Hand-Eye Calibration in terms of motion of lines using Geometric Algebra

E. Bayro-Corrochano, K. Daniilidis, G. Sommer
Computer Science Institute, Christian Albrechts University,
Preusserstrasse 1-9, 24105, Kiel, Germany.
email: edb,kd,gs@informatik.uni-kiel.d400.de

Abstract

In this paper we will show that the Clifford or geometric algebra is very well suited for the representation and manipulation of geometric objects useful in computer vision and kinematics and also that the computer implementations are straightforward. The power of this approach will be shown by the analysis of the geometry and algebra and optimal solution of the hand-eye calibration problem. The robustness of the algorithm is experimentally compared with classical approaches.

Categories: Computer vision; robotics; Clifford algebra; geometric algebra; rotors; motors; screws; hand-eye calibration.

1 Introduction

Geometric algebra is a coordinate-free approach to geometry based on the algebras of Grassmann and Clifford. The algebra is defined on a space whose elements are called *multivectors*; a multivector is a linear combination of objects of different type, e.g. scalars and vectors. It has an associative and non-commutative product called the **geometric** or **Clifford** product. The existence of such a product and the calculus associated with the geometric algebra give the system tremendous power. For a more complete treatment see [5] and for other brief summary see [1]. Some preliminary applications of geometric algebra in the field of computer vision and neural computing have already been given [1, 2], and here we would like to extend these applications to the robotics field. Firstly rotors and motors and their properties are explained. The next section models the 3-D motion of points, lines and planes useful for computer vision and robotics. It follows the analysis of the hand-eye calibration in geometric algebra terms. Analysis of the uniqueness of the solution and the estimation procedure of the motion is then discussed. Finally the conclusions are given.

2 Geometric Algebra: an outline

The algebras of Clifford and Grassmann are well known to pure mathematicians, but were long ago abandoned by physicists in favour of the vector algebra of Gibbs, which is indeed what is most commonly used today in most areas of physics. The approach to Clifford algebra we adopt here was pioneered in the 1960's by David Hestenes [4] 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.

2.1 Basic Definitions

A particular geometric algebra $\mathcal{G}_{p,q,r}$ can be defined according the amount of its basis elements which square to 1 for p, -1 for q and zero for r, where p+q+r=n. In short $\mathcal{G}_n$ will denote the geometric algebra of n-dimensions - this is a graded linear space. In this paper when r is zero its notation will ignored. As well as vector addition and scalar multiplication we have a non-commutative product which is associative and distributive over addition – this is the **geometric** or **Clifford** product. A further distinguishing feature of the algebra is that any vector squares to give a scalar. The geometric product of 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

$$\boldsymbol{a}\cdot\boldsymbol{b} = \; < \boldsymbol{ab} > \; = \frac{1}{2}(\boldsymbol{ab} + \boldsymbol{ba}) \tag{2}$$

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

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

Thus, $\boldsymbol{b}\wedge\boldsymbol{a}$ will have the opposite orientation making the wedge product anticommutative as given in equation 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 $\boldsymbol{c}$. The outer product of k vectors is a k-vector or k-blade, and such a quantity is said to have *grade* k, see Figure 1.b. A **multivector** (linear combination of objects of different type) is *homogeneous* if it contains terms of only a single grade. The geometric algebra provides a means of manipulating multivectors which allows us to keep track of different grade objects simultaneously – much as one does with complex number operations.

In a space of 3 dimensions we can construct a trivector $\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}$, but 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. The highest grade element in a space is called the **pseudoscalar**. The unit pseudoscalar is denoted by i and is crucial when discussing duality.

In a space of dimension n there are multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), grade 3 (trivectors), etc... up to grade n. Any two such multivectors can be multiplied using the geometric product. Consider two multivectors $\mathbf{A_r}$ and $\mathbf{B_s}$ of grades $\mathbf{r}$ and $\mathbf{s}$ respectively. The geometric product of $\mathbf{A_r}$ and $\mathbf{B_s}$ can be written as

$$\mathbf{A_r B_s} = \langle\mathbf{AB}\rangle_{\mathbf{r+s}} + \langle\mathbf{AB}\rangle_{\mathbf{r+s-2}} + \ldots + \langle\mathbf{AB}\rangle_{|\mathbf{r-s}|} \tag{4}$$

where $\langle\mathbf{M}\rangle_{\mathbf{t}}$ is used to denote the t-grade part of multivector $\mathbf{M}$, e.g. $\langle\mathbf{ab}\rangle = \langle\mathbf{ab}\rangle_{\mathbf{0}} + \langle\mathbf{ab}\rangle_{\mathbf{2}} = \mathbf{a}\cdot\mathbf{b}+\mathbf{a}\wedge\mathbf{b}$. In the following sections expressions of grade 0 will be written ignoring their subindex, i.e. $\langle\mathbf{ab}\rangle_{\mathbf{0}} = \langle\mathbf{ab}\rangle = \mathbf{a}\cdot\mathbf{b}$.

2.2 The Geometric Algebra of 3-D Space

In an n-dimensional space we can introduce an orthonormal basis of vectors $\{\sigma_i\}$ $i = 1, ..., n$, such that $\sigma_i\cdot\sigma_j = \delta_{ij}$. This leads to a basis for the entire algebra:

$$1, \quad \{\sigma_i\}, \quad \{\sigma_i\wedge\sigma_j\}, \quad \{\sigma_i\wedge\sigma_j\wedge\sigma_k\}, \quad \ldots, \quad \sigma_1\wedge\sigma_2\wedge\ldots\wedge\sigma_n. \tag{5}$$

Note that we shall not use bold symbols for these basis vectors. Any multivector can be expressed in terms of this basis.

