A Unified Language for Computer Vision and Robotics

Eduardo Bayro-Corrochano*, Joan Lasenby‡

* Computer Science Institute, Christian Albrechts University, Kiel, Germany.
email: edb@informatik.uni-kiel.de
‡Department of Engineering, Trumpington Street, Cambridge CB2 1PZ.
email: jl@eng.cam.ac.uk

Abstract. Geometric algebra is an universal mathematical language which provides very comprehensive techniques for analyzing the complex geometric situations occurring in artificial Perception Action Cycle systems. In the geometric algebra framework such a system is both easier to analyze and to control in real time computations. This paper describes the application of rotors and motors for tasks involving the algebra of the 3D kinematics. Using purely geometric derivations and the constraints for point and line correspondences in n-views projective invariants are computed and the projective depth is discussed in terms of the generalized cross-ratio.

Categories: Clifford algebra; geometric algebra; robotics; hand-eye calibration; computer vision; projective invariants; projective depth.

1 Introduction

Biological and artificial intelligent systems show particular behaviour according to their situation and environment. They exist in a manifold structure where the time is distinguished as an axis rotated orthogonally from any spatial axis. Survival depends on the system's interrelated perceptive and active capabilities. This observable dependency can be delimited in a cycle of success. Within a Perception Action Cycle (PAC) a system interacts with its environment via visual and non-visual sensors for learning or accomplishing its task. Since each component of PAC systems requires different mathematical techniques, the construction of artificial PAC systems demands the fusion of the fields of signal theory, computer vision, robotics and neural computing in a framework with a powerful representation capability and a strong geometric basis. Currently, different mathematical systems are routinely employed for each part of the cycle. Each of these systems is limited in its applicability to one part of the cycle, making communication between different processes very difficult. Clearly, our ability to control the PAC would be considerably enhanced if a single mathematical language were employed throughout. In this paper the authors propose to use *geometric algebra* to analyze and construct algorithms for each phase of the PAC [29]. Geometric algebra is a coordinate-free approach to geometry based on the algebras of Grassmann [13] and Clifford [9]. Some preliminary applications of geometric algebra in the field of computer vision, robotics, neural computing and low level signal processing have already been given [1, 21, 3, 19, 2, 4]. For a more complete introduction see [14] and for other brief summaries see [16, 1, 3].

The next section will give the basics for the modelling of a work space for the projective space in terms of geometric algebra. The geometric algebra for the 3D kinematics is explained in section three. As two typical cases of robotics the motion of a multi-link and the hand-eye calibration are presented in Section four. The discussion of interesting issues of computer vision like the computation of projective invariants using n-views is presented in Section 7 and the recovery of projective depth is analyzed in Section 8. Finally, in the conclusion section the relevance of the geometric algebra framework for artificial PAC systems is discussed.

2 The 4D geometric algebra for the projective space

In a geometric algebra $\mathcal{G}_{p,q,r}$ we identify p and q as the dimensions of the maximal subspaces with positive and negative signatures, respectively (the signature of a vector $\boldsymbol{a}$ is positive, negative or zero according as $\boldsymbol{a}^2 > 0, < 0, = 0$). It is important for real applications to regard the signature of the modeled space to facilitate the computations. In the case of $\mathcal{G}_{3,0,0}$ we are adopting the standard Euclidean signature for the ordinary space, E^3, this forces to adopt the same signature for the 4-dimensional space $\mathcal{G}_{1,3,0}$ which we associate with the projective space P^3. This is spanned with the following basis

$$\underbrace{1}_{scalar}, \underbrace{\gamma_k}_{4\ vectors}, \underbrace{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3}_{6\ \ bivectors}, \underbrace{i\gamma_k}_{4\ \ pseudovectors}, \underbrace{i}_{pseudoscalar} \tag{1}$$

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

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

The fourth basis vector γ_4 can be seen also as selected direction or *projective split* [3] in 4D. The basis element γ_4 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 [3].

3 The 4D geometric algebra for 3D kinematics

One alternative to model the work space for the robotic field could simply the geometric algebra $\mathcal{G}_{3,0,0}$ of the 3D space. Since general displacements are non-linear transformations it would be more beneficial if we compute in a higher dimensional space. That is why the authors chose the special 4D algebra of the motors $\mathcal{G}^+_{3,0,1}$ as an efficient framework for 3D kinematics.

3.1 Motors

Clifford introduced the biquaternions with the name motors which is the abbreviation of "moment and vector"[10] for the algebra of 3D kinematics. Motors are

dual numbers with the necessary condition of $i^2 = 0$. They can be found in a special even geometric algebra $\mathcal{G}^+_{3,0,1}$ which here will be called the algebra of the motors. For detailed discussion of the role of dual, double and complex numbers in geometric algebra see [2]. Actually the algebra of the motors is a subalgebra of $\mathcal{G}_{3,0,1}$ for the 4D space with a similar basis presented in (1) with the difference that γ_k^2=1 for k=1,2,3 and γ_4^2=0 and that the pseudoscalar $i = \gamma_1\gamma_2\gamma_3\gamma_4$ squares to zero.

The important role that the motors play as a linear transformation is that they can absorb the translation component of a rigid motion. Let us explain this in more detail. Since in R^3 the simple translation is a nonlinear transformation the general displacement will be too. Unfortunately the displacement in R^3 can not be represented as a 3x3 matrix transformation. The way how we can go around is embedding R^3 in the R^4 space. In this 4D space the general displacements will take place in the hyperplane X_4=1. For example the motion of any point $\mathbf{p}$ can be now expressed compactly using a 4x4 homogeneous transformation matrix, i.e.

$$\begin{bmatrix} \mathbf{p}' \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{1} \end{bmatrix}. \tag{3}$$

Recall that a general displacement can be also expressed in terms of dual matrix, i.e.

$$\begin{bmatrix} R & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \equiv R + i[\mathbf{t}]_x R \tag{4}$$

where R is a rotation matrix and $[\mathbf{t}]_x$ the tensorial notation of the antisymmetric matrix representing the translation. We will show below that this transformation is equivalent to a motor, see equation (9) . Note that the homogeneous coordinates are similar to the ones used in the geometric algebra of the projective space $\mathcal{G}_{1,3,0}$, however if we want to compute using motors, which requires that the pseudoscalar squares to zero, we are compelled to switch to the motor algebra or $\mathcal{G}^+_{3,0,1}$.

