zero will be written ignoring their subindex, i.e. $\langle \boldsymbol{ab}\rangle_0 = \langle \boldsymbol{ab}\rangle$.

As simple illustration the geometric product of $\boldsymbol{A} = 5\sigma_3 + 3\sigma_1\sigma_2$ and $\boldsymbol{b} = 9\sigma_2 + 7\sigma_3$ is

$$\begin{aligned}\boldsymbol{Ab} &= 35(\sigma_3)^2 + 27\sigma_1(\sigma_2)^2 + 45\sigma_3\sigma_2 + 21\sigma_1\sigma_2\sigma_3 \\ &= 35 + 27\sigma_1 - 45\sigma_2\sigma_3 + 21I, \end{aligned} \tag{5}$$

note that $\sigma_i\sigma_i = (\sigma_i)^2 = \sigma_i\cdot\sigma_i = 1$ and $\sigma_i\sigma_j = \sigma_i\wedge\sigma_j$, where the geometric product or equal unit basis vectors equals 1 and of different ones equals to their wedge, which for simple notation can be omitted. The reader can try computations in Clifford algebra using the software package CLICAL [26].

2.2. *The Geometric Algebra of the n-dimensional Space*

An n-dimensional vector space can be spanned by the orthonormal basis of vectors $\{\sigma_i\}$, $i = 1,\ldots,n$, such that $\sigma_i\cdot\sigma_j = \delta_{ij}$. This leads to a basis to span the linear space of the entire geometric algebra $\mathcal{G}_n$ of that vector space

$$\begin{aligned} &1, \quad \{\sigma_i\}, \quad \{\sigma_i\wedge\sigma_j\}, \quad \{\sigma_i\wedge\sigma_j\wedge\sigma_k\}, \ldots, \\ &\qquad I = \sigma_1\wedge\sigma_2\wedge\cdots\wedge\sigma_n. \end{aligned} \tag{6}$$

The dimension of the linear space is 2^n. In $\mathcal{G}_n$ we can find multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), grade 3 (trivectors), etc. up to grade n.

The multivector $I = \sigma_1\wedge\sigma_2\wedge\cdots\wedge\sigma_n$ is called unit pseudoscalar or unit hypervolume. Depending of the algebraic properties we want to enforce in a geometric algebra $\mathcal{G}_n$ we select basis vectors which square according $\boldsymbol{\sigma}_i^2 > 0$, <0, or $=0$. This indicates the dimensions of the maximal involved subspaces with positive, negative and zero signatures. Thus the signature of $\mathcal{G}_n$ will be uniquely specified by $\mathcal{G}_{p,q,r}$ where p, q and r stand for the numbers of basis vectors which square to $+1$, -1 and 0, respectively.

The multivector basis elements of even grade span a subalgebra of $\mathcal{G}_{p,q,r}$ which we will denote by $\mathcal{G}^+_{p,q,r}$ and the multivector basis elements of odd grade span the linear space $\mathcal{G}^-_{p,q,r}$ which does not actually constitute an algebra.

In the multivector basis of a geometric algebra there is a dual relationship between the individual multivector basis components. This property is the result of the geometric product between a t-blade $\boldsymbol{A}_t$ and the unit pseudoscalar I as follows,

$$\begin{aligned} \boldsymbol{A}^* &= I\boldsymbol{A} \\ \langle \boldsymbol{A}^*\rangle_{n-t} &= \langle \boldsymbol{A}\rangle_t\langle I\rangle_n, \end{aligned} \tag{7}$$

where $\boldsymbol{A}^*$ stands for the dual of $\boldsymbol{A}$. As simple examples of that dual relationship in $\mathcal{G}_{3,0,0}$ the duals of the vectors $\{\sigma_i\}$ are the bivectors $\sigma_i\wedge\sigma_j = I\sigma_k$ and the duals of scalars are the trivectors.

Since the multivector basis of some grade spans a subspace of the geometric algebra its dual multivector basis will span the dual subspace. The duality relates the dual subspaces and in addition to the signature of I indicates whether the duality is regarding complex numbers ($I^2 = -1$), double numbers ($I^2 = 1$) or dual numbers ($I^2 = 0$).

The concept of duality can be also seen in the dual relation between the outer and the inner products. This relation known as Hodge dual [27] involves the hypervolume or pseudoscalar as follows

$$\boldsymbol{A}\cdot\boldsymbol{B} = \boldsymbol{A}^*\wedge\boldsymbol{B} = (I\boldsymbol{A})\wedge\boldsymbol{B}, \tag{8}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are k-blades. The Hodge dual depends not only on the metric but also on the orientation of the pseudoscalar. This equation is very useful when we want to express an inner product in terms of the outer product for the simplification of complex equations.

2.3. *Geometric Algebra of General Complex Numbers*

The most general complex numbers [22, 40] can be categorized into three different systems which are ordinary complex numbers, double and dual numbers respectively. In general they can be represented as a *composed number* $\mathbf{a} = b + \omega c$ using the algebraic operator ω which in case of complex numbers $\omega^2 = -1$, in case of double numbers $\omega^2 = 1$ and in case of the dual numbers $\omega^2 = 0$. In case of dual numbers the term b is called the real part and c the dual part.

In this paper we will require the notion of a function of a dual variable. A differentiable real function f :

$\mathcal{R} \to \mathcal{R}$ with a dual argument $\alpha + \omega\beta$ where $\alpha, \beta \in \mathcal{R}$ can be expanded in terms of a Taylor series. Because of $\omega^2 = \omega^3 = \omega^4 = \cdots = 0$ the function reads

$$\begin{aligned} f(\alpha + \omega\beta) &= f(\alpha) + \omega f'(\alpha)\beta + \omega^2 f''(\alpha)\frac{\beta^2}{2!} + \cdots \\ &= f(\alpha) + \omega f'(\alpha)\beta. \end{aligned} \tag{9}$$

A useful illustration of this is the exponential function of a dual number

$$e^{\alpha+\omega\beta} = e^{\alpha} + \omega e^{\alpha}\beta = e^{\alpha}(1 + \omega\beta). \tag{10}$$

In the seminal paper *Preliminary sketch of biquaternions* [9] Clifford introduced using dual numbers *the motors* or bi-quaternions for representing screw motion. Later on Study [38] used the dual numbers to represent the relative position of two skew lines in space, i.e. $\hat{\theta} = \theta + \omega d$, where $\hat{\theta}$ stands for the dual angle, θ for the difference of the line orientation angles and d for the distance between both lines.

The algebras of complex, double and dual (hyperbolic) numbers are isomorphic to certain geometric algebras. For that we have to choose the appropriate multivector basis so that the unit pseudoscalar squares to 1 for the case of double numbers to -1 for complex numbers and to 0 for dual numbers. Note that the pseudoscalar for these numbers keeps its geometric interpretation as a unit hypervolume and like ω it is commutative with either vectors or bivectors depending only of the type of the geometric algebra.

Let us consider some examples of composed numbers in geometric algebra: the complex numbers can be found in $\mathcal{G}_{0,1,0}$, the double numbers in $\mathcal{G}_{0,1,1}$ and the dual complex numbers in $\mathcal{G}_{1,0,1}$.

Next subsections describe in some detail the complex and dual numbers in the geometric algebra of the 2D, 3D and 4D spaces. The dual numbers in this context will be used later for the modelling of points, lines and planes and also for the modelling of their motion.

2.4. 2D Geometric Algebras of the Plane

In this section we want to illustrate the application of different 2D geometric algebras for the modelling of group transformations on the plane. Doing that we can also see clearly the geometric interpretation and the use of the complex, double and dual numbers for the cases of rotation, affine and Lorentz transformations, respectively [33, 40]. These transformations we will find in various tasks of image processing. For the modelling of the 2D space we choose a geometric algebra which has $2^2 = 4$ elements given by:

$$\underbrace{1}_{\text{scalar}},\ \underbrace{\sigma_1, \sigma_2}_{\text{vectors}},\ \underbrace{\sigma_1\sigma_2}_{\text{bivector}} \equiv I. \tag{11}$$

The highest grade element for the 2D space is a bivector called unit pseudoscalar $I \equiv \sigma_1\sigma_2$. According to the basis vector signature we will get complex, double or dual numbers. These cases will be now illustrated one by one.

In the geometric algebra $\mathcal{G}_{2,0,0}$, where $I = \sigma_1\sigma_2$ with $I^2 = -1$, we will represent the rotation of the points (x, y) of the Euclidean plane. Here a rotation of the point $\boldsymbol{z} = x\sigma_1 + y\sigma_2 = r(\cos\alpha\sigma_1 + \sin\alpha\sigma_2) \in \mathcal{G}_{2,0,0}$ can be computed as the geometric product of the vector with the complex number $e^{I\frac{\theta}{2}} = \cos\frac{\theta}{2} + \sigma_1\sigma_2 \sin\frac{\theta}{2} = (\cos\frac{\theta}{2} + I\sin\frac{\theta}{2}) \in \mathcal{G}^{+}_{2,0,0}$ or $\in$ Spin(2) (spin group) as follows

$$\begin{aligned} \boldsymbol{z}' &= e^{-I\frac{\theta}{2}}\boldsymbol{z}e^{I\frac{\theta}{2}} \\ &= e^{-I\frac{\theta}{2}} r(\cos\alpha\sigma_1 + \sin\alpha\sigma_2) e^{I\frac{\theta}{2}} \\ &= \left(\cos\frac{\theta}{2} + I\sin\frac{\theta}{2}\right)^{-1} r(\cos\alpha\sigma_1 + \sin\alpha\sigma_2) \\ &\quad \left(\cos\frac{\theta}{2} + I\sin\frac{\theta}{2}\right) \\ &= r(\cos(\alpha+\theta)\sigma_1 + \sin(\alpha+\theta)\sigma_2). \end{aligned} \tag{12}$$

We can see in Fig. 2(b) that each point of the 2D image of the dice is rotated by θ. Note that this particular form to represent rotation applying $e^{I\frac{\theta}{2}} = (\cos\frac{\theta}{2} + I\sin\frac{\theta}{2})$ can be generalized to higher dimensions, see the 3D rotors in next section.

Let us now represent the points as dual numbers in $\mathcal{G}_{1,0,1}$, where $I^2 = 0$. A 2D point can be represented in $\mathcal{G}_{1,0,1}$ as $\boldsymbol{z} = x\sigma_1 + y\sigma_2 = x(\sigma_1 + s\sigma_2)$, where $s = \frac{y}{x}$ is the slope. The shear transformation of this point can be computed applying a unit shear dual number $e^{I\frac{\tau}{2}} = (1 + I\frac{\tau}{2}) \in \mathcal{G}_{1,0,1}$ as follows

$$\begin{aligned} \boldsymbol{z}' &= e^{-I\frac{\tau}{2}}\boldsymbol{z}e^{I\frac{\tau}{2}} \\ &= \left(1 - I\frac{\tau}{2}\right)(x(\sigma_1 + s\sigma_2))\left(1 + I\frac{\tau}{2}\right) \\ &= x(\sigma_1 + (s+\tau)\sigma_2). \end{aligned} \tag{13}$$

Note that the overall effect of the transformation is to shear the plane, where the points (x, y) lie parallel to the σ_2-axis through the shear τ with a shear angle

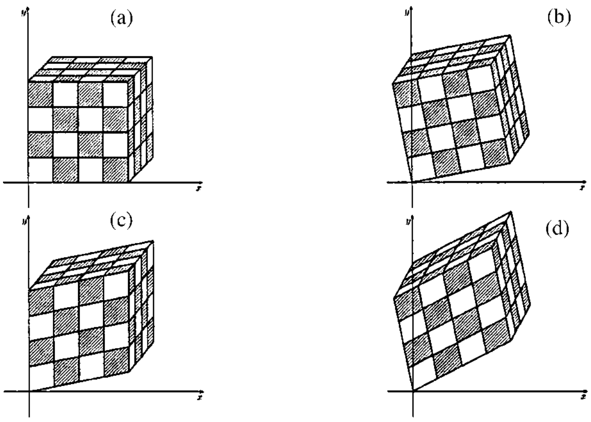


Figure 2. Action of the 2D transformations: (a) original figure, (b) rotation, (c) shear and (d) Lorentz transformations.

of $\tan^{-1}\tau$. Figure 2(c) depicts the effect of the shear transformation acting on the 2D image of the dice.