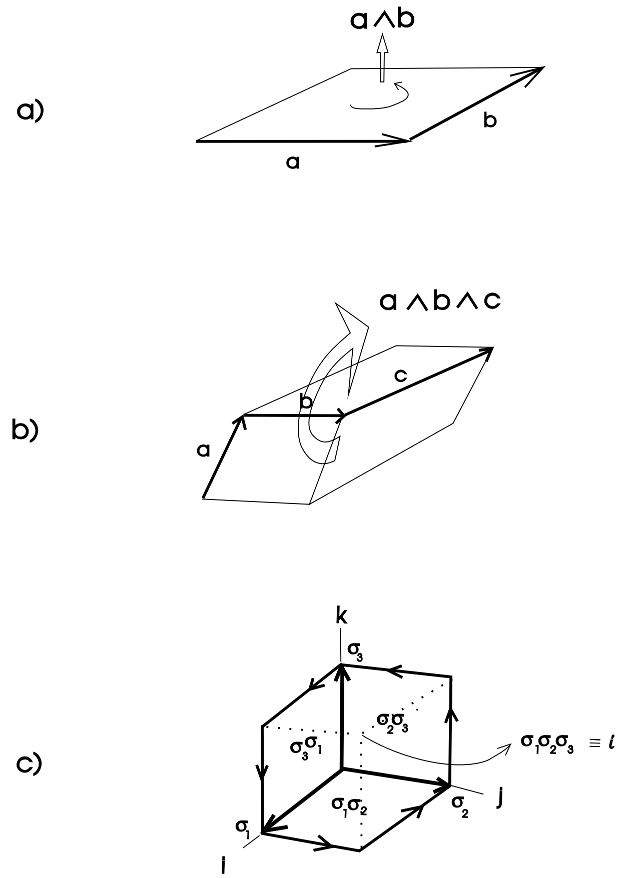

Figure 1: a) A bivector b) A trivector c) 3-D Basis .

The geometric algebra $\mathcal{G}_{3,0}$ for the 3-D space has $2^3 = 8$ elements given by:

$$\underbrace{1}_{scalar}, \underbrace{\{\sigma_1, \sigma_2, \sigma_3\}}_{vectors}, \underbrace{\{\sigma_1\sigma_2, \sigma_2\sigma_3, \sigma_3\sigma_1\}}_{bivectors}, \underbrace{\{\sigma_1\sigma_2\sigma_3\} \equiv i}_{trivector}. \tag{6}$$

It can easily be verified that the trivector or pseudoscalar $\sigma_1\sigma_2\sigma_3$ squares to -1 and commutes with all multivectors in the 3-D space. We therefore give it the symbol i; noting that this is not the uninterpreted commutative scalar imaginary j used in quantum mechanics and engineering.

2.2.1 Rotors

Multiplication of the three basis vectors σ_1, σ_2 and σ_3 by i results in the three basis bivectors $\sigma_1\sigma_2 = i\sigma_3 \quad \sigma_2\sigma_3 = i\sigma_1 \quad \sigma_3\sigma_1 = i\sigma_2$. 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 $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ of the quaternion algebra with $i\sigma_1, -i\sigma_2, i\sigma_3$, we see that the famous Hamilton relations are recovered $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. 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 3-D rotations, see Figure 1.c.

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

$$\boldsymbol{R} = \underbrace{a_0}_{scalar} + \underbrace{a_1(i\sigma_1) - a_2(i\sigma_2) + a_3(i\sigma_3)}_{bivectors}. \tag{7}$$

The quaternion algebra is therefore seen to be a subset of the geometric algebra of 3-space.
Any rotation can be formed 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 $\boldsymbol{a}_{-} - \boldsymbol{a}_{||} = -\boldsymbol{nan}$, where $\boldsymbol{a}_{-}$ and $\boldsymbol{a}_{||}$ respectively denote parts 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 a unit vector $\boldsymbol{m}$ results in a new vector $-\boldsymbol{m}(-\boldsymbol{nan})\boldsymbol{m} = (\boldsymbol{mn})\boldsymbol{a}(\boldsymbol{nm}) = \boldsymbol{Ra}\tilde{\mathbf{R}}$. We show now using the geometric product that the rotor $\boldsymbol{R}$ of equation 7 is a multivector consisting of a scalar and a bivector parts, i.e. $\boldsymbol{R} = \boldsymbol{mn} = \boldsymbol{m}\cdot\boldsymbol{n} + \boldsymbol{m}\wedge\boldsymbol{n}$. These parts correspond to the scalar and vector parts of an equivalent quaternion. Rotors combine in a straightforward manner, i.e. a rotation $\boldsymbol{R}_1$ followed by a rotation $\boldsymbol{R}_2$ is equivalent to an overall rotation $\boldsymbol{R}$ where $\boldsymbol{R} = \boldsymbol{R}_2\boldsymbol{R}_1$. The transformation $\boldsymbol{a} \mapsto \mathbf{R}\boldsymbol{a}\tilde{\mathbf{R}}$ 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.

2.3 The complex and the dual numbers in geometric algebra

The complex, double and dual numbers and represented them as a composed number $\boldsymbol{a} = \boldsymbol{b} + \omega\boldsymbol{c}$ using the algebraic operator ω which in case of the complex numbers $\omega^2 = -1$, the double numbers $\omega^2 = 1$ and the dual numbers $\omega^2 = 0$. In case of the dual numbers the term $\boldsymbol{b}$ is called the real part and $\boldsymbol{c}$ the dual part. In the famous paper *Preliminary sketch* of *biquaternions* [3] Clifford introduced the motors or biquaternions for representing screw motion. Later on Study [9] used the dual numbers to represent the relative position of two skew lines in space, i.e. $\hat{\theta} = \theta + \omega d$. The real part indicates the difference of the line orientation angles and the dual part the distance between both lines.
This paper uses dual numbers and only for comparison purposes the complex numbers are treated in the next section in some detail.