The algebra of the motors has the basis

$$\underbrace{1}_{scalar} \quad , \quad \underbrace{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3}_{6 \quad bivectors} \quad , \quad \underbrace{i}_{pseudoscalar} \tag{5}$$

where $i^2 = 0$. A motor $\boldsymbol{M}_g$ in general is

$$\boldsymbol{M}_g = \boldsymbol{m}\boldsymbol{M} \tag{6}$$

where $\boldsymbol{m} = \boldsymbol{a} + i\boldsymbol{b}$ is a dual number for dilation and $\boldsymbol{M}$ is a unit motor. From now on all equations referring motors use the unit motor $\boldsymbol{M}$. A basic geometric

interpretation of a unit motor $\boldsymbol{M}$ can be given using the sum of two non-coplanar lines expressed in terms of dual bivector basis, i.e.

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{X}_1\boldsymbol{X}_2 + \boldsymbol{X}_3\boldsymbol{X}_4 = (\boldsymbol{X}_1\cdot\boldsymbol{X}_2 + \boldsymbol{X}_1\wedge\boldsymbol{X}_2) + (\boldsymbol{X}_3\cdot\boldsymbol{X}_4 + \boldsymbol{X}_3\wedge\boldsymbol{X}_4)\\ &= (a_0 + a_1\gamma_2\gamma_3 + a_2\gamma_3\gamma_1 + a_3\gamma_1\gamma_2) + i(b_0 + b_1\gamma_2\gamma_3 + b_2\gamma_3\gamma_1 + b_3\gamma_3\gamma_2)\\ &= \mathbf{R} + i\mathbf{R}', \end{aligned} \tag{7}$$

This tells that a motor can be seen also as a dual rotor or dual quaternion. Let us now analyze the motor equation (7). If the lines are non-coplanar the motor represents a general displacement or rigid motion and it is exact equivalent to a screw [10], else being coplanar they build a new line which can be seen as a *degenerated* motor. Thus, it is also convenient if the translation is expressed as a sort of a rotor which might be called translator

$$\boldsymbol{T} = e^{\frac{1}{2}\boldsymbol{t}i} = 1 + i\frac{\boldsymbol{t}}{2}, \tag{8}$$

where $\boldsymbol{t} = t_1\gamma_2\gamma_3 + t_2\gamma_3\gamma_1 + t_3\gamma_1\gamma_2$. The motor in terms of a translator reads

$$\boldsymbol{M} = \mathbf{R} + i\mathbf{R}' = \boldsymbol{R} + i\frac{\boldsymbol{t}}{2}\boldsymbol{R} = (1 + i\frac{\boldsymbol{t}}{2})\boldsymbol{R} = \boldsymbol{T}\boldsymbol{R}. \tag{9}$$

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 also expressed in terms of the rotors using

$$\boldsymbol{R}'\tilde{\boldsymbol{R}} = (\frac{\boldsymbol{t}}{2}\boldsymbol{R})\tilde{\boldsymbol{R}} \tag{10}$$

therefore

$$\boldsymbol{t} = 2\boldsymbol{R}'\tilde{\boldsymbol{R}} \tag{11}$$

where the multiplication is a geometric product.

The norm of a motor $\boldsymbol{M}$ is defined as follows

$$|\boldsymbol{M}| = \boldsymbol{M}\tilde{\boldsymbol{M}} = \boldsymbol{T}\boldsymbol{R}\tilde{\boldsymbol{R}}\tilde{\boldsymbol{T}} = (1 + i\frac{\boldsymbol{t}}{2})\boldsymbol{R}\tilde{\boldsymbol{R}}(1 - i\frac{\boldsymbol{t}}{2}) = 1 + i\frac{\boldsymbol{t}}{2} - i\frac{\boldsymbol{t}}{2} = 1, \tag{12}$$

where $\tilde{\boldsymbol{M}}$ is the conjugate motor and 1 is the identity. The combination of two rigid motions can be expressed using two motors. The resultant motor describes the overall displacement, namely

$$\boldsymbol{M}_c = \boldsymbol{M}_a\boldsymbol{M}_b = (\boldsymbol{R}_a + i\boldsymbol{R}'_a)(\boldsymbol{R}_b + i\boldsymbol{R}'_b) = \boldsymbol{R}_c + i\boldsymbol{R}'_c. \tag{13}$$

Note that pure rotations combine multiplicatively and the dual parts, containing the translation components, combine additively.

3.2 Representing points, lines and planes in 4D

The special algebra of motors $\mathcal{G}^+_{3,0,1}$ has a bivector basis which in 4D span the line space. Thus let us start with the line using this bivector basis. Assume two points $\boldsymbol{X}_1 = (X_{11}, X_{12}, X_{13}, 1)$ and $\boldsymbol{X}_2 = (X_{21}, X_{22}, X_{23}, 1)$ lying on the hyperplane $X_4 = 1$ and belonging to this line. The line can be defined simply as an outer product of these points, i.e.

$$\begin{aligned}\boldsymbol{l}_d = \boldsymbol{X}_1 \wedge \boldsymbol{X}_2 = (X_{12}X_{23} - X_{13}X_{22})\gamma_2\gamma_3 + (X_{13}X_{21} - X_{11}X_{23})\gamma_3\gamma_1 + .. + \\ (X_{11}X_{22} - X_{12}X_{21})\gamma_1\gamma_2 + (X_{21} - X_{11})\gamma_4\gamma_1 + .. + \\ (X_{22} - X_{12})\gamma_4\gamma_2 + (X_{23} - X_{13})\gamma_4\gamma_3. \end{aligned} \tag{14}$$