Using the representation of the double number in $\mathcal{G}_{1,1,0}$, where $I^2 = 1$, we can implement the Lorentz transformation of the points. This transformation is commonly used in space-time of special relativity and proposed to be used in psychophysics [12, 20]. In this context a 2D point is associated with a double number $\boldsymbol{z} = t\sigma_1 + x\sigma_2 = \rho(\cosh\alpha\sigma_1 + \sinh\alpha\sigma_2) \in \mathcal{G}_{1,1,0}$. The lines $|t| = |x|$ divide the plane in two quadrants with $|t| > |x|$ and two others with $|t| < |x|$. If one applies a 2D unit displacement vector $e^{I\frac{\beta}{2}} = a + Ib = (\cosh\beta + I\sinh\beta) \in \mathcal{G}_{1,1,0}$ from one of the quadrants $|t| > |x|$ to an arbitrary point $\boldsymbol{z} = t + Ix$ we get

$$\begin{aligned}
\boldsymbol{z}' &= e^{-I\frac{\beta}{2}}\boldsymbol{z}e^{I\frac{\theta}{2}} \\
&= e^{-I\frac{\beta}{2}}\rho(\cosh\alpha\sigma_1 + \sinh\alpha\sigma_2)e^{I\frac{\theta}{2}} \\
&= \left(\cosh\frac{\beta}{2} + I\sinh\frac{\beta}{2}\right)^{-1}(\rho(\cosh\alpha\sigma_1 + \sinh\alpha\sigma_2)) \\
&\quad \left(\cosh\frac{\beta}{2} + I\sinh\frac{\beta}{2}\right) \\
&= \rho(\cosh(\alpha+\beta)\sigma_1 + \sinh(\alpha+\beta)\sigma_2).
\end{aligned} \tag{14}$$

The point is displaced along a particular hyperbolic path through the interval $\rho\beta$ in $|t| < |x|$. Figure 2(d) shows the effect of the Lorentz transformation acting on the 2D image of the dice.

3. Geometric Algebra of the Euclidean 3D Space

In the case of embedding the Euclidean 3D space we choose the geometric algebra $\mathcal{G}_{3,0,0}$ which has $2^3 = 8$ elements given by:

$$\underbrace{1}_{\text{scalar}}, \quad \underbrace{\{\sigma_1, \sigma_2, \sigma_3\}}_{\text{vectors}}, \quad \underbrace{\{\sigma_1\sigma_2, \sigma_2\sigma_3, \sigma_3\sigma_1\}}_{\text{bivectors}},$$
$$\underbrace{\{\sigma_1\sigma_2\sigma_3\} \equiv I}_{\text{trivector}}. \tag{15}$$

The highest grade algebraic element for the 3D space is a trivector called unit pseudoscalar $I \equiv \sigma_1\sigma_2\sigma_3$ which squares to -1 and which commutes with the scalars and bivectors in the 3D space. In the algebra of three dimensional space we can construct a trivector $\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c} = \lambda I$, where the points are in general position and $\lambda \in \mathcal{R}$. Note that no 4-vectors exist since there is no possibility of sweeping the volume element $\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c}$ over a 4th dimension.

Multiplication of the three basis vectors σ_1, σ_2, and σ_3 by I results in the three basis bivectors $\sigma_1\sigma_2 = I\sigma_3$, $\sigma_2\sigma_3 = I\sigma_1$ and $\sigma_3\sigma_1 = I\sigma_2$. 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 unit vectors $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ of the quaternion algebra with $I\sigma_1, -I\sigma_2, I\sigma_3$ the famous Hamilton relations $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$ can be recovered. Since the $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are really bivectors it comes as no surprise that they represent 90° rotations in orthogonal directions and provide a system well-suited for the representation of general 3D rotations, see Fig. 1(c).

3.1. *Rotors*

In geometric algebra a **rotor** (short name for rotator), $\boldsymbol{R}$, is an even-grade element of the algebra which satisfies $\boldsymbol{R}\tilde{\boldsymbol{R}} = 1$, where $\tilde{\boldsymbol{R}}$ stands for the conjugate of $\boldsymbol{R}$. If $A = \{a_0, a_1, a_2, a_3\} \in \mathcal{G}_{3,0,0}$ represents a unit quaternion, then the rotor which performs the same rotation is simply given by

$$\begin{aligned}
\boldsymbol{R} &= \underbrace{a_0}_{\text{scalar}} + \underbrace{a_1(I\sigma_1) - a_2(I\sigma_2) + a_3(I\sigma_3)}_{\text{bivectors}} \\
&= a_0 + a_1\sigma_2\sigma_3 - a_2\sigma_3\sigma_1 + a_3\sigma_3\sigma_1.
\end{aligned} \tag{16}$$

The quaternion algebra is therefore seen to be a subset of the geometric algebra of 3-space. The conjugate $\boldsymbol{R}$ is

$$\tilde{\boldsymbol{R}} = a_0 - a_1\sigma_2\sigma_3 + a_2\sigma_3\sigma_1 - a_3\sigma_3\sigma_1. \tag{17}$$

Consider in $\mathcal{G}_{3,0,0}$ two non-parallel vectors which are referred to the same origin. With a rotation operation

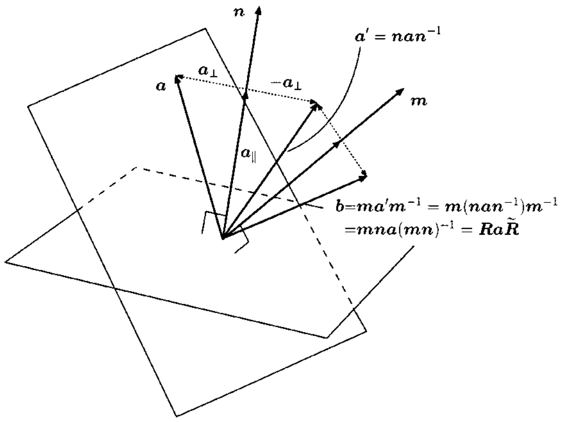


Figure 3. The rotor in the 3D space formed by a pair of reflections.

these vectors can be overlapped, see Fig. 3. In general a rotation can be performed by a pair of reflections. It can easily be shown that the result of reflecting a vector $\boldsymbol{a}$ in the plane perpendicular to a unit vector $\boldsymbol{n}$ is $\boldsymbol{a}_\perp - \boldsymbol{a}_\parallel = -\boldsymbol{nan}$, where $\boldsymbol{a}_\perp$ and $\boldsymbol{a}_\parallel$ respectively denote projections of $\boldsymbol{a}$ perpendicular and parallel to $\boldsymbol{n}$. Thus, a reflection of $\boldsymbol{a}$ in the plane perpendicular to $\boldsymbol{n}$, followed by a reflection in the plane perpendicular to another unit vector $\boldsymbol{m}$ results in a new vector $\boldsymbol{b} = -\boldsymbol{m}(-\boldsymbol{nan})\boldsymbol{m} = (\boldsymbol{mn})\boldsymbol{a}(\boldsymbol{nm}) = \boldsymbol{R}\boldsymbol{a}\tilde{\boldsymbol{R}}$. Using the geometric product it will be shown that the rotor $\boldsymbol{R}$ of Eq. (16) is a multivector consisting of both a scalar part and a bivector part, i.e. $\boldsymbol{R} = \boldsymbol{mn} = \boldsymbol{m} \cdot \boldsymbol{n} + \boldsymbol{m} \wedge \boldsymbol{n}$. These components correspond to the scalar and bivector parts of an equivalent unit quaternion in $\mathcal{G}_{3,0,0}$ and thus, $\boldsymbol{R} \in \mathcal{G}^+_{3,0,0}$. Considering the scalar and the bivector parts we can further write the Euler representation of a rotor as follows

$$\boldsymbol{R} = e^{\boldsymbol{n}\frac{\theta}{2}} = \cos\frac{\theta}{2} + \boldsymbol{n}\sin\frac{\theta}{2}, \tag{18}$$

where now $\boldsymbol{n} = n_1\sigma_2\sigma_3 + n_2\sigma_3\sigma_1 + n_3\sigma_1\sigma_2$ is spanned by the bivector basis of $\mathcal{G}^+_{3,0,0}$. Equation (18) is the spinor representation of the rotor. Let us compare it with the 2D rotation in $\mathcal{G}_{2,0,0}$ computed in terms of a spinor using complex numbers in Eq. (12). While complex and quaternionic numbers (see Eqs. (12 and 18)) used as operators of rotations are related to specific spaces, spinors are more general representations for rotation operators.

The transformation in terms of a rotor as spinor $\boldsymbol{a} \mapsto \boldsymbol{R}\boldsymbol{a}\tilde{\boldsymbol{R}} = \boldsymbol{b}$ is a very general way of handling rotations; it works for multivectors of any grade and in spaces of any dimension in contrast to quaternion calculus. Due to the group properties of spinors, rotors combine in a straightforward manner, i.e. a rotor $\boldsymbol{R}_1$ followed by a rotor $\boldsymbol{R}_2$ is equivalent to a total rotor $\boldsymbol{R}$ where $\boldsymbol{R} = \boldsymbol{R}_2\boldsymbol{R}_1$.

3.2. Representation of Points, Lines and Planes Using a 3D Geometric Algebra

The modelling of the points, lines and planes in the 3D Euclidean space will be done using the Euclidean geometric algebra $\mathcal{G}_{3,0,0}$ where the pseudoscalar $I^2 = -1$. A point in the 3D space represents a position, thus it can be simply spanned using the vector basis of $\mathcal{G}_{3,0,0}$

$$\boldsymbol{x} = x\sigma_1 + y\sigma_2 + z\sigma_3 \tag{19}$$

where $x,\ y,\ z \in \mathcal{R}$.

In the classical vector calculus a line is described by a position vector $\boldsymbol{x}$ touching any point of the line and a vector $\boldsymbol{n}$ for the line direction, i.e. $\boldsymbol{l} = \boldsymbol{x} + \alpha\boldsymbol{n}$, where $\alpha \in \mathcal{R}$. In geometric algebra we have the multivector concept, thus we can represent in $\mathcal{G}_{3,0,0}$ a line compactly using a vector $\boldsymbol{n}$ for its direction and a bivector $\boldsymbol{m}$ for the orientation of the plane within which this line lies, namely

$$\boldsymbol{l} = \boldsymbol{n} + \boldsymbol{x} \wedge \boldsymbol{n} = \boldsymbol{n} + \boldsymbol{m}, \tag{20}$$

note that the moment bivector $\boldsymbol{m}$ is computed as the outer product of the position vector $\boldsymbol{x}$ and the the line direction vector $\boldsymbol{n}$. We can also compute $\boldsymbol{m}$ as the dual of a vector, i.e. $I\boldsymbol{x} \times \boldsymbol{n} = \boldsymbol{m}$. The representation of the plane is even more striking. The plane is a geometric entity one grade higher than the line, so we should expect that the multivector representation of the plane should be a natural multivector grade extension from that of the line. In the classical vector calculus a plane is described in terms of the Hesse distance from the origin to the plane and a vector indicating the plane orientation, i.e. $\{d,\ \boldsymbol{n}\}$. Note that this description comprises of two separate attributes. Again in the geometric algebra we can resort to a compact expression with clear geometric sense. In $\mathcal{G}_{3,0,0}$ the extension of the line expression to a plane should be done in terms of a bivector $\boldsymbol{n}$ and a trivector Id as follows

$$\boldsymbol{h} = \boldsymbol{n} + \boldsymbol{x} \wedge \boldsymbol{n} = \boldsymbol{n} + Id, \tag{21}$$

where the bivector $\boldsymbol{n}$ is now indicating the plane orientation, and the outer product of the position vector $\boldsymbol{x}$ and the bivector $\boldsymbol{n}$ builds a trivector which can be expressed using the Hesse distance, a scalar value, and the unit pseudoscalar I. Note that the trivector is a volume with the Hesse distance d as a weight.

3.3. *Motion of Points, Lines and Planes in the 3D Geometric Algebra*

The 3D motion of a point $\boldsymbol{x}$ in $\mathcal{G}_{3,0,0}$ has the following equation

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t}. \tag{22}$$

Using the Eq. (20) the motion equation of the line reads

$$\begin{aligned}
\boldsymbol{l}' &= \boldsymbol{n}' + \boldsymbol{m}' = \boldsymbol{n}' + \boldsymbol{x}' \wedge \boldsymbol{n}' \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + (\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t}) \wedge (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}) \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{t} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \frac{\boldsymbol{t}}{2}\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} - \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}\frac{\boldsymbol{t}}{2} \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{n}\frac{\boldsymbol{t}}{2}\tilde{\boldsymbol{R}} + \frac{\boldsymbol{t}}{2}\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{m}\tilde{\boldsymbol{R}},
\end{aligned} \tag{23}$$

where $\boldsymbol{x}'$ stands for the rotated and shifted position vector, $\boldsymbol{n}'$ stands for the rotated orientation vector and $\boldsymbol{m}'$ for the new line moment. The model of the motion of the plane in $\mathcal{G}_{3,0,0}$ can be expressed in terms of the multivector Eq. (21) as follows

$$\begin{aligned}
\boldsymbol{h}' &= \boldsymbol{n}' + Id' = \boldsymbol{n}' + \boldsymbol{x}' \wedge \boldsymbol{n}' \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + (\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t}) \wedge (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}) \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{t} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{t} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{x} \wedge \boldsymbol{n}\tilde{\boldsymbol{R}} \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{t} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{R}(Id)\tilde{\boldsymbol{R}} \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + \boldsymbol{t}^* \cdot \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + Id \\
&= \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + I(\boldsymbol{t} \cdot (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}) + d),
\end{aligned} \tag{24}$$

where $\boldsymbol{n}'$ stands for the rotated bivector plane orientation, $\boldsymbol{x}'$ stands for the rotated and shifted position vector and d' for the new Hesse distance. Here we use the concept of duality to claim that $\boldsymbol{t} \wedge \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} = \boldsymbol{t}^* \cdot \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} = (\boldsymbol{I}\boldsymbol{t}) \cdot \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}$, see Eq. (8).