2.3.1 Octonions

Octonions is an example of complex numbers we find in the geometric algebra $\mathcal{G}_{1,3,0}$ for the 4-space with the basis

$$\underbrace{1}_{scalar}, \quad \underbrace{\gamma_k}_{4 \quad vectors}, \quad \underbrace{\gamma_4\gamma_l, i\gamma_4\gamma_l}_{6 \quad bivectors}, \quad \underbrace{i\gamma_k}_{4 \quad pseudovectors}, \quad \underbrace{i}_{1 \quad pseudoscalar} \tag{8}$$

where $\gamma_4^2 = +1$, $\gamma_l^2 = -1$ and $\gamma_4\gamma_l$ for l=1,2,3. The pseudoscalar is $i = \gamma_1\gamma_2\gamma_3\gamma_4$ with $i^2 = -1$. We can represent an octonion combining two rotors or quaternions the last expressed in terms of bivectors, i.e.

$$\boldsymbol{O} = R + iR' = (a_0 + a_1\gamma_4\gamma_1 - a_2\gamma_4\gamma_2 + a_3\gamma_4\gamma_3) + i(b_0 + b_1\gamma_4\gamma_1 - b_2\gamma_4\gamma_2 + b_3\gamma_4\gamma_3). \tag{9}$$

Note that the pseudoscalar i which squares to -1 is used as operator now instead of ω. It is no need to resort to an algebraic operator like the ω which has not a geometric interpretation. The octonion algebra is the even subalgebra $\mathcal{G}^0_{1,3,0}$ of the geometric algebra $\mathcal{G}_{1,3,0}$ of the 4-space, with the basis

$$\underbrace{1}_{scalar}\ ,\ \underbrace{\gamma_4\gamma_l, i\gamma_4\gamma_l}_{6\ \ bivectors}\ ,\ \underbrace{i}_{1\ \ pseudoscalar} \tag{10}$$

where $i^2 = -1$. Other way to see an octonions is as the result of the doubling procedure [6], which tells that doubling a complex number you get a quaternion and doubling a quaternion you get an octonion. An octonion as the geometric operator rotates geometric objects in 4-D.

2.3.2 Motors

Now we will discuss the motors expressed in terms of the dual sum of two rotors [3]. As we sad the necessary condition for dual numbers is that $i^2 = 0$, thus we require a geometric algebra for the 4-D space where one of its basis vectors squares to zero. This is the $\mathcal{G}_{0,3,1}$ with $\gamma_4^2 = 0$, $\gamma_k^2 = -1$, $k = 1, 2, 3$ and pseudoscalar $i = \gamma_1\gamma_2\gamma_3\gamma_4$ which thus squares to zero $i^2 = 0$. The 16 basis elements differ with the basis of $\mathcal{G}_{1,3,0}$ only in that γ_4 and i square to zero. The expression for the dual rotor or motor is similar to the equation 9 only now with the condition i^2=0. The dual rotors require the even subalgebra $\mathcal{G}^0_{0,3,1}$ of the geometric algebra $\mathcal{G}_{0,3,1}$, i.e.

$$\underbrace{1}_{scalar}\ ,\ \underbrace{\gamma_4\gamma_l, i\gamma_4\gamma_l}_{6\ \ bivectors}\ ,\ \underbrace{i}_{1\ \ pseudoscalar} \tag{11}$$

where $i^2 = 0$. Clifford introduced the biquaternions with the name motors which is the abbreviation of "moment and vector"[3]. The basic geometric interpretation of a motor $\boldsymbol{M}$ can be given using two non-coplanar lines, which can be expressed in terms of bivector basis as follows

$$\begin{aligned} \boldsymbol{M} &= \mathbf{R} + i\mathbf{R}' = \boldsymbol{X}_1\boldsymbol{X}_2 + \boldsymbol{X}_3\boldsymbol{X}_4 = (\boldsymbol{X}_1\cdot\boldsymbol{X}_2 + \boldsymbol{X}_1\wedge\boldsymbol{X}_2) + (\boldsymbol{X}_3\cdot\boldsymbol{X}_4 + \boldsymbol{X}_3\wedge\boldsymbol{X}_4) \\ &= (a_0 + a_1\gamma_4\gamma_1 - a_2\gamma_4\gamma_2 + a_3\gamma_4\gamma_3) + i(b_0 + b_1\gamma_4\gamma_1 - b_2\gamma_4\gamma_2 + b_3\gamma_4\gamma_3) \end{aligned} \tag{12}$$

Note that lines expressed in terms of bivectors can be added. So we can see a motor also as bivector. If the lines are not coplanar gives again a bivector or motor, whereas if the lines are coplanar the resultant line can be seen as a *degenerated* motor.
A motor is different than an octonion, it represents a general displacement or rigid motion and it is exact equivalent to an screw [3]. It will be more convenient if the translation is expressed as a sort of a rotor or translator $\boldsymbol{T}$, i.e.

$$\boldsymbol{M} = \mathbf{R} + i\mathbf{R}' = \boldsymbol{R} + i\frac{\boldsymbol{t}}{2}\boldsymbol{R} = (1 + i\frac{\boldsymbol{t}}{2})\boldsymbol{R} = \boldsymbol{T}\boldsymbol{R}. \tag{13}$$

The translator can be seen simply as the representation of a rotation plane displaced from the reference origin by $\boldsymbol{t}$ and with the same orientation of the vector $\boldsymbol{t}$. The vector $\boldsymbol{t}$ can be expressed in terms of the rotors using

$$\boldsymbol{R}'\tilde{\boldsymbol{R}} = \frac{\boldsymbol{t}}{2}\boldsymbol{R}\tilde{\boldsymbol{R}} \tag{14}$$

therefore

$$\boldsymbol{t} = 2\boldsymbol{R}'\tilde{\boldsymbol{R}} \tag{15}$$

where the multiplication is a geometric product.
The absolute value of a motor $\boldsymbol{M}$ is computed as follows

$$\boldsymbol{M}\tilde{\boldsymbol{M}} = \boldsymbol{T}\boldsymbol{R}\tilde{\boldsymbol{R}}\tilde{\boldsymbol{T}} = (1 + i\frac{\boldsymbol{t}}{2})\boldsymbol{R}\tilde{\boldsymbol{R}}(1 - i\frac{\boldsymbol{t}}{2}) = \mathbf{I} + i\frac{\boldsymbol{t}}{2} - i\frac{\boldsymbol{t}}{2} = \mathbf{I}, \tag{16}$$

where $\mathbf{I}$ is the identity. The combination of two rigid motions can be expressed using two motors. The resultant motor describes the overall displacement, namely

$$M_c = M_a M_b = (\boldsymbol{R}_a + i\boldsymbol{R}'_a)(\boldsymbol{R}_b + i\boldsymbol{R}'_b) = \boldsymbol{R}_a\boldsymbol{R}_b + i(\boldsymbol{R}_a\boldsymbol{R}'_b + \boldsymbol{R}'_a\boldsymbol{R}_b) = \boldsymbol{R}_c + i\boldsymbol{R}'_c \tag{17}$$

Note that rotations combines multiplicatively and in the dual part the translations additively.