Since this equation consists only of bivectors, it can be expressed straightforward in terms of the bivector basis, namely

$$\begin{aligned}\boldsymbol{l}_d &= (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2) + (L^{41}\gamma_4\gamma_1 + L^{42}\gamma_4\gamma_2 + L^{43}\gamma_4\gamma_3) \\ &= (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2) + i(L^{41}\gamma_2\gamma_3 + L^{42}\gamma_3\gamma_1 + L^{43}\gamma_1\gamma_2). \end{aligned} \tag{15}$$

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

$$l_d = \boldsymbol{n} + i(\boldsymbol{n} \times \boldsymbol{q}) = \boldsymbol{n} + i\boldsymbol{m}, \tag{16}$$

where $\boldsymbol{n} \times \boldsymbol{q} = -i\boldsymbol{n} \wedge \boldsymbol{q}$. This line representation using dual numbers is easier to understand and to manipulate algebraically and it is fully equivalent to the one in terms of Plücker coordinates.

For the case of the point representation, embedding a 3D point expressed as a vector $\boldsymbol{x}$ on the hyperplane $X_4 = 1$, the point $\boldsymbol{q}$ equation in $\mathcal{G}^+_{3,0,1}$ reads

$$\boldsymbol{q} = 1 + x_1\gamma_4\gamma_1 + x_2\gamma_4\gamma_2 + x_3\gamma_4\gamma_3 = 1 + i(x_1\gamma_2\gamma_3 + x_2\gamma_3\gamma_1 + x_3\gamma_1\gamma_2) = 1 + i\boldsymbol{x}. \tag{17}$$

Now, resorting to the duality principle we use the dual of the scalar i.e the pseudoscalar times d and the dual of the bivector basis to write straightforwardly the plane equation, i.e

$$\boldsymbol{\phi} = n_1\gamma_2\gamma_3 + n_2\gamma_3\gamma_1 + n_3\gamma_1\gamma_2 + id = \boldsymbol{n} + id \tag{18}$$

where $\boldsymbol{n}$ stands for the normal of the plane and d for the distance of the plane to the origin.

4 Modelling the 3D Motion of Points, Lines and Planes

In this section we will present the modelling of the 3D motion of basic geometric entities using rotors and motors. We will see below in the case of the

hand-eye calibration that is preferable to use motors for computing the rotation and translation of an unknown rigid motion simultaneously. Because using the rotor approach we compute the translation decoupled of the rotation increasing therefore the inaccuracy. Let us now consider the modelling of the motion of points, lines and planes in both R^3 and R^4.

In $\mathcal{G}_{3,0,0}$ 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(20). In the algebra of the motors $\mathcal{G}^+_{3,0,1}$ we expressed the line as equation (16) and proceed as follows

$$\boldsymbol{l}_a = \boldsymbol{n}_a + i\boldsymbol{m}_a = \boldsymbol{M}\boldsymbol{l}_b\tilde{\boldsymbol{M}} = \boldsymbol{R}\boldsymbol{n}_b\tilde{\boldsymbol{R}} + i(\boldsymbol{R}\boldsymbol{n}_b\tilde{\boldsymbol{R}}' + \boldsymbol{R}'\boldsymbol{n}_b\tilde{\boldsymbol{R}} + \boldsymbol{R}\boldsymbol{m}_b\tilde{\boldsymbol{R}}) \tag{19}$$

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

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t}. \tag{20}$$

In case of the algebra of the motors $\mathcal{G}^+_{3,0,1}$ we use the point representation of equation (17)

$$\boldsymbol{M}(1 + i\boldsymbol{x})\overline{\tilde{\boldsymbol{M}}} = \boldsymbol{T}\boldsymbol{R}(1 + i\boldsymbol{x})\tilde{\boldsymbol{R}}\boldsymbol{T} = 1 + i(\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t}). \tag{21}$$

The expression $\overline{\tilde{\boldsymbol{M}}} = \tilde{\boldsymbol{R}}\boldsymbol{T}$ was found independently by the authors and it is similar to the one presented by W. Blaschke [5]. Yet in general all our motor equations explain directly that motor expressions consist of the successive action of rotors and translators. Now, when dealing with the motion of planes as it is shown below again we apply from the right $\overline{\tilde{\boldsymbol{M}}}$. This is because the plane is the dual of the point. Since the algebra of the motors has a bivector basis which span the line space and if we use this basis for representing points (geometrically of a lower dimension) and for planes (one dimension higher) we require $\overline{\tilde{\boldsymbol{M}}}$ instead of simply $\tilde{\boldsymbol{M}}$ as a sort of compensation for this asymmetry.

For the plane in $\mathcal{G}_{3,0,0}$ 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

$$\boldsymbol{H}' = (\boldsymbol{R}\boldsymbol{x}\tilde{\boldsymbol{R}} + \boldsymbol{t})\cdot(\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}) + (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}}). \tag{22}$$

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

$$\boldsymbol{H}' = \boldsymbol{R}\boldsymbol{H}\tilde{\boldsymbol{R}} + (\boldsymbol{R}\boldsymbol{H}\tilde{\boldsymbol{R}})\cdot\boldsymbol{t}. \tag{23}$$

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 equation (18) the motion equation of the plane is

$$\boldsymbol{M}(\boldsymbol{n} + id)\overline{\tilde{\boldsymbol{M}}} = \boldsymbol{T}\boldsymbol{R}(\boldsymbol{n} + id)\tilde{\boldsymbol{R}}\boldsymbol{T} = \boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}} + i(d + (\boldsymbol{R}\boldsymbol{n}\tilde{\boldsymbol{R}})\cdot\boldsymbol{t}). \tag{24}$$

4.1 Application 1: movement of a robot arm

Let us consider a complicated robot mechanism in terms of a system of linked n-bars. This calls for an optimization approach to find out its configuration during a smooth movement. Let us analyze the problem first in 3D space and then in the 4D space.