4. The 4D Geometric Algebra for the Projective 3D Space

Until now we have dealt with transformations in 3D. When we use homogeneous coordinates we increase the dimension of the vector space by one. As a result the transformation of the 3D motion becomes linear. Let us now model the projective 3D space P^3. This space corresponds to the homogeneous extended space R^4. In real applications it is important to regard the signature of the modeled space to facilitate the computations. In the case of the modelling of the projective plane using homogeneous coordinates we adopt $\mathcal{G}_{3,0,0}$ of the ordinary space, E^3, which has the standard Euclidean signature. For the 4-dimensional space R^4 we are forced to adopt the same signature as in the case of the Euclidean space. This geometric algebra $\mathcal{G}_{1,3,0}$ is spanned with the following basis

$$\underbrace{1}_{\text{scalar}}, \quad \underbrace{\gamma_k}_{\text{4 vectors}}, \quad \underbrace{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3}_{\text{6 bivectors}}, \quad \underbrace{I\gamma_k}_{\text{4 pseudovectors}}, \quad \underbrace{I}_{\text{unit pseudoscalar}} \tag{25}$$

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

$$I^2 = (\gamma_1\gamma_2\gamma_3\gamma_4)(\gamma_1\gamma_2\gamma_3\gamma_4) = -(\gamma_3\gamma_4)(\gamma_3\gamma_4) = -1. \tag{26}$$

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

4.1. *The 4D Geometric Algebra for 3D Kinematics*

Usually problems of robotics are treated in algebraic systems of the 2D and 3D space. In the case of 3D rigid motion or Euclidean transformation we are confronted with a nonlinear mapping, however if we employ homogeneous coordinates in 4D geometric algebra we linearize the rigid motion in the 3D Euclidean space. That is why we choose three basis vectors which square to one and one which squares to zero to provide dual copies of the multivectors of the 3D space. In other words we extend the Euclidean geometric algebra $\mathcal{G}_{3,0,0}$ to the special or degenerated geometric algebra $\mathcal{G}_{3,0,1}$ which is spanned via the following basis

$$\underbrace{1}_{\text{scalar}},\quad \underbrace{\gamma_k}_{\text{4 vectors}},\quad \underbrace{\gamma_2\gamma_3,\gamma_3\gamma_1,\gamma_1\gamma_2,\gamma_4\gamma_1,\ \gamma_4\gamma_2,\gamma_4\gamma_3}_{\text{6 bivectors}},\quad \underbrace{I\gamma_k}_{\text{4 pseudovectors}},\quad \underbrace{I}_{\text{unit pseudoscalar}} \tag{27}$$

where $\gamma_4^2 = 0$, $\gamma_k^2 = +1$ for $k = 1, 2, 3$. The unit pseudoscalar is $I = \gamma_1\gamma_2\gamma_3\gamma_4$ with

$$I^2 = (\gamma_1\gamma_2\gamma_3\gamma_4)(\gamma_1\gamma_2\gamma_3\gamma_4) = -(\gamma_3\gamma_4)(\gamma_3\gamma_4) = 0. \tag{28}$$

4.2. *The Motor Algebra*

Clifford introduced the motors using the name biquaternions [5, 9]. The term motor is the abbreviation of "moment and vector". Motors are the dual numbers for 3D kinematics with the necessary condition of $I^2 = 0$. They can be found in the special 4D even subalgebra of $\mathcal{G}_{3,0,1}$ introduced in the previous subsection. This even subalgebra will be denominated by $\mathcal{G}^+_{3,0,1}$ and is spanned by the following basis

$$\underbrace{1}_{\text{scalar}},\quad \underbrace{\gamma_2\gamma_3,\gamma_3\gamma_1,\gamma_1\gamma_2,\gamma_4\gamma_1,\gamma_4\gamma_2,\ \gamma_4\gamma_3}_{\text{6 bivectors}},\quad \underbrace{I}_{\text{unit pseudoscalar}}. \tag{29}$$

This kind of basis structure allows also to represent spinors which are composed of a scalar and bivector parts. As a result, motors are spinors and as a such a special kind of rotors. Because an Euclidean transformation includes both rotation and translation, we will find in the following a spinor representation also for translation to achieve the group properties of spinors in the definition of motors. But first we will relate the motors with the screw motion theory.

Note that the bivector part of the basis corresponds to the same basis for spanning 3D lines. Also note that the dual of a scalar is the pseudoscalar P and the duals of the first three basis bivectors are the next three ones that is for example $(\gamma_2\gamma_3)^* = I\gamma_2\gamma_3 = \gamma_4\gamma_1$.

According to Clifford [9] a basic geometric interpretation of a motor can be seen as the necessary operation to convert the rotation axis of a rotor into another one. Each rotor can be geometrically represented as a rotation plane with the rotation axis normal to this plane. Thus, one rotor can be spanned using a scalar and the

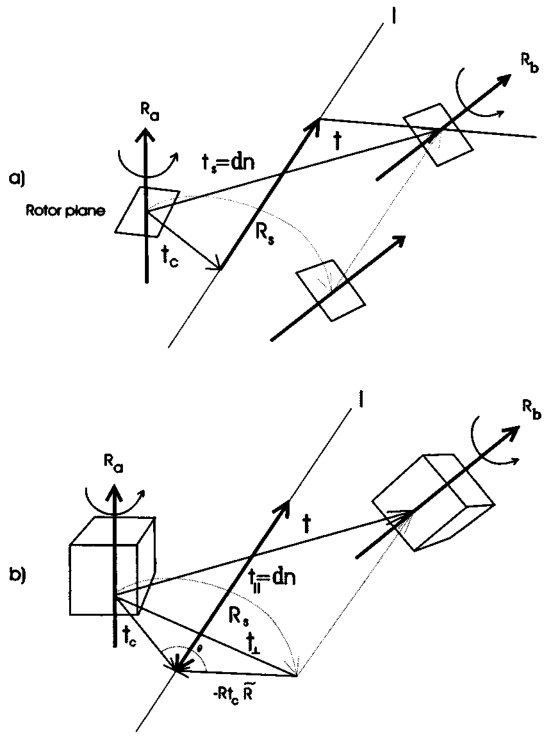


Figure 4. Screw motion about the line axis $\boldsymbol{l}$ ($\boldsymbol{t}_s$: longitudinal displacement by d and $\boldsymbol{R}_s$: rotation angle θ) (a) the motor relating two axis lines, (b) motor applied to an object. Note that the indicated vectors in the figures will be represented in Subsection 4.3 as bivectors.

bivector basis $\gamma_2\gamma_3$, $\gamma_3\gamma_1$, $\gamma_1\gamma_2$ and the dual one by a pseudoscalar and $\gamma_4\gamma_1$, $\gamma_4\gamma_2$, $\gamma_4\gamma_3$. Figure 4(a) depicts a motor action in detail where the rotor axes are now considered as rotation lines. In the Figure let us first turn the orientation of the axis of one rotor $\boldsymbol{R}_a$ parallel to the other one $\boldsymbol{R}_b$ by applying the rotor $\boldsymbol{R}_s$. Then slide it the distance d along the connecting axis into the position of the axis of the second rotor. These operations can be seen together as forming a twist about a screw with the line axis $\boldsymbol{l}$ and the relation called *pitch* which equals to $|\boldsymbol{t}_s| = \frac{d}{\theta}$ for $\theta \neq 0$. The Fig. 4(b) shows an action of a motor on a real object. In this case the motor relates the rotation axis line of the initial position of the object to the rotation axis line of its final position. Note that in both figures the angle and sliding distance indicate how the rigid displacement takes place around and along a screw line axis $\boldsymbol{l}$ respectively.

We said in Subsection 3.1 that a rotor relates two vectors. Now, in the case of a motor it relates the rotation axes of two rotors. A motor is specified only by its direction and position of the screw axis line, twist angular magnitude and pitch.

4.3. Motors, Rotors and Translators in $\mathcal{G}^+_{3,0,1}$

Since a rigid motion consists of the transformations rotation and translation, it should be possible to split multiplicatively a motor in terms of these two spinor transformations which we will call a rotor and a translator. In the following we will denote all bivector components of a spinor by bold lower case letters. Let us now express this procedure algebraically. First of all let us consider a simple rotor in its Euler representation for a rotation by an angle θ,

$$\begin{aligned}\boldsymbol{R} &= a_0 + a_1\gamma_2\gamma_3 + a_2\gamma_3\gamma_1 + a_3\gamma_1\gamma_2\\ &= a_0 + \boldsymbol{a}\\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\boldsymbol{n}\\ &= a_c + a_s\boldsymbol{n}\end{aligned} \tag{30}$$

where $\boldsymbol{n}$ is the unit 3D bivector of the rotation-axis spanned by the bivector basis $\gamma_2\gamma_3$, $\gamma_3\gamma_1$, $\gamma_1\gamma_2$ and a_c, $a_s \in \mathcal{R}$. Now dealing with the rotor of a screw motion the rotation axis vector should be represented as screw axis line. For that we have to relate this rotation axis to a reference coordinate system at the distance $\boldsymbol{t}_c$. A translation $\boldsymbol{t}_c$ in 3D in the motor algebra $\mathcal{G}^+_{3,0,1}$ is represented by a spinor $\boldsymbol{T}_c$ called translator. Applying a translator from the left and its conjugated from the right to the rotor $\boldsymbol{R}$ we get a modified rotor

$$\begin{aligned}\boldsymbol{R}_s &= \boldsymbol{T}_c\boldsymbol{R}\tilde{\boldsymbol{T}}_c\\ &= \left(1 + I\frac{\boldsymbol{t}_c}{2}\right)(a_0 + \boldsymbol{a})\left(1 - I\frac{\boldsymbol{t}_c}{2}\right)\\ &= a_0 + \boldsymbol{a} + Ia_0\frac{\boldsymbol{t}_c}{2} + I\frac{\boldsymbol{t}_c}{2}\boldsymbol{a} - Ia_0\frac{\boldsymbol{t}_c}{2} - I\boldsymbol{a}\frac{\boldsymbol{t}_c}{2}\\ &= a_0 + \boldsymbol{a} + I\left(\frac{\boldsymbol{t}_c}{2}\boldsymbol{a} - \boldsymbol{a}\frac{\boldsymbol{t}_c}{2}\right)\\ &= a_0 + \boldsymbol{a} + I\left(\boldsymbol{a}\frac{\boldsymbol{t}_c}{2} - \frac{\boldsymbol{t}_c}{2}\boldsymbol{a}\right)\\ &= a_0 + \boldsymbol{a} + I(\boldsymbol{a}\wedge\boldsymbol{t}_c),\end{aligned} \tag{31}$$

which is a dual quaternion representation. Note that $\boldsymbol{t}_c\boldsymbol{a} = \boldsymbol{a}\boldsymbol{t}_c$, because bivectors commute.

Expressing the last equation in Euler terms we get the spinor representation

$$\begin{aligned}\boldsymbol{R}_s &= a_0 + a_s\boldsymbol{n} + Ia_s\boldsymbol{n}\wedge\boldsymbol{t}_c\\ &= a_c + a_s(\boldsymbol{n} + I\boldsymbol{m})\\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(\boldsymbol{n} + I\boldsymbol{m})\\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\boldsymbol{l}.\end{aligned} \tag{32}$$

This result is really interesting because the new rotor called $\boldsymbol{R}_s$ is a rotor to be applied now with respect to an axis line $\boldsymbol{l}$ expressed in dual terms of direction $\boldsymbol{n}$ and moment $\boldsymbol{m} = \boldsymbol{n}\wedge\boldsymbol{t}_c$.

Now to finally define the motor let us slide along the rotation axis line $\boldsymbol{l}$ the distance $\boldsymbol{t}_s = d\boldsymbol{n}$. Since a motor is applied from the left and its conjugated from the right we should use the half of $\boldsymbol{t}_s$ by the spinor $\boldsymbol{T}_s$ when we define the motor

$$\begin{aligned}\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s &= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)(a_0 + \boldsymbol{a} + I\boldsymbol{a}\wedge\boldsymbol{t}_c)\\ &= \left(1 + I\frac{d\boldsymbol{n}}{2}\right)(a_c + a_s\boldsymbol{n} + Ia_s\boldsymbol{n}\wedge\boldsymbol{t}_c)\\ &= a_c + a_s\boldsymbol{n} + Ia_s\boldsymbol{n}\wedge\boldsymbol{t}_c + I\frac{d}{2}a_c\boldsymbol{n} - I\frac{d}{2}a_s\boldsymbol{n}\boldsymbol{n}\\ &= \left(a_c - I\frac{d}{2}a_s\right) + \left(a_s + Ia_c\frac{d}{2}\right)(\boldsymbol{n} + I\boldsymbol{n}\wedge\boldsymbol{t}_c)\\ &= \left(a_c - Ia_s\frac{d}{2}\right) + \left(a_s + Ia_c\frac{d}{2}\right)\boldsymbol{l}.\end{aligned} \tag{33}$$

Note that this expression of the motor makes explicit the unit line bivector of the screw axis line $\boldsymbol{l}$.