2.4 Representation of the point, line and plane using dual numbers

This section introduces the representation of points, lines and planes in the framework of the sub algebra $\mathcal{G}^0_{0,3,1}$ of motors. A point in the $\mathcal{G}_{0,3,1}$ or 4-D space is

$$\boldsymbol{X} = X_1\gamma_1 + X_2\gamma_2 + X_3\gamma_3 + X_4\gamma_4 \tag{18}$$

if we want to express as a dual number using only bivectors basis we apply the geometric product with the projective split [7] γ_4 and divide by the four coordinate coefficient

$$\begin{aligned} \frac{\gamma_4}{X_4}\boldsymbol{X} &= 1 + \frac{X_1}{X_4}\gamma_4\gamma_1 + \frac{X_2}{X_4}\gamma_4\gamma_2 + \frac{X_3}{X_4}\gamma_4\gamma_3 \\ &= 1 + i(x_1\gamma_2\gamma_3 + x_2\gamma_3\gamma_1 + x_3\gamma_1\gamma_2) \\ \boldsymbol{x}_d &= 1 + i\boldsymbol{x}. \end{aligned} \tag{19}$$

The $\boldsymbol{x}$ vector expressed in terms of bivectors corresponds to the 3-D point expression. A line can be seen as a degenerated motor, setting a_0 and b_0 to zero in the equation 12 we get straightforward the dual line equation in terms of bivector basis, namely

$$\begin{aligned} \boldsymbol{l}_d &= (L^{41}\gamma_4\gamma_1 + L^{42}\gamma_4\gamma_2 + L^{43}\gamma_4\gamma_3) + (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2) \\ &= (L^{41}\gamma_4\gamma_1 + L^{42}\gamma_4\gamma_2 + L^{43}\gamma_4\gamma_3) + i(L^{23}\gamma_4\gamma_1 + L^{31}\gamma_4\gamma_2 + L^{21}\gamma_4\gamma_3) \end{aligned} \tag{20}$$

Note that this is equivalent to the line expression using Plücker coordinates. The real part can be seen as the line direction and the dual part as the moment which is nothing else as the outer product between $\boldsymbol{n}$ and any vector $\boldsymbol{p}$ touching the line, i.e.

$$l_d = \boldsymbol{n} + i\boldsymbol{n}\wedge\boldsymbol{p} = \boldsymbol{n} + i\boldsymbol{m}. \tag{21}$$

This line representation using dual numbers is easier to understand and to manipulate algebraically than the one in terms of Plücker coordinates.

In 4-D, the dual geometric object of a point is a plane which can be represented in terms of the dual of the vector basis, i.e. the trivector basis as follows

$$\Phi = X'_1\gamma_4\gamma_1\gamma_2 + X'_2\gamma_4\gamma_2\gamma_3 + X'_3\gamma_4\gamma_3\gamma_1 + X'_4\gamma_1\gamma_2\gamma_3, \tag{22}$$

now if we apply the geometric product with γ_4 to the left and divide by X_4

$$\begin{aligned} \frac{\gamma_4}{X_4}\Phi &= \frac{X'_1}{X_4}\gamma_4\gamma_4\gamma_1\gamma_2 + \frac{X'_2}{X_4}\gamma_4\gamma_4\gamma_2\gamma_3 + \frac{X'_3}{X_4}\gamma_4\gamma_4\gamma_3\gamma_1 + \frac{X'_4}{X_4}\gamma_4\gamma_3\gamma_2\gamma_1 \\ &= \frac{X'_1}{X_4}\gamma_1\gamma_2 + \frac{X'_2}{X_4}\gamma_2\gamma_3 + \frac{X'_3}{X_4}\gamma_3\gamma_1 + \frac{X'_4}{X_4}\gamma_4\gamma_3\gamma_2\gamma_1 \\ &= x'_1\gamma_1\gamma_2 + x'_2\gamma_2\gamma_3 + x'_3\gamma_3\gamma_1 + id \\ \boldsymbol{p}_d &= \boldsymbol{x}' + id \end{aligned} \tag{23}$$

Note the dual part is a constant and the real part a vector, the opposite as in the case of the expression for the point.

Note that the use of the projective split γ_4 helps to map geometric objects of 4-D space to the 3-D subspace [7]. Vectors, bivectors and trivectors in 4-D will represent points, lines and planes in 3-D. As we choose γ_4 as a selected direction in 4-D, we define a mapping which associates the bivectors $\gamma_4\gamma_i$, $i = 1, 2, 3$, in 4-D with the vectors σ_i, $i = 1, 2, 3$, in 3-D;

$$\sigma_1 \equiv \gamma_4\gamma_1 \quad \sigma_2 \equiv \gamma_4\gamma_2 \quad \sigma_3 \equiv \gamma_4\gamma_2. \tag{24}$$

That is why we can say that a 3-D point is

$$\boldsymbol{x} = x_1\sigma_1 + x_2\sigma_2 + x_3\sigma_3 \equiv \frac{X_1}{X_4}\gamma_4\gamma_1 + \frac{X_2}{X_4}\gamma_4\gamma_2 + \frac{X_3}{X_4}\gamma_4\gamma_3 \tag{25}$$

and a 3-D line

$$\boldsymbol{l} = l_1\sigma_1 + l_2\sigma_2 + l_3\sigma_3 \equiv (L^{41}\gamma_4\gamma_1 + L^{42}\gamma_4\gamma_2 + L^{43}\gamma_4\gamma_3) \tag{26}$$

Note that after the mapping the only information that remains of the 4-D line for the line in 3-D is its orientation.
The dual expressions for the point, line and plane are now ready for the modelling of their motion using motors. We will show that this kind of modelling is very useful when we deal with real problems like the hand-eye calibration problem.