In 3D using the geometric algebra $\mathcal{G}_{3,0,0}$ we define reference-frames attached to each turnable join. Any connected two bars, the $j-th$ bar and the $j+1$-bar, are referred simple by the relative position of the $j-th$ join with the $j-1$-th join and the next bar moved by its own join. This can be simply expressed by

$$\boldsymbol{x}_{j+1} = \boldsymbol{R}_{j-1}\boldsymbol{x}_{j-1}\tilde{\boldsymbol{R}}_{j-1} + \boldsymbol{R}_j\boldsymbol{x}'_j\tilde{\boldsymbol{R}}_j = \boldsymbol{R}_{j-1}\boldsymbol{x}_{j-1}\tilde{\boldsymbol{R}}_{j-1} + \boldsymbol{R}_j\boldsymbol{R}_{j-1}\boldsymbol{x}_j\tilde{\boldsymbol{R}}_{j-1}\tilde{\boldsymbol{R}}_j \quad (25)$$

where $\boldsymbol{R}_j$ is the rotor applied to the reference frame attached to the j-join and $\boldsymbol{x}'$ is the translation from reference frame $j-1$ to reference frame j which corresponds simply to the length of the j-bar. The equation for the position of end effector considering the whole mechanism reads

$$\boldsymbol{x} = \boldsymbol{R}_1\boldsymbol{x}_1\tilde{\boldsymbol{R}}_1 + \boldsymbol{R}_2\boldsymbol{x}'_2\tilde{\boldsymbol{R}}_2 + \boldsymbol{R}_3\boldsymbol{x}'_3\tilde{\boldsymbol{R}}_3 + ... + \boldsymbol{R}_n\boldsymbol{x}'_n\tilde{\boldsymbol{R}}_n. \quad (26)$$

Now for the 4D space we will use the algebra of the motors $\mathcal{G}^+_{3,0,1}$. In the 4D the equation (25) reads

$$\boldsymbol{x}'_{j+1} = \boldsymbol{R}_j\boldsymbol{T}_j\boldsymbol{R}_{j-1}\boldsymbol{x}'_{j-1}\tilde{\boldsymbol{R}}_{j-1}\boldsymbol{T}_j\tilde{\boldsymbol{R}}_j, \quad (27)$$

where $\boldsymbol{x}'_{j-1} = (1+i\boldsymbol{x}_{j-1})$ and $\boldsymbol{x}'_{j+1} = (1+i\boldsymbol{x}_{j+1})$ referred to their own coordinate systems. Assuming that all the robot bars are of the same length $\boldsymbol{x}$, then $\boldsymbol{T}_1 = (1+i\boldsymbol{x}) = \boldsymbol{T}_2 = \boldsymbol{T}_3 = ... = \boldsymbol{T}_n$. Equation (26) for the whole mechanism in 4D is now

$$\boldsymbol{x}'_n = \boldsymbol{R}_{n-1}...\boldsymbol{T}_3\boldsymbol{R}_2\boldsymbol{T}_2\boldsymbol{R}_1\boldsymbol{x}'_1\tilde{\boldsymbol{R}}_1\boldsymbol{T}_2\tilde{\boldsymbol{R}}_2\boldsymbol{T}_3...\tilde{\boldsymbol{R}}_{n-1}. \quad (28)$$

Comparing the 4D and 3D expressions, the formers are linear and more simple. This can be exploited when computing any robot arm motion, i.e. we can for a particular motion simplify the equation (28) by canceling some redundant or conjugate translators and rotors. This happens for instance when some degree of freedom of the robot arm joins remains for this particular motion invariant. Recall that the rotors and translators are bivectors and you can commute and associate them without acting the sign.

4.2 Application 2: Motors for hand-eye calibration as a case of motion of lines

The well known hand-eye equation firstly formulated by Shiu and Ahmad [27] and Tsai and Lenz [28] reads

$$\mathbf{AX} = \mathbf{XB} \quad (29)$$

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. Here matrices are represented in bold. The geometry of the hand-eye system is depicted in Figure 1. From the expression (29) the following matrix equation and a vector equation can be derived $\mathbf{R_A R_X = R_X R_B}$ and $(\mathbf{R}_A - \mathbf{I})\mathbf{t}_X = \mathbf{R}_X\mathbf{t}_B - \mathbf{t}_A$. Most of the approaches estimate first the rotation matrix decoupled from the translation. The problem requires at least two motions with rotations having not parallel axes [28]. Horaud and Dornaika [18] showed the instability of the computation of the $\mathbf{A_i}$ matrices given the projective matrices $\mathbf{M}_i = \mathbf{CA}_i = (\mathbf{CR}_{A_i}\mathbf{Ct}_{A_i})$. Let us assume that the matrix of the intrinsic parameters $\mathbf{C}$ remains constant during motions and that one extrinsic calibration $\mathbf{A_2}$ is known. Introducing $\mathbf{N}_i = \mathbf{CR}_{A_i}$ and $\mathbf{n}_i = \mathbf{Ct}_{A_i}$ and replacing $\mathbf{X}$=$\mathbf{A_2 Y}$, we get now as the hand-eye unknown $\mathbf{Y}$. Thus the equation (29) 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.

$$\mathbf{A}_2^{-1}\mathbf{A}_1 = \begin{bmatrix} \mathbf{N}_2^{-1}\mathbf{N}_1 & \mathbf{N}_2^{-1}(\mathbf{n}_1 - \mathbf{n}_2) \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix}. \tag{30}$$

Taking into consideration the selected matrices and relations, this result allows