Now let us express a motor using Euler representation. Substituting the constants $a_c = \cos(\frac{\theta}{2})$ and $a_s = \sin(\frac{\theta}{2})$ in the motor Eq. (33) and using the property of Eq. (9) we get

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{T}_s\boldsymbol{R}_s\\ &= \left(\cos\left(\frac{\theta}{2}\right) - I\sin\left(\frac{\theta}{2}\right)\frac{d}{2}\right)\\ &\quad + \left(\sin\left(\frac{\theta}{2}\right) + I\cos\left(\frac{\theta}{2}\right)\frac{d}{2}\right)\boldsymbol{l}\\ &= \cos\left(\frac{\theta}{2} + I\frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I\frac{d}{2}\right)\boldsymbol{l},\end{aligned} \tag{34}$$

which is a dual number representation of the spinor. Analyzing the obtained expressions

$$\begin{aligned}\boldsymbol{R} &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\boldsymbol{n}\\ \boldsymbol{R}_s &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\boldsymbol{l}\\ \boldsymbol{M} &= \cos\left(\frac{\theta}{2} + I\frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I\frac{d}{2}\right)\boldsymbol{l}\end{aligned} \tag{35}$$

we can see how from a simple rotor $\boldsymbol{R}$ expressed in terms of angle and the rotation axis $\boldsymbol{n}$, in the case of a pure screw rotation we change this axis to a rotation line axis $\boldsymbol{l}$ resulting $\boldsymbol{R}_s$ and finally for the motor the information of the sliding distance d is made explicit in terms of dual arguments of the trigonometric functions. It is also nice to see that the expression for the motor simply extends the expression of $\boldsymbol{R}_s$ using dual angles instead.

If we expand the exponential function of the dual bivectors using the Taylor series we get again the motor expression as spinor

$$e^{\boldsymbol{l}\frac{\theta}{2}+I\frac{\boldsymbol{t}_s}{2}} = \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)e^{\boldsymbol{l}\frac{\theta}{2}} = \boldsymbol{T}_s\boldsymbol{R}_s \tag{36}$$

where $I\frac{\boldsymbol{t}_s}{2} = I\frac{1}{2}(t_1\sigma_2\sigma_3 + t_2\sigma_3\sigma_1 + t_3\sigma_1\sigma_2) = \frac{1}{2}(t_1\sigma_4\sigma_1 + t_2\sigma_4\sigma_2 + t_3\sigma_4\sigma_3)$.

If we want to express the motor using only rotors in dual spinor representation we proceed as follows

$$\begin{aligned}\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s &= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\boldsymbol{R}_s \\ &= \boldsymbol{R}_s + I\frac{\boldsymbol{t}_s}{2}\boldsymbol{R}_s.\end{aligned} \tag{37}$$

Let us consider carefully the dual part of the motor. This is the geometric product of the bivector $\boldsymbol{t}_s$ and the rotor $\boldsymbol{R}_s$. Since both are expressed in terms of the same bivector basis their geometric product will be also expressed in this basis and this can be seen as a new rotor $\boldsymbol{R}'_s$. Thus, we can further write

$$\boldsymbol{M} = \boldsymbol{R}_s + I\frac{\boldsymbol{t}_s}{2}\boldsymbol{R}_s = \boldsymbol{R}_s + I\boldsymbol{R}'_s \tag{38}$$

In this equation the line axes of the rotors are differently oriented in space, see Fig. 4(a). That means that they represent the general case of non-coplanar rotors. If the sliding distance $\boldsymbol{t}_s$ is zero then the motor will degenerate to a rotor

$$\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s = \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\boldsymbol{R}_s = \left(1 + I\frac{0}{2}\right)\boldsymbol{R}_s = \boldsymbol{R}_s. \tag{39}$$

In this case the two generating axis lines of the motor are coplanar, thus the motor is called a degenerated one.

Finally, the bivector $\boldsymbol{t}_s$ can be expressed in terms of the rotors using previous results

$$\boldsymbol{R}'_s\tilde{\boldsymbol{R}}_s = \left(\frac{\boldsymbol{t}_s}{2}\boldsymbol{R}_s\right)\tilde{\boldsymbol{R}}_s, \tag{40}$$

therefore,

$$\boldsymbol{t}_s = 2\boldsymbol{R}'_s\tilde{\boldsymbol{R}}_s. \tag{41}$$

Figure 4 shows that **t** is a distance bivector relating the two rotation axes of the rotors $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$, and $\boldsymbol{t}_s$ is a bivector representing a displacement along the motor axis line. According to Fig. 4(b) the distance $\boldsymbol{t}$ considered here as a bivector can be computed in terms of the bivectors $\boldsymbol{t}_c$ and $\boldsymbol{t}_s$ as follows

$$\begin{aligned}\boldsymbol{t} &= \boldsymbol{t}_\perp + \boldsymbol{t}_\| \\ \boldsymbol{t} &= (\boldsymbol{t}_c - \boldsymbol{R}_s\boldsymbol{t}_c\tilde{\boldsymbol{R}}_s) + (\boldsymbol{t}\cdot\boldsymbol{n})\boldsymbol{n} \\ &= (\boldsymbol{t}_c - \boldsymbol{R}_s\boldsymbol{t}_c\tilde{\boldsymbol{R}}_s) + d\boldsymbol{n} \\ &= \boldsymbol{t}_c - \boldsymbol{R}_s\boldsymbol{t}_c\tilde{\boldsymbol{R}}_s + \boldsymbol{t}_s \\ &= \boldsymbol{t}_c - \boldsymbol{R}_s\boldsymbol{t}_c\tilde{\boldsymbol{R}}_s + 2\boldsymbol{R}'_s\tilde{\boldsymbol{R}}_s.\end{aligned} \tag{42}$$

So far the motor was studied from a geometrical point of view, next its more relevant algebraic properties are given.

4.4. Properties of Motors

A general motor can be expressed as

$$\boldsymbol{M}_\alpha = \alpha\boldsymbol{M} \tag{43}$$

where $\alpha \in \mathcal{R}$ and $\boldsymbol{M}$ is a unit motor as in previous sections. In this section we deal further with unit motors. The norm of a motor $\boldsymbol{M}$ is defined as follows

$$\begin{aligned}|\boldsymbol{M}| = \boldsymbol{M}\tilde{\boldsymbol{M}} &= \boldsymbol{T}_s\boldsymbol{R}_s\tilde{\boldsymbol{R}}_s\tilde{\boldsymbol{T}}_s \\ &= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\boldsymbol{R}_s\tilde{\boldsymbol{R}}_s\left(1 - I\frac{\boldsymbol{t}_s}{2}\right) \\ &= 1 + I\frac{\boldsymbol{t}_s}{2} - I\frac{\boldsymbol{t}_s}{2} = 1,\end{aligned} \tag{44}$$

where $\tilde{\boldsymbol{M}}$ is the conjugate motor and 1 is the identity of the motor multiplication. Now using the Eq. (38) and considering the unit motor magnitude we find two useful properties

$$\begin{aligned}|\boldsymbol{M}| = \boldsymbol{M}\tilde{\boldsymbol{M}} &= (\boldsymbol{R}_s + I\boldsymbol{R}'_s)(\tilde{\boldsymbol{R}}_s + I\tilde{\boldsymbol{R}}'_s) \\ &= \boldsymbol{R}_s\tilde{\boldsymbol{R}}_s + I(\tilde{\boldsymbol{R}}_s\boldsymbol{R}'_s + \tilde{\boldsymbol{R}}'_s\boldsymbol{R}_s) = 1.\end{aligned} \tag{45}$$

This requires that

$$\begin{aligned} \boldsymbol{R}_s \tilde{\boldsymbol{R}}_s &= 1 \\ \tilde{\boldsymbol{R}}_s \boldsymbol{R}'_s + \tilde{\boldsymbol{R}}'_s \boldsymbol{R}_s &= 0. \end{aligned} \tag{46}$$

The combination of two rigid motions can be expressed using two motors. The resultant motor describes the overall displacement, namely

$$\begin{aligned} \boldsymbol{M}_c = \boldsymbol{M}_a \boldsymbol{M}_b &= \left(\boldsymbol{R}_{s_a} + I\boldsymbol{R}'_{s_a}\right)\left(\boldsymbol{R}_{s_b} + I\boldsymbol{R}'_{s_b}\right) \\ &= \boldsymbol{R}_{s_a}\boldsymbol{R}_{s_b} + I\left(\boldsymbol{R}_{s_a}\boldsymbol{R}'_{s_b} + \boldsymbol{R}'_{s_a}\boldsymbol{R}_{s_b}\right) \\ &= \boldsymbol{R}_{s_c} + I\boldsymbol{R}'_{s_c}. \end{aligned} \tag{47}$$

Using the Eq. (38) let us express a motor in terms of dual spinors

$$\begin{aligned} \boldsymbol{M} = \boldsymbol{T}_s \boldsymbol{R}_s &= \boldsymbol{R}_s + I\boldsymbol{R}'_s \\ &= (a_0 + a_1\gamma_2\gamma_3 + a_2\gamma_3\gamma_2 + a_3\gamma_2\gamma_1) \\ &\quad + I(b_0 + b_1\gamma_2\gamma_3 + b_2\gamma_3\gamma_2 + b_3\gamma_2\gamma_1) \\ &= (a_0 + \boldsymbol{a}) + I(b_0 + \boldsymbol{b}). \end{aligned} \tag{48}$$

We can use another notation to enhance clearly the components of the real and dual parts of the motor as follows

$$\boldsymbol{M} = (a_0, \boldsymbol{a}) + I(b_0, \boldsymbol{b}), \tag{49}$$

where each term within the brackets consists of a scalar part and a 3D bivector.

A motor expressed in terms of a translator and a rotor is applied similarly as in the case of a rotor from the left and its conjugate from the right (motor reflections) to build an automorphism equivalent to the screw. Yet conjugating only the rotor or only the translator for the second reflection we can derive different types of automorphisms.

Changing the sign of the scalar and bivector in the real and the dual parts of the motor we get the following variations of a motor

$$\begin{aligned} \boldsymbol{M} &= (a_0 + \boldsymbol{a}) + I(b_0 + \boldsymbol{b}) = \boldsymbol{T}_s \boldsymbol{R}_s \\ \tilde{\boldsymbol{M}} &= (a_0 - \boldsymbol{a}) + I(b_0 - \boldsymbol{b}) = \tilde{\boldsymbol{R}}_s \tilde{\boldsymbol{T}}_s \\ \bar{\boldsymbol{M}} &= (a_0 + \boldsymbol{a}) - I(b_0 + \boldsymbol{b}) = \boldsymbol{R}_s \tilde{\boldsymbol{T}}_s \\ \tilde{\bar{\boldsymbol{M}}} &= (a_0 - \boldsymbol{a}) - I(b_0 - \boldsymbol{b}) = \tilde{\boldsymbol{R}}_s \boldsymbol{T}_s. \end{aligned} \tag{50}$$

The first, the second and the fourth versions will be used for the modelling of the motion of points, lines and planes.

Using the relations from above we get

$$\begin{aligned} a_0 &= \tfrac{1}{4}(\boldsymbol{M} + \tilde{\boldsymbol{M}} + \bar{\boldsymbol{M}} + \tilde{\bar{\boldsymbol{M}}}) \\ Ib_0 &= \tfrac{1}{4}(\boldsymbol{M} + \tilde{\boldsymbol{M}} - \bar{\boldsymbol{M}} - \tilde{\bar{\boldsymbol{M}}}) \\ \boldsymbol{a} &= \tfrac{1}{4}(\boldsymbol{M} - \tilde{\boldsymbol{M}} + \bar{\boldsymbol{M}} - \tilde{\bar{\boldsymbol{M}}}) \\ I\boldsymbol{b} &= \tfrac{1}{4}(\boldsymbol{M} - \tilde{\boldsymbol{M}} - \bar{\boldsymbol{M}} + \tilde{\bar{\boldsymbol{M}}}). \end{aligned} \tag{51}$$

4.5. *Representation of Points, Lines and Planes Using a 4D Geometric Algebra*

Now we will represent points, lines and planes in the 4D space. For that we choose the special algebra of the motors $\mathcal{G}^+_{3,0,1}$ which using a bivector basis spans in 4D the line space.

For the case of the point representation, we proceed embedding a 3D point on the hyperplane $X_4 = 1$, thus the equation of the point $\boldsymbol{X} \in \mathcal{G}^+_{3,0,1}$ reads

$$\begin{aligned} \boldsymbol{X} &= 1 + x_1\gamma_4\gamma_1 + x_2\gamma_4\gamma_2 + x_3\gamma_4\gamma_3 \\ &= 1 + I(x_1\gamma_2\gamma_3 + x_2\gamma_3\gamma_1 + x_3\gamma_1\gamma_2) \\ &= 1 + I\boldsymbol{x} \equiv (1, 0) + I(0, \boldsymbol{x}). \end{aligned} \tag{52}$$

We can see that in this expression the real part consists of the scalar 1 and the dual part of only of 3D bivector.