3 Modelling the 3-D Motion of Points, Lines and Planes using dual numbers

The reason why we are interested to model 3-D motion using motors as opposite to the rotor based modelling is that with the former approach we can compute in case of the hand-eye problem the rotation and translation of the unknown rigid motion simultaneously. In case of the rotor approach we are unfortunately compelled to compute the translation decouple of rotation increasing therefore the inaccuracy. This will be shown in detail in the next sections. In this section we will present the 3-D motion modelling using dual numbers but also using vectors and rotors for comparison purposes.
The 3-D motion of a point $\boldsymbol{x}$ in $\mathcal{G}_{3,0}$ has the equation

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}}' + \boldsymbol{t}. \tag{27}$$

In case of $\mathcal{G}^0_{0,3,1}$ we use the point representation of equation (19)

$$\begin{aligned} \boldsymbol{M}(1 + i\boldsymbol{x})\overline{\tilde{\boldsymbol{M}}} &= \boldsymbol{M}(1)\overline{\tilde{\boldsymbol{M}}} + i\boldsymbol{M}\boldsymbol{x}\overline{\tilde{\boldsymbol{M}}} \\ &= 1 + i(\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}}' + \boldsymbol{t}) \end{aligned} \tag{28}$$

where $\overline{\tilde{\boldsymbol{M}}} = \tilde{\boldsymbol{R}} - i\tilde{\boldsymbol{R}}'$. In $\mathcal{G}_{3,0}$ a line can be described in terms of any couple of points lying on the line, i.e. $\boldsymbol{x} = \theta\boldsymbol{p}_1 + \boldsymbol{p}_2$. The motion equation of the line is then the same as for the point equation equation (28). In $\mathcal{G}^0_{0,3,1}$ we expressed the line as equation equation (21) and proceed as before

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{l}_b + i\boldsymbol{m}_b)\tilde{\boldsymbol{M}} &= (1 + i\frac{\boldsymbol{t}}{\boldsymbol{2}})\boldsymbol{R}(\boldsymbol{l}_b + i\boldsymbol{m}_b)\tilde{\boldsymbol{R}}(1 - i\frac{\boldsymbol{t}}{2}) \\ &= (1 + i\frac{\boldsymbol{t}}{\boldsymbol{2}})(\boldsymbol{R}\boldsymbol{l}_b\tilde{\boldsymbol{R}} + i\boldsymbol{R}\boldsymbol{m}_b\tilde{\boldsymbol{R}} - i\boldsymbol{R}\boldsymbol{l}_b\tilde{\boldsymbol{R}}\frac{\boldsymbol{t}}{2}) \\ \boldsymbol{l}_a + i\boldsymbol{m}_a &= \boldsymbol{R}\boldsymbol{l}_b\tilde{\boldsymbol{R}} + i(\boldsymbol{R}\boldsymbol{l}_b\tilde{\boldsymbol{R}}' + \boldsymbol{R}'\boldsymbol{l}_b\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{m}_b\tilde{\boldsymbol{R}}) \end{aligned} \tag{29}$$

where $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{R}} + i\tilde{\boldsymbol{R}}'$. For the plane in $\mathcal{G}_{3,0}$ we use a multivector representation of the formula of Hesse, i.e. $\boldsymbol{H} = d + \boldsymbol{n}$. Note that this multivector consists of a scalar and a vector. Any point lying on this plane fulfills $\boldsymbol{x}\cdot\boldsymbol{n} - d{=}0$. Using this we can now write the motion of the plane

$$\boldsymbol{H}' = (\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t})\cdot(\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}) + (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}). \tag{30}$$

Since $(\boldsymbol{Rx}\tilde{\boldsymbol{R}})\cdot(\boldsymbol{Rn}\tilde{\boldsymbol{R}}) = \boldsymbol{x}\cdot\boldsymbol{n}$, this becomes $\boldsymbol{H}' = \boldsymbol{x}\cdot\boldsymbol{n} + \boldsymbol{Rn}\tilde{\boldsymbol{R}} + \boldsymbol{t}\cdot(\boldsymbol{Rn}\tilde{\boldsymbol{R}})$ which can be finally written as

$$\boldsymbol{H}' = \boldsymbol{RH}\tilde{\boldsymbol{R}} + < \boldsymbol{RH}\tilde{\boldsymbol{R}}\boldsymbol{t} > . \tag{31}$$

The motion of a plane in $\mathcal{G}^0_{0,3,1}$ can be seen as the motion of the dual of the point, thus using the expression equation (23) the motion equation of the plane is

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{n}+id)\widetilde{\boldsymbol{M}} &= \boldsymbol{Rn}\tilde{\boldsymbol{R}} + i(-\boldsymbol{Rn}\tilde{\boldsymbol{R}}\frac{\boldsymbol{t}}{2} + \frac{\boldsymbol{t}}{2}\boldsymbol{Rn}\tilde{\boldsymbol{R}}) + id = \boldsymbol{Rn}\tilde{\boldsymbol{R}} + i(d - \boldsymbol{Rn}\tilde{R}\boldsymbol{t}) \\ &= \boldsymbol{Rn}\tilde{\boldsymbol{R}} + i(d - < \boldsymbol{RnRt} >). \end{aligned} \tag{32}$$