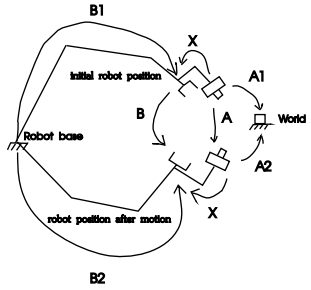

Fig. 1. Abstraction of the hand-eye system.

anyway to consider the formulation of the hand-eye problem again with the standard equation (29) which can be solved in terms of motors as

$$\boldsymbol{M}_A\boldsymbol{M}_X = \boldsymbol{M}_X\boldsymbol{M}_B \tag{31}$$

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 to the congruence theorem of Chen [7] in this kind of problem the rotation angle and pitch of $\boldsymbol{M}_A$ and $\boldsymbol{M}_B$ remain invariant through out all the hand movements. Thus the consideration of this information can be neglected. It suffices to regard the rotation axis of the involved motors, i.e. the previous equation is reduced

as the motion of the line axis of the hand towards the line axis of the camera. For that we can use the equation (19) for the computation of the real and dual components of $\boldsymbol{l}_A$, i.e.

$$\boldsymbol{l}_A = \boldsymbol{a} + i\boldsymbol{a}' = \boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}} + i(\boldsymbol{R}\boldsymbol{b}\tilde{\boldsymbol{R}}^{'} + \boldsymbol{R}\boldsymbol{b}'\tilde{\boldsymbol{R}} + \boldsymbol{R}'\boldsymbol{b}\tilde{\boldsymbol{R}}). \tag{32}$$

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

$$\begin{bmatrix} \mathbf{a}-\mathbf{b} & [\mathbf{a}+\mathbf{b}]_\times & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 3} \\ \mathbf{a}'-\mathbf{b}' & [\mathbf{a}'+\mathbf{b}']_\times & \mathbf{a}-\mathbf{b} & [\mathbf{a}+\mathbf{b}]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{R}' \end{bmatrix} = 0 \tag{33}$$

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. More technical details about the foundations and implementation of this approach can be found in [12, 2].

5 Projective Invariants

In the last two decades invariant theory captured also the attention of the computer vision community. This interest in invariants results from their usefulness in tasks like reconstruction, object recognition and hand-eye calibration. These are some examples of a much wider spectrum of invariants arising in a PAC system. In this section we will show the power of geometric algebra by computing a well known invariant which results when we consider six 3D points $P_i, i = 1, .., 6$ in general position, represented by vectors $\{\boldsymbol{x}_i, \mathbf{X}_i\}$ in E^3 and R^4 respectively.

5.1 Projective invariants using 2 uncalibrated cameras

3D projective invariants can be formed from these points, and an example of such an invariant is

$$Inv = \frac{[\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4][\mathbf{X}_4\mathbf{X}_5\mathbf{X}_2\mathbf{X}_6]}{[\mathbf{X}_1\mathbf{X}_2\mathbf{X}_4\mathbf{X}_5][\mathbf{X}_3\mathbf{X}_4\mathbf{X}_2\mathbf{X}_6]}. \tag{34}$$

It will be highly desirable to compute the brackets $[\mathbf{X}_i\mathbf{X}_j\mathbf{X}_k\mathbf{X}_l]$ simply in terms of **image coordinates** of points $P_i,\ P_j,\ P_k,\ P_l$, in order to compute this invariant straightforwardly. Carlsson [6] discussed the computation of such invariants from a pair of images in terms of the image coordinates and the fundamental matrix, $\boldsymbol{F}$, using the dual algebra. Subsequent work by Csurka and Faugeras [11] discussed corrections to Carlsson's expressions by including a series of scale factors. In contrast using geometric algebra we benefit of the duality principle and the projective split which allows us to simplify enormously the algebraic manipulation of the equations. Consider the scalar S_{1234} formed from the bracket of 4 points

$$S_{1234} = [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4] = (\mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3 \wedge \mathbf{X}_4) I_4^{-1} = (\mathbf{X}_1 \wedge \mathbf{X}_2) \wedge (\mathbf{X}_3 \wedge \mathbf{X}_4) I_4^{-1}. \tag{35}$$

The quantities $(\mathbf{X}_1 \wedge \mathbf{X}_2)$ and $(\mathbf{X}_3 \wedge \mathbf{X}_4)$ represent the line joining points P_1 and P_2, and P_3 and P_4. Let us represent the optical centres of both cameras by $\boldsymbol{a}_0$ and $\boldsymbol{b}_0$ and their image planes by $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ and $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$. Let the projection of points $\{P_i\}$ through the centres of projection onto the image planes be given by the vectors $\{\boldsymbol{a}'_i\}$ and $\{\boldsymbol{b}'_i\}$ which are ordinary vectors in E^3. The representations of these vectors in R^4 will be $\mathbf{A}_i, \mathbf{B}_i, \mathbf{A}'_i, \mathbf{B}'_i$..., etc.

In [3] it is shown that the bracket of these 4 points (in R^4) can be equated as

$$S_{1234} = [\mathbf{X}_1 \mathbf{X}_2 \mathbf{X}_3 \mathbf{X}_4] \equiv [\mathbf{A}_0 \mathbf{B}_0 \mathbf{A}'_{1234} \mathbf{B}'_{1234}]. \tag{36}$$