Since we are working in the algebra $\mathcal{G}^+_{3,0,1}$ spanned only by bivectors and scalars, we can see this special geometric algebra as the appropriate system for line modelling. As opposite to the line representation, the point and the plane are in some sense unsymmetric representations with respect to the scalar and bivector parts. Let us now rewrite the line equation (20) of $\mathcal{G}_{3,0,0}$ in the degenerated geometric algebra $\mathcal{G}^+_{3,0,1}$. We can express the vector and the dual vector of Eq. (20) in $\mathcal{G}^+_{3,0,1}$ as a bivector and a dual bivector. Since the product of the unit pseudoscalar $I = \gamma_1\gamma_2\gamma_3\gamma_4$ with any dual bivectors build from the basis $\{\gamma_4\gamma_1,\ \gamma_4\gamma_2,\ \gamma_4\gamma_3\}$ results in zero, we have to select the bivector basis $\{\gamma_2\gamma_3,\ \gamma_3\gamma_1,\ \gamma_1\gamma_2\}$ for representing the line

$$\boldsymbol{L} = \boldsymbol{n} + I\boldsymbol{m}, \tag{53}$$

where the bivectors for the line direction and the moment are computed using two bivector points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ lying on the line as follows

$$\begin{aligned} \boldsymbol{n} &= (\boldsymbol{x}_2 - \boldsymbol{x}_1) \\ &= (x_{21} - x_{11})\gamma_2\gamma_3 + (x_{22} - x_{12})\gamma_3\gamma_1 \end{aligned}$$

$$
\begin{aligned}
&\quad + (x_{23} - x_{13})\gamma_1\gamma_2 \\
&= L_{n_1}\gamma_2\gamma_3 + L_{n_1}\gamma_3\gamma_1 + L_{n_3}\gamma_1\gamma_2 \\
\boldsymbol{m} &= \boldsymbol{x}_1 \times \boldsymbol{x}_2 \qquad (54)\\
&= (x_{12}x_{23} - x_{13}x_{22})\gamma_2\gamma_3 + (x_{13}x_{21} - x_{11}x_{23})\gamma_3\gamma_1 \\
&\quad + (x_{11}x_{22} - x_{12}x_{21})\gamma_1\gamma_2 \\
&= L_{m_1}\gamma_2\gamma_3 + L_{m_2}\gamma_3\gamma_1 + L_{m_3}\gamma_1\gamma_2
\end{aligned}
$$

This line representation using dual numbers is easy to understand and to manipulate algebraically and it is fully equivalent to the one in terms of Plücker coordinates. Using the notation with brackets the line equation reads

$$
\boldsymbol{L} \equiv (0, \boldsymbol{n}) + I(0, \boldsymbol{m}). \qquad (55)
$$

where the $\boldsymbol{n}$ and $\boldsymbol{m}$ are spanned with a 3D bivector basis.

For the equation of the plane we can proceed similarly as for the Eq. (21). We represent the orientation of the plane via the bivector $\boldsymbol{n}$ and the outer product between a bivector touching the plane and its orientation $\boldsymbol{n}$. Since this outer product results in a quatrivector, we can express it as the Hesse distance $d = (\boldsymbol{x} \cdot \boldsymbol{n})$ multiplied by the unit pseudoscalar

$$
\begin{aligned}
\boldsymbol{H} &= \boldsymbol{n} + \boldsymbol{x} \wedge \boldsymbol{n} = \boldsymbol{n} + I(\boldsymbol{x} \cdot \boldsymbol{n}) = \boldsymbol{n} + Id \\
&\equiv (0, \boldsymbol{n}) + I(d, 0). \qquad (56)
\end{aligned}
$$

Note that the plane equation is the dual of the point equation

$$
\boldsymbol{H} = (d + I\boldsymbol{n})^* = (I\boldsymbol{n})^* + (d)^* = \boldsymbol{n} + Id. \qquad (57)
$$

where we consider instead for the plane orientation the unit bivector $\boldsymbol{n}$ and for the scalar 1 the Hesse distance d. Figure 5 presents a comparison of the representations using classical vector calculus, the Euclidean geometric algebra $\mathcal{G}_{3,0,0}$ and the motor algebra $\mathcal{G}^+_{3,0,1}$.

4.6. Motion of Points, Lines and Planes in the 4D Geometric Algebra

The modelling of the 3D motion of the geometric primitives using the motor algebra $\mathcal{G}^+_{3,0,1}$ takes place in a 4D space where rotation and translation are multiplicative operators which are applied as multiplicative operators, as a result the 3D general motion becomes linear. Having a linear method we can then compute for example the unknown rotation and translation simultaneously as we will show in detail in the next sections.

For the modelling of the point motion we use the point representation of Eq. (52) and the motor relations given in Eq. (50) with $I^2 = 0$.

$$
\begin{aligned}
\boldsymbol{X}' &= 1 + I\boldsymbol{x}' = \boldsymbol{M}\boldsymbol{X}\tilde{\bar{\boldsymbol{M}}} = \boldsymbol{M}(1 + I\boldsymbol{x})\tilde{\bar{\boldsymbol{M}}} \\
&= \boldsymbol{T}_s\boldsymbol{R}_s(1 + I\boldsymbol{x})\tilde{\boldsymbol{R}}_s\boldsymbol{T}_s \\
&= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\boldsymbol{R}_s(1 + I\boldsymbol{x})\tilde{\boldsymbol{R}}_s\left(1 + I\frac{\boldsymbol{t}_s}{2}\right) \\
&= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)(1 + I\boldsymbol{R}_s\boldsymbol{x}\tilde{\boldsymbol{R}}_s)\left(1 + I\frac{\boldsymbol{t}_s}{2}\right) \\
&= 1 + I\frac{\boldsymbol{t}_s}{2} + I\boldsymbol{R}_s\boldsymbol{x}\tilde{\boldsymbol{R}}_s + I\frac{\boldsymbol{t}_s}{2} \\
&= 1 + I(\boldsymbol{R}_s\boldsymbol{x}\tilde{\boldsymbol{R}}_s + \boldsymbol{t}_s). \qquad (58)
\end{aligned}
$$

Note that the dual part of this equation in the 4D space is fully equivalent to Eq. (22) in the 3D space.

Using the line Eq. (53) we can express the motion of a line as follows

$$
\begin{aligned}
\boldsymbol{L}' &= \boldsymbol{n}' + I\boldsymbol{m}' = \boldsymbol{M}\boldsymbol{L}\tilde{\boldsymbol{M}} = \boldsymbol{M}(\boldsymbol{n} + I\boldsymbol{m})\tilde{\boldsymbol{M}} \\
&= \boldsymbol{T}_s\boldsymbol{R}_s(\boldsymbol{n} + I\boldsymbol{m})\tilde{\boldsymbol{R}}_s\tilde{\boldsymbol{T}}_s \\
&= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\boldsymbol{R}_s(\boldsymbol{n} + I\boldsymbol{m})\tilde{\boldsymbol{R}}_s\left(1 - I\frac{\boldsymbol{t}_s}{2}\right) \\
&= \left(1 + I\frac{\boldsymbol{t}_s}{2}\right)\left(\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + I\boldsymbol{R}_s\boldsymbol{m}\tilde{\boldsymbol{R}}_s - I\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s\frac{\boldsymbol{t}_s}{2}\right) \\
&= \boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + I\left(\boldsymbol{R}_s\boldsymbol{n}\frac{\boldsymbol{t}_s}{2}\tilde{\boldsymbol{R}}_s + \frac{\boldsymbol{t}_s}{2}\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + \boldsymbol{R}_s\boldsymbol{m}\tilde{\boldsymbol{R}}_s\right) \\
&= \boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + I(\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s{}' + \boldsymbol{R}_s'\boldsymbol{n}\tilde{\boldsymbol{R}}_s + \boldsymbol{R}_s\boldsymbol{m}\tilde{\boldsymbol{R}}_s) \qquad (59)
\end{aligned}
$$

Note that in Eq. (59) before we merge the bivector $\frac{\boldsymbol{t}_s}{2}$ with the rotor $\boldsymbol{R}_s$ or $\tilde{\boldsymbol{R}}_s$ the real and the dual parts are fully equivalent with the elements of the line Eq. (23) of $\mathcal{G}_{3,0,0}$.

The motion of a plane in $\mathcal{G}^+_{3,0,1}$ can be seen as the motion of the dual of the point, thus using the expression of Eq. (56) the motion equation of the plane is

$$
\begin{aligned}
\boldsymbol{H}' &= \boldsymbol{n}' + Id' = \boldsymbol{M}\boldsymbol{H}\tilde{\bar{\boldsymbol{M}}} = \boldsymbol{M}(\boldsymbol{n} + Id)\tilde{\bar{\boldsymbol{M}}} \\
&= \boldsymbol{T}_s\boldsymbol{R}_s(\boldsymbol{n} + Id)\tilde{\boldsymbol{R}}_s\boldsymbol{T}_s \\
&= \left(1 + I\frac{\boldsymbol{t}_s}{\boldsymbol{2}}\right)(\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + Id)\left(1 + I\frac{\boldsymbol{t}_s}{\boldsymbol{2}}\right) \\
&= \boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + I\left(\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s\frac{\boldsymbol{t}_s}{2} + \frac{\boldsymbol{t}_s}{2}\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + d\right) \\
&= \boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s + I(\boldsymbol{t}_s \cdot (\boldsymbol{R}_s\boldsymbol{n}\tilde{\boldsymbol{R}}_s) + d). \qquad (60)
\end{aligned}
$$

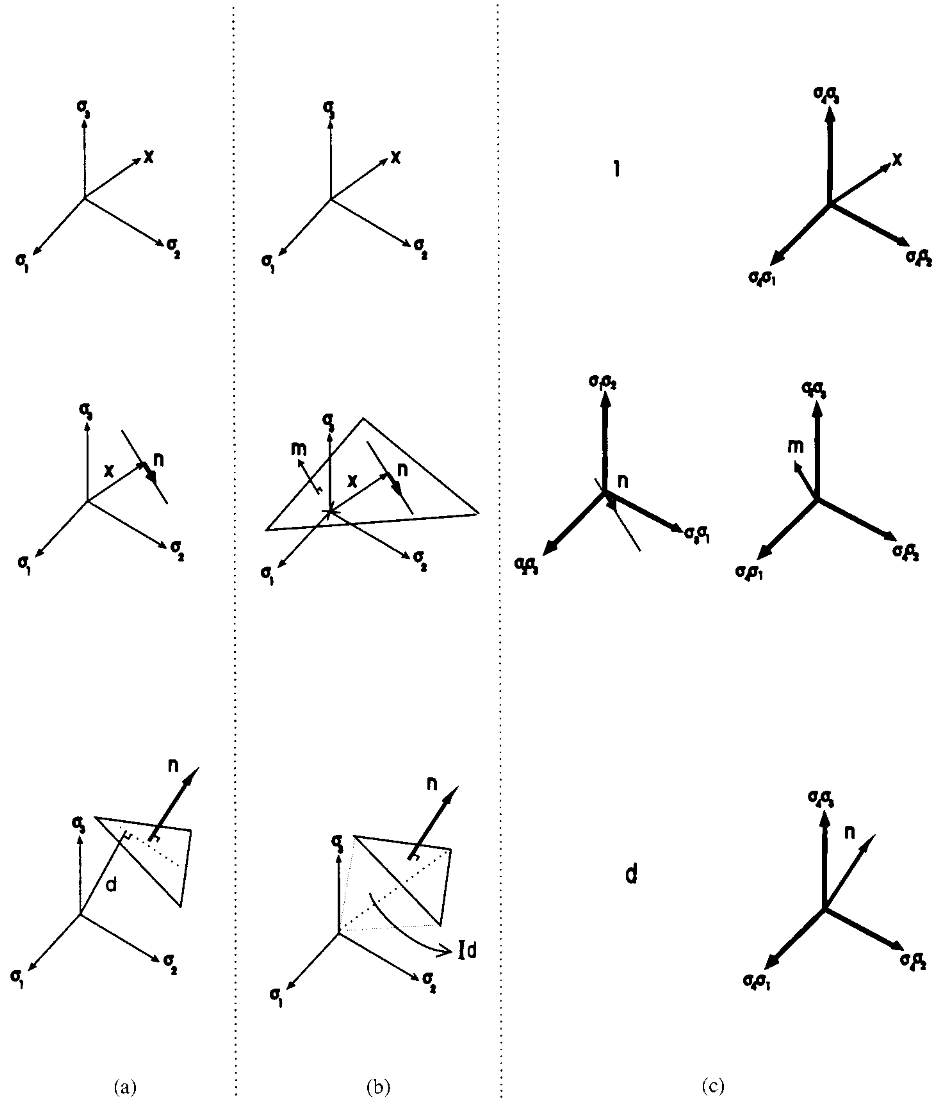


Figure 5. Comparison of representations in the vector calculus, $\mathcal{G}_{3,0,0}$ and $\mathcal{G}_{3,0,1}$ of (a) the point, (b) the line, (c) the plane.

The real part and the dual part of this expression are equivalent in a higher dimension to the bivector and trivector parts of the Eq. (24) in $\mathcal{G}_{3,0,0}$.