4 The Hand-Eye Problem

The well known hand-eye equation firstly formulated by Shiu and Ahmad [8] and Tsai and Lenz [10] reads

$$\mathbf{AX} = \mathbf{XB} \tag{33}$$

where $\mathbf{A} = \mathbf{A}_1\mathbf{A}_2^{-1}$ and $\mathbf{B} = \mathbf{B}_1\mathbf{B}_2^{-1}$ express the elimination of the transformation hand-base to world. From the expression equation (33) the following matrix and a vector equations can be derived $\mathbf{R_A R_X} = \mathbf{R_X R_B}$ and $(\mathbf{R}_A - \mathbf{I})\vec{\mathbf{t}}_X = \mathbf{R}_X\vec{\mathbf{t}}_B - \vec{\mathbf{t}}_A$. Most of the approaches estimate first the rotation matrix decoupled from the translation [10, 13]. The problem requires at least two motions with rotations having not parallel axes [10]. Horaud and Dornaika [11] showed the instability of the computation of the $\mathbf{A_i}$ matrices given the projective matrices $\mathbf{M}_i = \mathbf{C}\mathbf{A}_i = (\mathbf{C}\mathbf{R}_{A_i}\mathbf{C}\vec{\mathbf{t}}_{A_i})$. Let us assume that the matrix of the intrinsic parameters $\mathbf{C}$ remains constant during the motions and that one extrinsic calibration $\mathbf{A_2}$ is known. Introducing $\mathbf{N}_i = \mathbf{C}\mathbf{R}_{A_i}$ and $\vec{\mathbf{n}}_i = \mathbf{C}\vec{\mathbf{t}}_{A_i}$ and replacing $\mathbf{X}{=}\mathbf{A_2 Y}$, we get now as the hand-eye unknown $\mathbf{Y}$. Thus the equation equation (33) can be reformulated as $\mathbf{A}_2^{-1}\mathbf{A}_1\mathbf{Y} = \mathbf{YB}$. Now if $\mathbf{A}_2^{-1}\mathbf{A}_1$ is written as a function of the projection parameters it is possible to get an expression fully independent of the intrinsic parameters $\mathbf{C}$, i.e.

$$\mathbf{A}_2^{-1}\mathbf{A}_1 = \begin{pmatrix} \mathbf{N}_2^{-1}\mathbf{N}_1 & \mathbf{N}_2^{-1}(\vec{\mathbf{n}}_1 - \vec{\mathbf{n}}_2) \\ 0^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{pmatrix}. \tag{34}$$

Taking into consideration the selected matrices and relations, this result allows anyway to consider the formulation of the hand-eye problem again with the standard equation equation (33) which can be solved with all the known methods and the one presented in this paper.

4.1 Solving AX=XB using motors

The equation system equation (33) can be expressed in terms of motors as

$$\boldsymbol{M}_A\boldsymbol{M}_X = \boldsymbol{M}_X\boldsymbol{M}_B \tag{35}$$

where $\boldsymbol{M}_A = \boldsymbol{A} + i\boldsymbol{A}'$, $\boldsymbol{M}_B = \boldsymbol{B} + i\boldsymbol{B}'$ and $\boldsymbol{M}_X = \boldsymbol{R} + i\boldsymbol{R}'$. According the congruence theorem of Chen [12] in this kind of problem the rotation and pitch of $\boldsymbol{M}_A$ and $\boldsymbol{M}_B$ are always equal through out all the hand movements. Thus it is redundant the consideration of this information. It suffices to regard the rotation axis of the involved motors, i.e. the previous equation is reduced as the motion of the axis line of the hand towards the axis line of the camera. For that we can use the equation equation (29) for the computation of the real and dual components of $\boldsymbol{l}_A$, i.e.

$$\boldsymbol{l}_A = \boldsymbol{a} + i\boldsymbol{a}' = \boldsymbol{Rb}\tilde{\boldsymbol{R}} + i(\boldsymbol{Rb}\tilde{\boldsymbol{R}}' + \boldsymbol{Rb}'\tilde{\boldsymbol{R}} + \boldsymbol{R}'\boldsymbol{b}\tilde{\boldsymbol{R}}). \tag{36}$$

After some simple manipulations according the relation $\tilde{\boldsymbol{R}}\boldsymbol{R}'+\tilde{\boldsymbol{R}}'\boldsymbol{R}=0$ we get the following matrix vector equation

$$\begin{pmatrix} \vec{\boldsymbol{a}}-\vec{\boldsymbol{b}} & [\vec{\boldsymbol{a}}+\vec{\boldsymbol{b}}]_\times & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 3} \\ \vec{\boldsymbol{a}}'-\vec{\boldsymbol{b}}' & [\vec{\boldsymbol{a}}'+\vec{\boldsymbol{b}}']_\times & \vec{\boldsymbol{a}}-\vec{\boldsymbol{b}} & [\vec{\boldsymbol{a}}+\vec{\boldsymbol{b}}]_\times \end{pmatrix} \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{R}' \end{pmatrix} = 0 \tag{37}$$

where the matrix - we will call $\boldsymbol{S}$ - is a 6×8 matrix and the vector of unknowns $(\boldsymbol{R}^T, \boldsymbol{R}'^T)$ is 8-dimensional. Recall that we have two constraints on the unknowns so that the result is a unit dual rotor

$$< \boldsymbol{R}\boldsymbol{R}' >= 1 \qquad \text{and} \qquad < \boldsymbol{R}\boldsymbol{R}' >= 0. \tag{38}$$

We could think that six equations plus two constraints would suffice, however, the vectors $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$ are unit vectors and the vectors $\vec{\boldsymbol{a}}'$ and $\vec{\boldsymbol{b}}'$ are perpendicular to $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$ so that two equations are redundant. This is nothing new, since it is well known that at least two lines are necessary so that 3D motion can be estimated from their correspondences. Thus, we need at least two motions of the hand-eye system in order to get two lines from the corresponding screws. Chen [12] recognized also this fact and analyzed the uniqueness of the problem. He geometrically proved that even in the case of two parallel rotation axis we can compute all parameters up to the pitch. Suppose now that $n\geq 2$ motions are given. We construct the $6n\times 8$ matrix