This is achieved by the process of splitting up the bracket into two parts, $\mathbf{X}_1 \wedge \mathbf{X}_2$ and $\mathbf{X}_3 \wedge \mathbf{X}_4$ and then expressing each of these lines (bivectors) as the meet of two planes (trivectors). During this algebraic computation, since we are working in R^4, we are effectively ignoring any scale factors due to the arbitrary choices of the γ_4 components. Thus, when we take ratios of brackets to form our invariants we must ensure that, if we want to express the brackets in the form of equation (36), the same decomposition of $\mathbf{X}_i \wedge \mathbf{X}_j$ must occur in the numerator and denominator so that these arbitrary factors cancel. In the case of Inv, we have

$$Inv = \frac{\{(\mathbf{X}_1 \wedge \mathbf{X}_2) \wedge (\mathbf{X}_3 \wedge \mathbf{X}_4)\} {I_4}^{-1} \{(\mathbf{X}_4 \wedge \mathbf{X}_5) \wedge (\mathbf{X}_2 \wedge \mathbf{X}_6)\} {I_4}^{-1}}{\{(\mathbf{X}_1 \wedge \mathbf{X}_2) \wedge (\mathbf{X}_4 \wedge \mathbf{X}_5)\} {I_4}^{-1} \{(\mathbf{X}_3 \wedge \mathbf{X}_4) \wedge (\mathbf{X}_2 \wedge \mathbf{X}_6)\} {I_4}^{-1}}. \tag{37}$$

Expanding the bracket in equation(36) by expressing the intersection points in terms of the $\mathbf{A}$'s and $\mathbf{B}$'s ($\mathbf{A}'_i = \alpha_{ij} \mathbf{A}_j$ and $\mathbf{B}'_i = \beta_{ij} \mathbf{B}_j$) and defining a matrix $\tilde{\boldsymbol{F}}$ such that

$$\tilde{F}_{ij} = [\mathbf{A}_0 \mathbf{B}_0 \mathbf{A}_i \mathbf{B}_j] \tag{38}$$

and vectors $\boldsymbol{\alpha}_{1234} = (\alpha_{1234,1}, \alpha_{1234,2}, \alpha_{1234,3})$ and $\boldsymbol{\beta}_{1234} = (\beta_{1234,1}, \beta_{1234,2}, \beta_{1234,3})$ it is easy to see that we can write $S_{1234} = {\boldsymbol{\alpha}^T}_{1234} \tilde{\boldsymbol{F}} \boldsymbol{\beta}_{1234}$ [6]. The ratio

$$Inv = \frac{({\boldsymbol{\alpha}^T}_{1234} \tilde{\boldsymbol{F}} \boldsymbol{\beta}_{1234})({\boldsymbol{\alpha}^T}_{4526} \tilde{\boldsymbol{F}} \boldsymbol{\beta}_{4526})}{({\boldsymbol{\alpha}^T}_{1245} \tilde{\boldsymbol{F}} \boldsymbol{\beta}_{1245})({\boldsymbol{\alpha}^T}_{3426} \tilde{\boldsymbol{F}} \boldsymbol{\beta}_{3426})} \tag{39}$$

is therefore seen to be an invariant. Note that equation (39) is invariant whatever values of the γ_4 components of the vectors $\mathbf{A}_i, \mathbf{B}_i, \mathbf{X}_i$ etc. are chosen. A confusion arises if we attempt to express the Inv of Eq. (39) in terms of what we actually observe, i.e. the 3D image coordinates and the fundamental matrix calculated from these image coordinates. To avoid that it is necessary to transfer the computations of Eq. (39) carried out in R^4 to 3D. Let us explain now this procedure.

A point P_i will be projected onto a point in image plane 1, say $\boldsymbol{a}'_i$, which can be written as

$$\boldsymbol{a}'_i = \boldsymbol{a}_1 + \lambda_i(\boldsymbol{a}_2 - \boldsymbol{a}_1) + \mu_i(\boldsymbol{a}_3 - \boldsymbol{a}_1) = \delta_{i1}\boldsymbol{a}_1 + \delta_{i2}\boldsymbol{a}_2 + \delta_{i3}\boldsymbol{a}_3 \tag{40}$$

so that $\sum_{j=1}^{3}\delta_{ij} = 1$. Similarly, we have $\boldsymbol{b}'_i = \epsilon_{i1}\boldsymbol{b}_1 + \epsilon_{i2}\boldsymbol{b}_2 + \epsilon_{i3}\boldsymbol{b}_3$ (so that $\sum_{j=1}^{3}\epsilon_{ij} = 1$). Using the projective split we can now write the α_{ij}'s and β_{ij}'s in terms of the δ_{ij}'s and ϵ_{ij}'s:

$$\alpha_{ij} = \frac{\mathbf{A}'_i\cdot\gamma_4}{\mathbf{A}_j\cdot\gamma_4}\delta_{ij} \qquad \beta_{ij} = \frac{\mathbf{B}'_i\cdot\gamma_4}{\mathbf{B}_j\cdot\gamma_4}\epsilon_{ij} \tag{41}$$

The 'fundamental' matrix $\tilde{\boldsymbol{F}}$ is such that ${\boldsymbol{\alpha}^T}_i\tilde{\boldsymbol{F}}\boldsymbol{\beta_i} = 0$, if $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$ are the vectors of coefficients of the points in planes 1 and 2 produced by the same world point P_i. Now, given more than eight pairs of corresponding observed points in the two planes, $(\boldsymbol{\delta_i}, \boldsymbol{\epsilon_i})$, $i = 1,..,8$, we can form an 'observed' fundamental matrix $\boldsymbol{F}$ such that

$${\boldsymbol{\delta}^T}_i\boldsymbol{F}\boldsymbol{\epsilon}_i = 0. \tag{42}$$

This $\boldsymbol{F}$ can be found by some method such as the Longuet-Higgins 8-point algorithm [23] or, more correctly, by some method which gives an $\boldsymbol{F}$ which has the true structure [24]. Therefore, if we define $\tilde{\boldsymbol{F}}$ by

$$\tilde{F}_{kl} = (\mathbf{A}_k\cdot\gamma_4)(\mathbf{B}_l\cdot\gamma_4)F_{kl} \tag{43}$$

then it follows from equations (41) that

$$\alpha_{ik}\tilde{F}_{kl}\beta_{il} = (\mathbf{A}'_i\cdot\gamma_4)(\mathbf{B}'_i\cdot\gamma_4)\delta_{ik}F_{kl}\epsilon_{il}\,. \tag{44}$$

If $\boldsymbol{F}$ is estimated then an $\tilde{\boldsymbol{F}}$ defined as in equation (43) will also act as a fundamental matrix in R^4.