5. Case of Study: The Hand-Eye Problem

In this section we want to illustrate the use of the motor algebra for solving an exemplary task of visual guided robotics. We choose the so called hand-eye calibration problem. We find this kind of task in the area of visual guided robotics where cameras are attached to robot arms or mounted on a vehicle and they have to be directed towards a goal. The cameras capture visual cues of the 3D visual space and have their own world reference coordinate system. On the other hand the robot arm or vehicle moves relatively to a reference coordinate system. By computing the intrinsic and extrinsic parameters through the camera movements we can gain the geometrical relationship between the camera positions relative to the world coordinates. The position of the robot arm or vehicle, on the other hand, is always known due to the angular position of the step motors of the device which are permanently controlled by the computer. The problem of hand-eye calibration

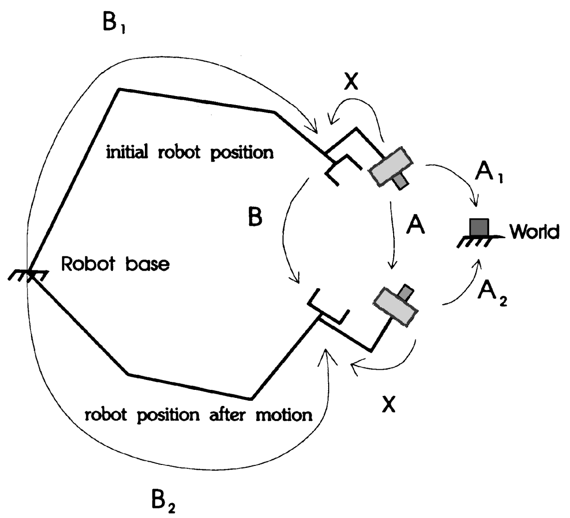


Figure 6. Abstraction of the hand-eye system.

arises when we try to find out the group transformation between the reference coordinates of the mechanical device and the coordinate frame of the camera.

An abstraction of the geometry of a camera mounted on an robot arm is depicted in Fig. 6. The classical way to describe the hand-eye problem in mathematical terms is by using transformation matrices. This problem was firstly formulated as a matrix equation of an Euclidean transformation by Shiu and Ahmad [36] and Tsai and Lenz [39], respectively

$$\mathrm{AX} = \mathrm{XB} \tag{61}$$

where the matrices $\mathrm{A} = \mathrm{A}_1\mathrm{A}_2^{-1}$ and $\mathrm{B} = \mathrm{B}_1\mathrm{B}_2^{-1}$ express the elimination of the transformation between hand–base and world. From the expression (61) the following matrix equation and a vector equation can be derived by splitting the Euclidean transformation into rotation and translation components

$$\mathrm{R}_A\mathrm{R}_X = \mathrm{R}_X\mathrm{R}_B \tag{62}$$

$$(\mathrm{R}_A - \mathrm{I})\mathrm{t}_X = \mathrm{R}_X\mathrm{t}_B - \mathrm{t}_A. \tag{63}$$

In the literature we find a variety of methods to estimate the rotation matrix R_X from Eq. (62), see the survey by Wang [42]. Most of the approaches estimate first the rotation matrix decoupled from the translation. Tsai and Lenz showed that to solve the problem at least two motions are required with rotations having no parallel axes [39]. The most relevant approaches consider for the computation either the axis and the angle of rotation [36, 39], they use quaternions [8] or a canonical matrix representation [25].

Horaud and Dornaika [21] were the first authors who applied a nonlinear method to compute R_X and t_X simultaneously. In their work they showed the instability of the computation of the matrices A_i given the projective matrices $\mathrm{M}_i = \mathrm{SA}_i = \mathrm{SR}_{\mathrm{A}_i}\ \mathrm{St}_{\mathrm{A}_i}$, where S stands for matrix of the the camera intrinsic parameters. Let us assume that the matrix of the intrinsic parameters S remains constant during the camera motions and that one extrinsic calibration A_2 is known. Introducing $\mathrm{N}_i = \mathrm{SR}_{\mathrm{A}_i}$ and $\mathrm{n}_i = \mathrm{St}_{\mathrm{A}_i}$ and replacing $\mathrm{X} = \mathrm{A}_2\mathrm{Y}$, we get now as the hand-eye unknown Y. Thus, the Eq. (61) can be reformulated as

$$\mathrm{A}_2^{-1}\mathrm{A}_1\mathrm{Y} = \mathrm{YB}. \tag{64}$$

Now if $\mathrm{A}_2^{-1}\mathrm{A}_1$ is written as a function of the projection parameters it is possible to get an expression fully independent of the intrinsic parameters S, i.e.

$$\begin{aligned} \mathrm{A}_2^{-1}\mathrm{A}_1 &= \begin{bmatrix} \mathrm{N}_2^{-1}\mathrm{N}_1 & \mathrm{N}_2^{-1}(\mathrm{n}_1 - \mathrm{n}_2) \\ 0_3^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{R} & t \\ 0_3^T & 1 \end{bmatrix}. \end{aligned} \tag{65}$$

Taking into consideration the selected matrices and relations, this result allows to consider the formulation of the hand-eye problem again with the standard Eq. (61) which can be solved with all the known methods and the one presented in this paper.

Other relevant contribution to enlighten the hand-eye problem was made by Chen [6]. According to a geometric point of view he formulated the hand-eye calibration problem using the screw theory. In this attempt the author discovered the important property that the hand-eye transformation is fully independent of the pitch and the angle of the camera and hand motions. The unknown transformation depends simply on the parameters which relate to the screw axis line on the hand motion and the screw axis line of the camera motion. In this paper we will use an unitary motor which is completely isomorph with the unit screw and we will present an algorithm which as opposite to the one of Horaud and Dornaika [21] computes the rotation and the translation simultaneously in a linear manner.

5.1. Solving AX = XB Using the Motor Algebra

In terms of motors the system Eq. (61) can be expressed as follows

$$\boldsymbol{M}_A\boldsymbol{M}_X = \boldsymbol{M}_X\boldsymbol{M}_B \tag{66}$$

or

$$\boldsymbol{M}_A = \boldsymbol{M}_X\boldsymbol{M}_B\tilde{\boldsymbol{M}}_X, \tag{67}$$

where $\boldsymbol{M}_A = \boldsymbol{A} + I\boldsymbol{A}'$, $\boldsymbol{M}_B = \boldsymbol{B} + I\boldsymbol{B}'$ and $\boldsymbol{M}_X = \boldsymbol{R} + I\boldsymbol{R}'$. Next we will simplify this equation to a motor relation between the motor line axis of the camera $\boldsymbol{L}_A$ and the motor line axis of the hand $\boldsymbol{L}_B$.

If we isolate the scalar part of $\boldsymbol{M}_A$ by using the grade operator $\langle\boldsymbol{M}_A\rangle_0 \equiv \langle\boldsymbol{M}_A\rangle$, according Eq. (51) and using the previous equation we get

$$\begin{aligned}\langle\boldsymbol{M}_A\rangle &= \tfrac{1}{2}(\boldsymbol{M}_X\boldsymbol{M}_B\tilde{\boldsymbol{M}}_X + \boldsymbol{M}_X\tilde{\boldsymbol{M}}_B\tilde{\boldsymbol{M}}_X)\\ &= \tfrac{1}{2}\boldsymbol{M}_X(\boldsymbol{M}_B + \tilde{\boldsymbol{M}}_B)\tilde{\boldsymbol{M}}_X = \boldsymbol{M}_X\langle\boldsymbol{M}_B\rangle\tilde{\boldsymbol{M}}_X\\ &= \boldsymbol{M}_X\tilde{\boldsymbol{M}}_X\langle\boldsymbol{M}_B\rangle = \langle\boldsymbol{M}_B\rangle.\end{aligned} \tag{68}$$

Thus, we can equalize the scalar parts of $\boldsymbol{M}_A$ and $\boldsymbol{M}_B$, and regarding Eq. (34) we obtain

$$\cos\left(\frac{\theta_A}{2} + I\frac{d_A}{2}\right) = \cos\left(\frac{\theta_B}{2} + I\frac{d_B}{2}\right) \tag{69}$$

or by separating the real and dual parts

$$\begin{aligned}\cos\left(\frac{\theta_A}{2}\right) &= \cos\left(\frac{\theta_B}{2}\right)\\ d_A\sin\left(\frac{\theta_A}{2}\right) &= d_B\sin\left(\frac{\theta_B}{2}\right).\end{aligned} \tag{70}$$

Taking this fact into account we can be sure that only the bivector parts of $\boldsymbol{M}_A$ and of $\boldsymbol{M}_B$ will contribute to the computation of the unknown $\boldsymbol{M}_X$. Then using Eq. (34) the hand-eye equation reduces to

$$\begin{aligned}&\sin\left(\frac{\theta_A}{2} + I\frac{d_A}{2}\right)\boldsymbol{L}_A\\ &\quad= \boldsymbol{M}_X\left(\sin\left(\frac{\theta_B}{2} + I\frac{d_B}{2}\right)\boldsymbol{L}_B\right)\tilde{\boldsymbol{M}}_X\\ &\quad= \sin\left(\frac{\theta_B}{2} + I\frac{d_B}{2}\right)\boldsymbol{M}_X\boldsymbol{L}_B\tilde{\boldsymbol{M}}_X.\end{aligned} \tag{71}$$

If θ_A and θ_B do not abandon the range from 0 to 360 degrees we can get rid off the sines and get the simplified expression

$$\boldsymbol{L}_A = \boldsymbol{M}_X\boldsymbol{L}_B\tilde{\boldsymbol{M}}_X. \tag{72}$$

This shows that in this kind of problem formulation the rotation and pitch of $\boldsymbol{M}_A$ and $\boldsymbol{M}_B$ are always equal throughout all the hand movements. Thus the consideration of this information can be neglected. It suffices to regard the rotation axes of the involved motors, i.e. the previous equation is reduced to the motion of the line axis of the hand $\boldsymbol{L}_B$ towards the line axis of the camera $\boldsymbol{L}_A$. This property known as the congruence theorem has been pointed out by Chen [7]. However thanks to the use of the motor algebra the proof of this theorem was reduced to one step shown by the Eq. (68). This simplification of the hand-eye problem is depicted in the Fig. 7.

Since the hand-eye problem is a matter of motion of lines the Eq. (59) can be used for the estimation of the unknown 3D transformation which relates to the screw line axes of the hand and of the camera.

$$\begin{aligned}\boldsymbol{L}_A &= \boldsymbol{a} + I\boldsymbol{a}'\\ &= \boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}} + I(\boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}}' + \boldsymbol{R}\boldsymbol{b}'\tilde{\boldsymbol{R}} + \boldsymbol{R}'\boldsymbol{b}\tilde{\boldsymbol{R}}),\end{aligned} \tag{73}$$

where $\boldsymbol{a}$, $\boldsymbol{a}'$, $\boldsymbol{b}$, $\boldsymbol{b}'$ are spanned by bivectors. Separating the real and dual parts

$$\begin{aligned}\boldsymbol{a} &= \boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}}\\ \boldsymbol{a}' &= \boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}}' + \boldsymbol{R}\boldsymbol{b}'\tilde{\boldsymbol{R}} + \boldsymbol{R}'\boldsymbol{b}\tilde{\boldsymbol{R}}\end{aligned} \tag{74}$$

and multiplying from the right with motor $\boldsymbol{R}$ and using the relation $\tilde{\boldsymbol{R}}\boldsymbol{R}' + \tilde{\boldsymbol{R}}'\boldsymbol{R} = 0$ we get the following multivector relations

$$\begin{aligned}\boldsymbol{a}\boldsymbol{R} - \boldsymbol{R}\boldsymbol{b} &= 0\\ (\boldsymbol{a}'\boldsymbol{R} - \boldsymbol{R}\boldsymbol{b}') + (\boldsymbol{a}\boldsymbol{R}' - \boldsymbol{R}'\boldsymbol{b}) &= 0.\end{aligned} \tag{75}$$

These equations can be expressed with a matrix consisting of the scalar and bivector parts and the outer product as follows

$$\begin{bmatrix}\boldsymbol{a} - \boldsymbol{b} & [\boldsymbol{a} + \boldsymbol{b}]_\times & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{a}' - \boldsymbol{b}' & [\boldsymbol{a}' + \boldsymbol{b}']_\times & \boldsymbol{a} - \boldsymbol{b} & [\boldsymbol{a} + \boldsymbol{b}]_\times\end{bmatrix}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{R}'\end{bmatrix} = 0, \tag{76}$$

where the matrix—we will call $\boldsymbol{D}$—is a 6×8 matrix and the vector of unknowns $(\boldsymbol{R}, \boldsymbol{R}')^T$ is 8-dimensional.