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{S}_1^T & \boldsymbol{S}_2^T & \ldots & \boldsymbol{S}_n^T \end{pmatrix}^T \tag{39}$$

which in the noise-free case has rank 6. Since in the noise-free case the equations arise from natural constraints the null-space contains at least the solution $(\boldsymbol{R}, \boldsymbol{R}')$. It is trivial to see that an additional orthogonal solution is $(\mathbf{0}_{4\times 1}, \boldsymbol{R})$. Hence, the matrix is maximally of rank 6. If all axes $\vec{\boldsymbol{b}}$ are mutually parallel then the rank of the matrix is 5.
We compute the Singular Value Decomposition (SVD) $\boldsymbol{T}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{\Sigma}$ is a diagonal matrix with the singular values, the columns of $\boldsymbol{U}$ are the left singular vectors, and the columns of $\boldsymbol{V}$ are the right singular vectors. If the rank is 6 than the last two right singular vectors $\vec{\boldsymbol{v}}_{\boldsymbol{7}}$ and $\vec{\boldsymbol{v}}_{\boldsymbol{8}}$ - corresponding to the two vanishing singular values - span the nullspace of $\boldsymbol{T}$. We write them as composed of two 4×1 vectors $\vec{\boldsymbol{v}}_{\boldsymbol{7}}{}^T = (\vec{\boldsymbol{u}}_1^T, \vec{\boldsymbol{v}}_1^T)$ and $\vec{\boldsymbol{v}}_{\boldsymbol{8}}{}^T = (\vec{\boldsymbol{u}}_2^T, \vec{\boldsymbol{v}}_2^T)$. A vector $(\boldsymbol{R}^T, \boldsymbol{R}'^T)$ satisfying $\boldsymbol{T}(\boldsymbol{R}^T, \boldsymbol{R}'^T)^T = 0$ must be a linear combination of $\vec{\boldsymbol{v}}_{\boldsymbol{7}}$ and $\vec{\boldsymbol{v}}_{\boldsymbol{8}}$, hence

$$\begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{R}' \end{pmatrix} = \lambda_1 \begin{pmatrix} \vec{\boldsymbol{u}}_1 \\ \vec{\boldsymbol{v}}_1 \end{pmatrix} + \lambda_2 \begin{pmatrix} \vec{\boldsymbol{u}}_2 \\ \vec{\boldsymbol{v}}_2 \end{pmatrix}.$$

The two degrees of freedom are fixed by the constraints (38) which imply two quadratic equations in λ_1 and λ_2:

$$\begin{aligned} \lambda_1^2 \vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{u}}_1 + 2\lambda_1\lambda_2 \vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{u}}_2 + \lambda_2^2 \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{u}}_2 &= 1 && (40) \\ \lambda_1^2 \vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_1 + \lambda_1\lambda_2 (\vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_2 + \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_1) + \lambda_2^2 \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_2 &= 0 && (41) \end{aligned}$$

Since λ_1 and λ_2 never both vanish, assume w.l.o.g. that $\vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_1 \neq 0$ so that $\lambda_2 \neq 0$. Setting $s=\lambda_1/\lambda_2$ we first solve (41) obtaining two solutions for s. Inserting $\lambda_1 = s\lambda_2$ in (40) yields

$$\lambda^2 (s^2 \vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_1 + s(\vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_2 + \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_1) + \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_2) = 1 \tag{42}$$

which has two solutions of opposite sign. The sign variation is due to the sign invariance of the solution: Both $(\boldsymbol{R}^T, \boldsymbol{R}'^T)$ and $(-\boldsymbol{R}^T, -\boldsymbol{R}'^T)$ satisfy both the motion equations and the constraints. From the other two solutions it turns out that the second solution for s causes always the vanishing of the factor in the left hand side of (42). It corresponds to the solution $(\mathbf{0}_{4\times 1}, \boldsymbol{R})$ which does not satisfy the first constraint. The computation algorithm consists of the following steps:

1. Given n motor motions $(\boldsymbol{b}_i, \boldsymbol{b}_i')$ and corresponding camera motions $(\boldsymbol{a}_i, \boldsymbol{a}_i')$ check if the scalar parts are equal. Then extract the line directions and moments of the screw axes and construct the matrix $\boldsymbol{T}$ in (39).
2. Compute the SVD of $\boldsymbol{T}$ and check if only two singular values are almost equal to zero (due to noise we apply a threshold). Take the corresponding right singular vectors $\vec{\boldsymbol{v}}_7$ and $\vec{\boldsymbol{v}}_8$.
3. Compute the coefficients of (41) and solve it finding two solutions for s.
4. For these two values of s compute $(s^2 \vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_1 + s(\vec{\boldsymbol{u}}_1^T \vec{\boldsymbol{v}}_2 + \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_1) + \vec{\boldsymbol{u}}_2^T \vec{\boldsymbol{v}}_2)$ and choose the largest of them to compute λ_2 and then λ_1.
5.The result is $\lambda_1 \vec{\boldsymbol{v}}_7 + \lambda_2 \vec{\boldsymbol{v}}_8$.

5 Experiments

We present here results on simulations performed with our algorithm as well as with an existing two-step algorithm similar to [13]. The latter one estimates the quaternion rotation $\boldsymbol{q}$ from the equation $\boldsymbol{aq} = \boldsymbol{qb}$, then computes the rotation matrix $\boldsymbol{R}_X$ and solves for the translation $\vec{\boldsymbol{t}}_X$ from the corresponding vector equation (see section 6). The simulation procedure runs as follows: we establish n hand motions $(\boldsymbol{R}_b, \vec{\boldsymbol{t}}_b)$, we add Gaussian noise of relative standard deviation of 1% corresponding to the angle readings. We assume a hand-eye set up and compute the camera motions $(\boldsymbol{R}_a, \vec{\boldsymbol{t}}_a)$ to which we add also Gaussian noise of varying standard deviation. The noise is added as absolute value to the rotation axis direction and as relative value to the angle and the translation. For every noise setting the algorithm runs 1000 times and outputs the estimated rotation quaternion $\hat{\boldsymbol{q}}$ and the estimated translation $\hat{\vec{\boldsymbol{t}}}$ between gripper and camera. To qualify the results we take 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 $\|\vec{\boldsymbol{t}} - \hat{\vec{\boldsymbol{t}}}\|/\|\vec{\boldsymbol{t}}\|$. In the first experiment we tested a standard configuration of 20 hand motions with totally different rotation axes and large angles and a translation of 10-20mm. In Fig. 2 we compare our algorithm - marked as DUAL - with the two step algorithm - marked as SEPARATE. The superiority is shown especially in rotation where in our algorithm also the information from the hand and camera translations is used. In