Now let us look again at the invariant Inv. According to the above, we can write the invariant as

$$Inv = \frac{({\boldsymbol{\delta}^T}_{1234}\boldsymbol{F}\boldsymbol{\epsilon}_{1234})({\boldsymbol{\delta}^T}_{4526}\boldsymbol{F}\boldsymbol{\epsilon}_{4526})\phi_{1234}\phi_{4526}}{({\boldsymbol{\delta}^T}_{1245}\boldsymbol{F}\boldsymbol{\epsilon}_{1245})({\boldsymbol{\delta}^T}_{3426}\boldsymbol{F}\boldsymbol{\epsilon}_{3426})\phi_{1245}\phi_{3426}} \tag{45}$$

where $\phi_{pqrs} = (\mathbf{A}'_{pqrs}\cdot\gamma_4)(\mathbf{B}'_{pqrs}\cdot\gamma_4)$. We see therefore that the ratio of the $\boldsymbol{\delta}^T\boldsymbol{F}\boldsymbol{\epsilon}$ terms which resembles the expression for the invariant in R^4, but uses only the observed coordinates and the estimated fundamental matrix, will not be an invariant. Instead, we need to include the factors ϕ_{1234} etc., which do not cancel. It is relatively easy to show [20] that these factors can be formed as follows. Since $\boldsymbol{a}'_3$, $\boldsymbol{a}'_4$ and $\boldsymbol{a}'_{1234}$ are collinear we can write $\boldsymbol{a}'_{1234} = \mu_{1234}\boldsymbol{a}'_4 + (1-\mu_{1234})\boldsymbol{a}'_3$. Then, by expressing $\mathbf{A}'_{1234}$ as the intersection of the line joining $\mathbf{A}'_1$ and $\mathbf{A}'_2$ with the plane through $\mathbf{A}_0, \mathbf{A}'_3, \mathbf{A}'_4$ we can projective split and equate terms to give

$$\frac{(\mathbf{A}'_{1234}\cdot\gamma_4)(\mathbf{A}'_{4526}\cdot\gamma_4)}{(\mathbf{A}'_{3426}\cdot\gamma_4)(\mathbf{A}'_{1245}\cdot\gamma_4)} = \frac{\mu_{1245}(\mu_{3426}-1)}{\mu_{4526}(\mu_{1234}-1)}. \tag{46}$$

The values of μ are readily obtainable from the images. The factors $\mathbf{B}'_{pqrs}\cdot\gamma_4$ are found in a similar way so that if $\boldsymbol{b}'_{1234} = \lambda_{1234}\boldsymbol{b}'_4 + (1-\lambda_{1234})\boldsymbol{b}'_3$ etc., the overall

expression for the invariant becomes

$$Inv = \frac{(\boldsymbol{\delta}^T{}_{1234}\boldsymbol{F}\boldsymbol{\epsilon}_{1234})(\boldsymbol{\delta}^T{}_{4526}\boldsymbol{F}\boldsymbol{\epsilon}_{4526})}{(\boldsymbol{\delta}^T{}_{1245}\boldsymbol{F}\boldsymbol{\epsilon}_{1245})(\boldsymbol{\delta}^T{}_{3426}\boldsymbol{F}\boldsymbol{\epsilon}_{3426})} \frac{\mu_{1245}(\mu_{3426}-1)}{\mu_{4526}(\mu_{1234}-1)} \cdot \frac{\lambda_{1245}(\lambda_{3426}-1)}{\lambda_{4526}(\lambda_{1234}-1)}. \quad (47)$$

Thus, to summarize, given the coordinates of a set of 6 corresponding points in the two image planes (where these 6 points are projections from arbitrary world points but with the assumption that they are not coplanar) we can form 3D projective invariants provided we have some estimate of $\boldsymbol{F}$. A more detailed discussion on this issue you can find in [22].

6 Projective Structure Using n Uncalibrated Cameras

In this section we present the application of cross-ratio [20] for computing the *projective depth* discovered by Sashua [26]. This can be easily calculated using the cross-ratio of projected points lying on an epipolar line of any of the n cameras. This relation remains constant also for the ratio of the segments of an optical ray delimited by a tetrahedron or reference frame as is depicted in the Figure 2.

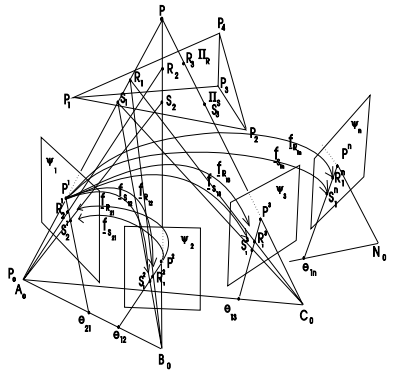

Fig. 2. Invariant projective depth using n uncalibrated cameras.

6.1 Homomorphic transformations

Consider a world point P and 4 other distinct points $P_i, i = 1,2,3,4$ defining a tetrahedron. Let $\Pi_R = \mathbf{P}_1 \wedge \mathbf{P}_3 \wedge \mathbf{P}_4$ and $\Pi_S = \mathbf{P}_1 \wedge \mathbf{P}_2 \wedge \mathbf{P}_3$ and assume P does not lie on either of these two planes – see figure 2. Let $\mathbf{R}_i$ and $\mathbf{S}_i$ be the intersections of the line joining the optical centre of the ith camera with point P with the planes Π_R and Π_S, e.g. $\mathbf{R}_1 = \Pi_R \vee (\mathbf{A}_0 \wedge \mathbf{P})$. Let $\mathbf{R}_i^n$ and $\mathbf{S}_i^n$ be the projections of the points $\mathbf{R}_i$ and $\mathbf{S}_i$ onto the nth image planes – e.g. $\mathbf{R}_1^2 = (\mathbf{B}_0 \wedge \mathbf{R}_1) \vee (\mathbf{B}_1 \wedge \mathbf{B}_2 \wedge \mathbf{B}_3)$ etc. Note that $\mathbf{R}_i^i$ and $\mathbf{S}_i^i$ are simply the projections of the world point P onto the ith image plane, e.g. $\mathbf{R}_1^1 = \mathbf{S}_1^1 = (\mathbf{A}_0 \wedge \mathbf{P}) \vee (\mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3)$. Let us call the ith image plane ψ_i.