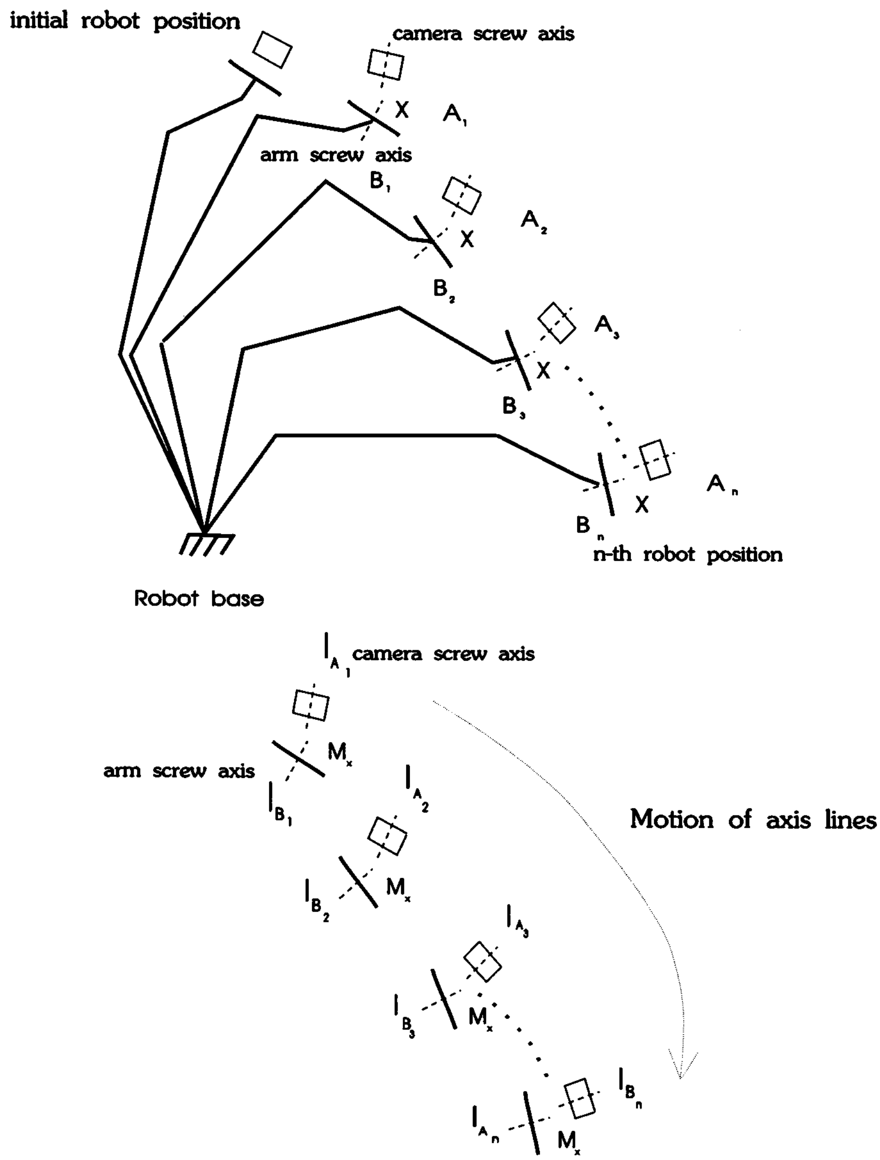


Figure 7. The hand-eye system as the motion of related axes lines.

The notation of $[\boldsymbol{a}+\boldsymbol{b}]_\times$ stands for the vector cross product as an antisymmetric matrix [28]. Recall that we have two constraints on the unknowns so that the result is a unit motor with the properties

$$\boldsymbol{R}\tilde{\boldsymbol{R}} = 1 \quad \text{and} \quad \boldsymbol{R}\boldsymbol{R}' = 0. \tag{77}$$

So we have got six equations and two constraints so far. However, due to the fact that the unit bivectors $\boldsymbol{a}$, $\boldsymbol{b}$ are perpendicular to the bivectors $\boldsymbol{a}'$, $\boldsymbol{b}'$, respectively, we conclude that two equations are necessarily redundant. That is not a surprise really, because it is known that at least two lines are required to estimate 3D motion from their correspondences [35]. As a result at least two motions of the hand-eye system are required in order to compute two lines from the involved screws. Chen [6] clearly noticed this fact and analyzed the uniqueness of the problem. He proved geometrically that even in case of two parallel rotation axis lines it is still possible to compute all the parameters up to the pitch.

5.2. Estimation of the Hand-Eye Motor Using SVD

The hand-eye problem was reduced to the Eq. (76) which depends only of 3D bivectors. Thus we can resort for the estimation to the singular value decomposition method (SVD) [32] which is actually a vector approach for finding singular values. Since we are dealing only with bivectors we can use the SVD method. If we were dealing with a simultaneous estimation of multivectors of different grade of course we should extend the SVD to the multivector concept.

Let us consider that $n \geq 2$ motions are available. Thinking in the singular value decomposition method we build the following $6n \times 8$ matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{D}_1^T & \boldsymbol{D}_2^T & \cdots & \boldsymbol{D}_n^T \end{bmatrix}^T. \tag{78}$$

Due to the fact that the 3D motion has six degrees of freedom this matrix will be in the case of noise-free data at most of rank 6. Let us analyze in more detail the nature of this matrix. Regarding the equations in the noise-free case which were found out basically using geometric and algebraic concepts we should expect that the null-space necessarily contains at least the solution $(\boldsymbol{R}, \boldsymbol{R}')$. The solution $(\mathbf{0}_{4\times1}, \boldsymbol{R})$ ("pure rotation") is the trivial one. Thus, we can reaffirm that the matrix has maximally the rank of six. In the particular case where all the $\boldsymbol{b}$ axis lines were mutually parallel one degree of freedom remains constant thus, the matrix will be of rank 5.

For the solution of the Eq. (78) we use the SVD method. This procedure decomposes the matrix $\boldsymbol{C}$ in three matrices as follows: $\boldsymbol{C} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$. The columns of the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices correspond to the left and right singular vectors, respectively and Σ is a diagonal matrix with singular values. Since the rank of the matrix $\boldsymbol{C}$ is 6 the last two right singular vectors $\boldsymbol{v}_7$ and $\boldsymbol{v}_8$ correspond to the two vanishing singular values which span the null space of $\boldsymbol{C}$. For convenience these will be now expressed in terms of two 4×1 vectors $\boldsymbol{v}_7^T = (\boldsymbol{u}_1, \boldsymbol{v}_1)^T$ and $\boldsymbol{v}_8^T = (\boldsymbol{u}_2, \boldsymbol{v}_2)^T$. Since $(\boldsymbol{R}, \boldsymbol{R}')^T$ is a null vector of $\boldsymbol{C}$, specifically $\boldsymbol{C}(\boldsymbol{R}, \boldsymbol{R}')^T = 0$, then it must be expressed as a linear combination of $\boldsymbol{v}_7$ and $\boldsymbol{v}_8$ as follows

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{R}' \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{v}_1 \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{u}_2 \\ \boldsymbol{v}_2 \end{bmatrix}.$$

Now taking into account the two degrees of freedom imposed by the Eq. (77) we obtain two quadratic equations in α and β:

$$\alpha^2 \boldsymbol{u}_1^T \boldsymbol{u}_1 + 2\alpha\beta \boldsymbol{u}_1^T \boldsymbol{u}_2 + \beta^2 \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \tag{79}$$

$$\alpha^2 \boldsymbol{u}_1^T \boldsymbol{v}_1 + \alpha\beta\big(\boldsymbol{u}_1^T \boldsymbol{v}_2 + \boldsymbol{u}_2^T \boldsymbol{v}_1\big) + \beta^2 \boldsymbol{u}_2^T \boldsymbol{v}_2 = 0 \tag{80}$$

Regarding that $\alpha \neq 0$ and $\beta \neq 0$ and without lost of generality that $\boldsymbol{u}_1^T \boldsymbol{v}_1 \neq 0$ we can set $\mu = \alpha/\beta$ and substitute in Eq. (80) to obtain two solutions for μ. Coming back to Eq. (79) and inserting in it the relation $\alpha = \mu\beta$ we obtain the following quadratic expression

$$\beta^2\big(\mu^2 \boldsymbol{u}_1^T \boldsymbol{u}_1 + \mu\big(2\boldsymbol{u}_1^T \boldsymbol{u}_2\big) + \boldsymbol{u}_1^T \boldsymbol{u}_2\big) = 1 \tag{81}$$

which yields two solutions of opposite sign. This sign variation is simply an effect of sign invariance of the solution. Both, $[\boldsymbol{R}, \boldsymbol{R}']^T$ and $[-\boldsymbol{R}, -\boldsymbol{R}']^T$, satisfy the motion equations and the involved constraints.

Since the equation has μ squared we should also consider the other two solutions. In this case we can see that the second solution for μ causes necessarily the vanishing of the factor in the left hand side of (81). This corresponds to the solution $(\mathbf{0}_{4\times1}, \boldsymbol{R})$ and clearly does not satisfy the first constraint of Eqs. (77). The algorithm can be finally summarized as the following procedure.

1. Consider n hand motions $(\boldsymbol{b}_i, \boldsymbol{b}_i')$ and their corresponding camera motions $(\boldsymbol{a}_i, \boldsymbol{a}_i')$. Check if their scalar parts are equal (Chen's invariance theorem). By extracting the line directions and moments of the screw axes lines construct the matrix $\boldsymbol{C}$ as in (78).
2. Apply the SVD procedure to $\boldsymbol{C}$ and prove if only two singular values are almost equal to zero. In case of noise data we keep the four biggest singular values. Select the related right singular vectors $\boldsymbol{v}_7$ and $\boldsymbol{v}_8$.
3. Setting the coefficients for $\alpha^2 \boldsymbol{u}_1^T \boldsymbol{v}_1 + \alpha\beta(\boldsymbol{u}_1^T \boldsymbol{v}_2 + \boldsymbol{u}_2^T \boldsymbol{v}_1) + \beta^2 \boldsymbol{u}_2^T \boldsymbol{v}_2 = 0$ solve it and find two solutions for μ.
4. Using these two values of μ solve $\beta^2(\mu^2 \boldsymbol{u}_1^T \boldsymbol{u}_1 + \mu(2\boldsymbol{u}_1^T \boldsymbol{u}_2) + \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1$ and select the largest solution to compute α and then β.
5. The final solution will be $\alpha\boldsymbol{v}_7 + \beta\boldsymbol{v}_8$.

6. Experimental Results

This section is devoted to test the new algorithm and to compare its performance with a two-step algorithm similar to the one introduced by Chou and Kamel [8]. These authors estimated the quaternion rotation $\boldsymbol{q}$ direct from the equation $\boldsymbol{a}\boldsymbol{q} = \boldsymbol{q}\boldsymbol{b}$ and then they computed the rotation matrix R_X and solved the translation t_X component using the vector equation (63).

The experiments were carried out using a computer simulation. Firstly, n hand motions (R_b, t_b) were created and Gaussian noise was added with relative standard deviation of 1% simulating the inaccuracy of angle readings. To simulate the hand-eye scenario we generated the camera motions (R_a, t_a) adding similarly Gaussian noise of varying standard deviation. In this case the noise was added as absolute value to the rotation axis direction and as relative value to both the angle and the translation. In order to compute the estimated rotor $\boldsymbol{R}$ and the translation component $\boldsymbol{t}$ between the hand and camera the algorithm was run 1000 times for each value of added noise. The quantification of both algorithms was done according the RMS of the absolute errors in the rotation unit rotor $\|\boldsymbol{R} - \hat{\boldsymbol{R}}\|$ and the RMS of the relative errors in the translation $\|\boldsymbol{t}-\hat{\boldsymbol{t}}\| / \|\boldsymbol{t}\|$.

6.1. First Experiment

In this test a set of 20 hand motions was prepared with quite different rotation axes and large rotation angles and a translation varying within 10 to 20 mm. The Fig. 8 shows the comparison of the results where our algorithm is labeled with MOTOR and the two step algorithm with SEPARATE.

The upper graphic shows the RMS rotation error and the one below the RMS relative to the translation error. The performance of our algorithm is undoubtly better. It is clearly shown that computing the rotation simultaneously with the translation we get much better estimation of the rotation than in the case of a separate computation.

6.2. Second Experiment

Here we wanted to explore the estimation performance by zero translation. As it was expected the behavior of both algorithms is almost the same, see Fig. 9. This effect is easy to explain if we consider Eq. (76). Since the translation is zero the dual parts of the measurements $(\boldsymbol{a}', \boldsymbol{b}')$ become zero. Therefore, the left lower block of the matrix in (76) will disappear which obliges the separate computation of $\boldsymbol{R}$ and $\boldsymbol{R}'$.

6.3. Third Experiment

In the last experiment we were interested in the performance of both algorithms when the noise level is kept constant and the number of motions is gradually increased. Of course in general using more hand and camera motions it is expected to get a much better estimation. The noise level was kept at 5% and the number

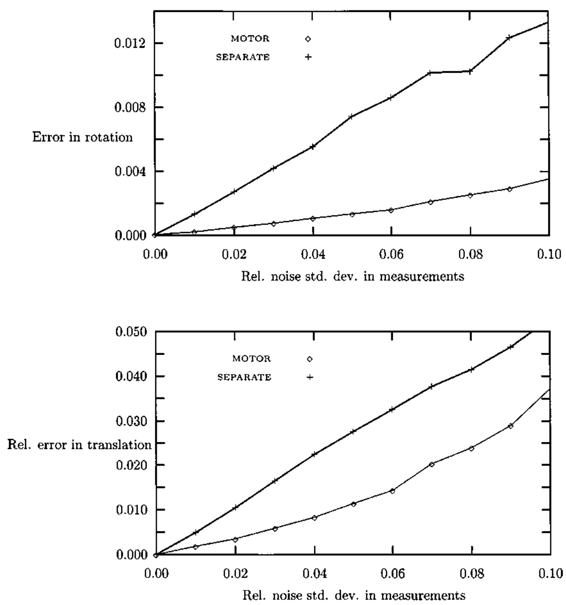


Figure 8. Behavior of the proposed algorithm (MOTOR) and of a two-step algorithm (SEPARATE) with variation of noise.