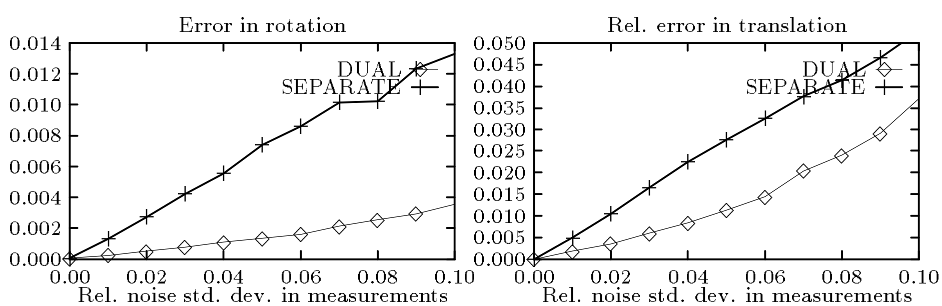

Figure 2: Behavior of the here proposed algorithm (DUAL) and of a two-step algorithm (SEPARATE) in variation of noise. On the left is shown the RMS rotation error and on the right the RMS relative translation error.

the second experiment we assumed no translation in the hand motions. The behavior of both algorithms is about the same (Fig. 3). This was expected because in absence of translations the dual parts of the measurements $(\boldsymbol{a}', \boldsymbol{b}')$ become zero. Then the left lower block of the matrix in (37) vanishes causing the separate computation of $\boldsymbol{R}$ and $\boldsymbol{R}'$. In a third experiment we kept the noise level at 5% and we varied the number of motions from 2 to 20. We observed that the behavior is about the same for two motions about our algorithms is superior in multiple motions.

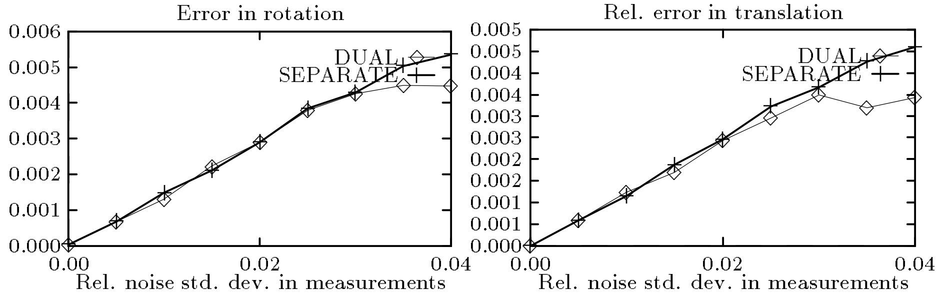

Figure 3: Both algorithms have the same performance in absence of translation.

6 Conclusion

This paper has presented the geometric algebra system for computations in computer vision and robotics. The rigid motions of the point, line and plane in 3-D and 4-D are elegantly expressed using rotors, motors and concepts of duality. It is shown that the system can operate simultaneously in different algebras transferring parameters for dealing with different needs like duality in geometric or in operational sense .
The invariance of the angle and the pitch helps to reduce the complexity of the hand-eye problem to a problem solvable using algebra of lines. The resultant parameterization enabled us to establish a linear homogeneous systems for resolving the dual rotor parameters. The computation of the nullspace with SVD and the consideration of the constraints for the dual rotors to be unit yields a simple algorithm avoiding non-linear steps. The algebraic structure of the linear system helps to understand much better the performance of the algorithm.

References

[1] Bayro-Corrochano E., Lasenby J., Sommer G. Geometric Algebra: A framework for computing point and line correspondences and projective structure using n-uncalibrated cameras *To appear in IEEE ICPR'96 Viena, Austria*, August, 1996.

[2] Bayro-Corrochano E., Buchholz S., Sommer G. Selforganizing Clifford Neural Network *To appear in IEEE ICNN'96 Washington, DC*, June, 1996.

[3] W.K. Clifford. Preliminary sketch of bi-quaternions. *Proc. London Math. Soc.*, 4:381–395, 1873.

[4] Hestenes, D. 1966. Space-Time Algebra. *Gordon and Breach.*

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

[6] Kantor, I.L. and Solodovnikov, A.S. . Hypercomplex Numbers: An Elementary Introduction to Algebras. Springer Verlag, New York Inc., 1989

[7] Lasenby, J. , Bayro-Corrochano E.J., Lasenby, A. and Sommer G. 1996. A new methodology for computing invariants in computer vision. Proceedings of ICPR'96, Viena.

[8] Shiu Y.C. and Ahmad S. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$. *IEEE Trans. Robotics and Automation*, 5:16–27, 1989.

[9] Study E. Geometrie der Dynamen. Leipsig 1903.

[10] Tsai R.Y. and Lenz R.K. A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. *IEEE Trans. Robotics and Automation*, 5:345–358, 1989.

[11] Horaud R. and Dornaika F.. Hand-eye calibration. *Intern. Journal of Robotics Research*, 14:195–210, 1995.

[12] Chen H. A screw motion approach to uniqueness analysis of head-eye geometry. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 145–151, Maui, Hawaii, June 3-6, 1991.

[13] Chou J.C.K. and Kamel M. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *Intern. Journal of Robotics Research*, 10(3):240-254, 1991.