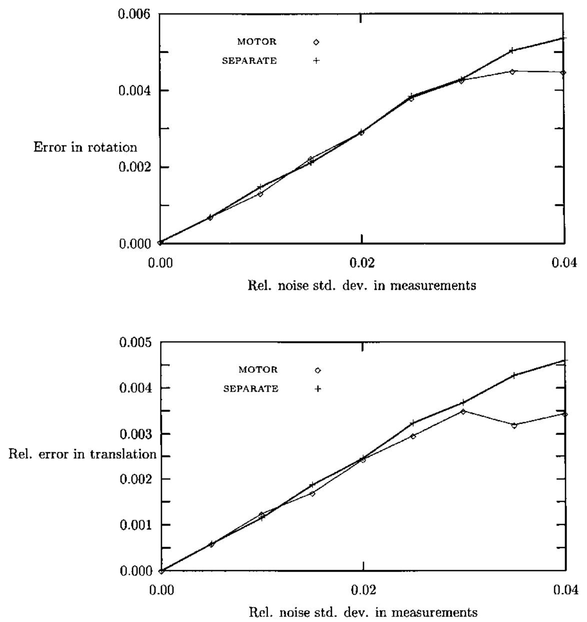


Figure 9. Both algorithms have almost the same performance in absence of translation.

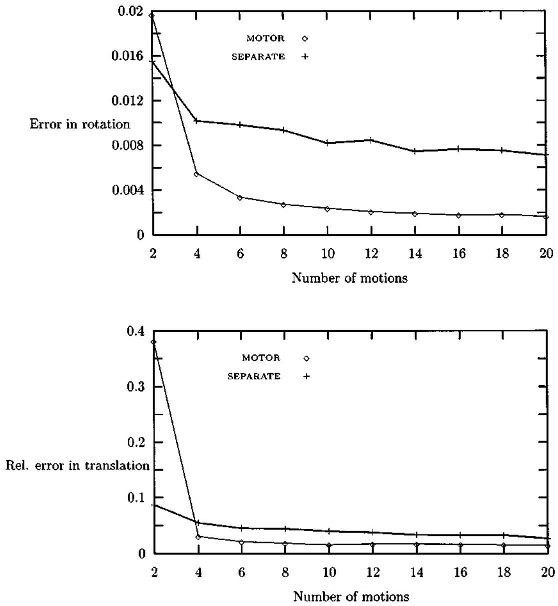


Figure 10. The errors in rotation (upper) and translation (below) as a function of the number of hand and camera motions.

of motions was varied from 2 to 20. The Fig. 10 shows that from the fourth motion onwards our algorithms has a superior performance.

7. Conclusion

This work presents the Clifford or geometric algebra for computations in visually guided robotics. Looking for other suitable ways of representing algebraic relations of geometric primitives we consider the complex and dual numbers in the geometric algebra framework. It turns out that in this framework the algebra of motors is well suited to express the 3D kinematics. Doing that we can linearize the nonlinear 3D rigid motion transformation. In this paper the geometric primitives points, lines and planes are represented using the 3D Euclidean geometric algebra and the 4D motor algebra. Next the rigid motions of these geometric primitives are elegantly expressed using rotors, motors and concepts of duality. In the algebra of motors we extend the 3D Euclidean space representation to a 4D space by means of a dual copy of scalars, vectors and rotors or quaternions.

In the literature it was shown that the invariance of the angle and the pitch of the screws of the camera and hand helps to reduce the complexity of the hand-eye calibration problem. We used this fundamental idea to simplify the hand-eye problem to a problem of motion of lines. For this case we used the algebra of motors which is the one indicated for treating problems involving algebra of lines. The resultant simplified parameterization of the problem enabled us to establish a linear homogeneous system for finding out the motor parameters. The computation of the null-space with SVD and the consideration of the constraints for the dual rotors to be unit helps to get a simple algorithm which avoids non-linear steps. In this work it can be seen that the algebraic structure of the gained equations helps to understand much better the performance of the algorithm.

Acknowledgment

This work has been partially supported by the DFG grant SO-320-2-1 (E. Bayro-Corrochano and G. Sommer).

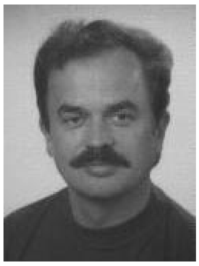
References

1. N.A. Aspragathos and J.K. Dimitros, "A comparative study of three methods for robot kinematics," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, Vol. 28, No. 2, pp. 135–145, 1998.
2. R.S. Ball, *A Treatise on the Theory of Screws*, Cambridge University Press, 1900.
3. E. Bayro-Corrochano, S. Buchholz, and G. Sommer, "Selforganizing Clifford neural network," in *Proceedings of the International Conference of Neural Networks, ICNN'96*, Washington D.C., USA, June 3–6, 1996, Vol. 1, pp. 120–125.
4. E. Bayro-Corrochano, J. Lasenby, and G. Sommer, "Geometric algebra: A framework for computing point and line correspondences and projective structure using *n* uncalibrated cameras," in *IEEE Proceedings of ICPR'96 Viena, Austria*, August, 1996, Vol. I, pp. 334–338.
5. E. Bayro-Corrochano, "The geometry and algebra of kinematics," in *Geometric Computing with Clifford Algebra*, G. Sommer (Ed.), Springer Verlag, 2000.
6. A. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, June 23–25, 1998, pp. 8–15.
7. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *IEEE Conf. Computer Vision and Pattern Recognition*, Maui, Hawaii, June 3–6, 1991, pp. 145–151.
8. D.P. Chevalier, "Lie algebras, modules, dual quaternions and algebraic methods in kinematics," *Mech. Mach. Theory*, Vol. 26, No. 6, pp. 613–627, 1991.
9. J.C.K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Intern. Journal of Robotics Research*, Vol. 10, No. 3, pp. 240–254, 1991.

10. W.K. Clifford, “Preliminary sketch of bi-quaternions,” *Proc. London Math. Soc.*, Vol. 4, pp. 381–395, 1873.
11. W.K. Clifford, “Applications of Grassmann’s extensive algebra,” *Am. J. Math.*, Vol. 1, pp. 350–358, 1878.
12. K. Daniilidis and E. Bayro-Corrochano, “The dual quaternion approach to hand-eye calibration,” in *IEEE Proceedings of the International Conference on Pattern Recognition ICPR’96*, Viena, Austria, August, 1996, Vol. I, pp. 318–322.
13. P.C. Dodwell, “The Lie transformation group model of visual perception,” *Perception and Psychophysics*, Vol. 34, No. 1, pp. 1–16, 1983.
14. J. Funda and R.P. Paul, “A computational analysis of screw transformation in robotics,” *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, pp. 348–356, 1990.
15. Y.L. Gu and J.Y.S. Luh, “Dual-number transformation and its applications to robotics,” *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 6, pp. 615–623, 1987.
16. H. Grassmann, “Der Ort der Hamilton’schen Quaternionen in der Ausdehnungslehre,” *Math. Ann.*, Vol. 12, p. 375, 1877.
17. D. Hestenes, *Space-Time Algebra*, Gordon and Breach, 1966.
18. D. Hestenes, *New Foundations for Classical Mechanics*, D. Reidel: Dordrecht, 1986.
19. D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, D. Reidel: Dordrecht, 1984.
20. D. Hestenes and R. Ziegler, “Projective geometry with Clifford algebra,” *Acta Applicandae Mathematicae*, Vol. 23, pp. 25–63, 1991.
21. W.C. Hoffman, “The Lie algebra of visual perception,” *Journal of Mathematical Psychology*, Vol. 3, pp. 65–98, 1966.
22. R. Horaud and F. Dornaika, “Hand-eye calibration,” *Intern. Journal of Robotics Research*, Vol. 14, pp. 195–210, 1995.
23. I.L. Kantor and A.S. Solodovnikov, *Hypercomplex Numbers: An Elementary Introduction to Algebras*, Springer Verlag: New York Inc., 1989.
24. J.H. Kim and V.R. Kumar, “Kinematics of robot manipulators via line transformations,” *Journal of Robotic Systems*, Vol. 7, No. 4, pp. 649–674, 1990.
25. J. Lasenby, E.J. Bayro-Corrochano, A. Lasenby, and G. Sommer, “A new methodology for computing invariants in computer vision,” in *IEEE Proceedings of the International Conference on Pattern Recognition*, *ICPR’96*, August, 1996, Viena, Austria, Vol. I, pp. 393–397.
26. M. Li and D. Betsis, “Hand-Eye calibration,” in *Proc. Int. Conf. on Computer Vision*, Boston, MA, June 20–23, 1995, pp. 40–46.
27. P. Lounesto, CLICAL User Manual, Research Report A428, Institute of Mathematics, Helsinki University of Technology, 1987.
28. P. Lounesto, *Clifford Algebras and Spinors*, Cambridge University Press: Cambridge, 1997.
29. J.M. McCarthy, “Dual orthogonal matrices in manipulator kinematics,” *International Journal of Robotics Research*, Vol. 5, No. 2, pp. 45–51, 1986.
30. J.M. McCarthy, *Introduction to Theoretical Kinematics*, MIT Press: Massachusetts, 1990.
31. R.M. Murray, L. Zexiang, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press: Florida, 1994.
32. G.R. Pennock and A.T. Yang, “Application of dual-number matrices to the inverse kinematics problem of robot manipulators,” *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, pp. 201–208, 1985.
33. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1994.
34. J. Rooney, “On the three types of complex number and planar transformations,” *Environment and Planning B*, Vol. 5, pp. 89–99, 1978.
35. J. Rooney, “A comparison of representations of general spatial screw displacement,” *Environment and Planning B*, Vol. 5, pp. 45–88, 1978.
36. R. Sabata and J.K. Aggarwal, “Estimation of motion from a pair of range images: A review,” in *CVGIP: Image Understanding*, Vol. 54, pp. 309–324, 1991.
37. Y.C. Shiu and S. Ahmad, “Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$. *IEEE Trans. Robotics and Automation*, Vol. 5, pp. 16–27, 1989.
38. G. Sommer, “The global algebraic frame of the perception-action cycle,” in *Handbook of Computer Vision and Applications*, Vol. III, B. Jähne, H. Haussecker and P. Geissler (Eds.), Academic Press: San Diego, pp. 221–264, 1999.
39. E. Study, *Geometrie der Dynamen*, Teubner Verlag, Leipzig, 1901.
40. R.Y. Tsai and R.K. Lenz, “A new technique for fully autonomous and efficient 3D robotics hand/eye calibration,” *IEEE Trans. Robotics and Automation*, Vol. 5, No. 3, pp. 345–358, 1989.
41. M. Yaglom, *Complex Numbers in Geometry*, Academic Press: Leicester, 1968.
42. M.W. Walker, “Manipulator kinematics and the epsilon algebra,” in *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, pp. 186–192, 1988.
43. C. Wang, “Extrinsic calibration of a vision sensor mounted on a robot,” *IEEE Trans. Robotics and Automation*, Vol. 8, No. 2, pp. 161–175, 1992.

Eduardo 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.

Kostas Daniilidis is Assistant Professor of Computer and Information Science, University of Pennsylvania, affiliated with the interdisciplinary GRASP laboratory. He obtained his Ph.D. in Computer Science from the University Karlsruhe, 1992 and a Diploma in Electrical Engineering from the National Technical University of Athens, 1986. Prior to his current appointment he was with the Cognitive Systems Group, University of Kiel. Nowadays, his research centers on omni-directional vision and vision techniques for tele-immersion and augmented reality. Daniilidis is the chair of the IEEE Workshop on Omnidirectional Vision 2000. He has served as Program Co-Chair of the CAIP-97 conference, as member of the Program Committee at CVPR-98, ICPR-96, ICPR-00, SMILE-00, and he is reviewer in multiple journals.



Gerald Sommer received a diploma degree in physics from Friedrich–Schiller–Universität Jena, Germany, in 1969, a Ph.D. degree in physics from the same university in 1975, and a habilitation degree in engineering from Technical University Illmenau, Germany, in 1988. From 1969 to 1991 he worked at several departments of the Friedrich–Schiller–Universität Jena. From 1991 to 1993 he was the head of the division for medical image processing at the Research Center for Environment and Health (GSF–Medis) in Munich–Neuherberg. Since 1993 he is professor for computer science at the Christian–Albrechts–University Kiel. He is leading the research group cognitive systems. Contemporary his main interests are the design of behavior based systems. The research covers signal theory and signal processing, neural computation for pattern recognition and robot control.