In order to compute a cross-ratio which will be defined later, we must be able to calculate the image coordinates of $\mathbf{R}_i^n, \mathbf{S}_i^n$. We can do this by finding the *homomorphic* transformations or homographies relating projected points in one image plane to the projected points in another. Consider the homography between image planes ψ_i and ψ_j due to the plane Π_S. If the projections of $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ onto ψ_i and ψ_j are $\{\mathbf{P}_k^i\}$ and $\{\mathbf{P}_k^j\}$, for $k = 1, 2, 3$, then the linear function $\underline{f}_{ij}^S$ representing this transformation must satisfy

$$\underline{f}_{ij}^S(\mathbf{P}_k^i) = \mathbf{P}_k^j \quad \text{for } k = 1, 2, 3. \tag{48}$$

Here we are working in R^3 so that the non-linear projective transformations in $\mathcal{E}^2$ (plane to plane) become linear – the above linear-function representation is outlined in [21]. Similarly, the corresponding homography due to the plane Π_R is represented by the linear function $\underline{f}_{ij}^R$ given by

$$\underline{f}_{ij}^R(\mathbf{P}_k^i) = \mathbf{P}_k^j \quad \text{for } k = 1, 3, 4. \tag{49}$$

If four point correspondences from each plane are known then these linear functions can be recovered up to a scale factor by simple linear techniques. Since the homographies must map the epipole in one image plane onto the epipole in the other, we can choose the epipoles as the fourth point if these are known; $\underline{f}_{ij}^R(\mathbf{E}_{ji}) = \mathbf{E}_{ij}$ etc.

6.2 Computing the projective depth

The fundamental projective invariant in 1D is the cross-ratio. We can form a cross-ratio from the collinear points $\mathbf{P}, \mathbf{R}_1, \mathbf{S}_1, \mathbf{A}_0$, namely

$$\rho = \frac{(\mathbf{R}_1 \wedge \mathbf{A}_0) I_2^{-1}}{(\mathbf{P} \wedge \mathbf{A}_0) I_2^{-1}} \frac{(\mathbf{P} \wedge \mathbf{S}_1) I_2^{-1}}{(\mathbf{R}_1 \wedge \mathbf{S}_1) I_2^{-1}}. \tag{50}$$

The cross-ratio ρ will be invariant when projected onto any other image plane. Consider this cross-ratio in the image plane of the second camera;

$$\rho = \frac{(\mathbf{R}_1^2 \wedge \mathbf{E}_{12}) I_2^{-1}}{(\mathbf{P}^2 \wedge \mathbf{E}_{12}) I_2^{-1}} \frac{(\mathbf{P}^2 \wedge \mathbf{S}_1^2) I_2^{-1}}{(\mathbf{R}_1^2 \wedge \mathbf{S}_1^2) I_2^{-1}}. \tag{51}$$

If we know the linear functions $\underline{f}_{12}^S, \underline{f}_{12}^R$, then we can write this ratio as

$$\rho = \frac{(\underline{f}_{12}^R(\mathbf{P}^1) \wedge \mathbf{E}_{12}) I_2^{-1}}{(\mathbf{P}^2 \wedge \mathbf{E}_{12}) I_2^{-1}} \frac{(\mathbf{P}^2 \wedge \underline{f}_{12}^S(\mathbf{P}^1)) I_2^{-1}}{(\underline{f}_{12}^R(\mathbf{P}^1) \wedge \underline{f}_{12}^S(\mathbf{P}^1)) I_2^{-1}}, \tag{52}$$

and the general form for the i-camera and j-camera

$$\rho = \Big(\frac{(\underline{f}_{ij}^R(\mathbf{P}^i) \wedge \mathbf{E}_{ij}) I_2^{-1}}{(\mathbf{P}^j \wedge \mathbf{E}_{ij}) I_2^{-1}} \Big) \Big(\frac{(\mathbf{P}^j \wedge \underline{f}_{ij}^S(\mathbf{P}^i)) I_2^{-1}}{(\underline{f}_{ij}^R(\mathbf{P}^i) \wedge \underline{f}_{ij}^S(\mathbf{P}^i)) I_2^{-1}} \Big). \tag{53}$$

The term in the right bracket is termed *projective depth* in [26]. If we have a number of views available then, in this framework, a more robust estimate of k would be given by

$$k = \frac{1}{n} \sum_{(i \neq j)} \frac{(\mathbf{P}^j \wedge \underline{f}_{ij}^S(\mathbf{P}^i))\, I_2^{-1}}{(\underline{f}_{ij}^R(\mathbf{P}^i) \wedge \underline{f}_{ij}^R(\mathbf{P}^i))\, I_2^{-1}}, \tag{54}$$

where n is the number of estimates used.

Finally according with k the reconstruction of 3D coordinates of points is straightforward [26]. Considering the relations $\mathbf{P} \cong \mathbf{R}_1 + k\mathbf{S}_1$ and that of the fifth point (mapped onto the focal center $\mathbf{A}_0$) $\mathbf{P}_0 \cong \mathbf{R}_1 + k'\mathbf{S}_1$ (scaled so that k'=1), it can be seen that the depth k is actually an invariant up to uniform scale. Using this k we can reconstruct for each five points in general position (three lying on Π_1 and three lying on Π_2) the 3D coordinates of a point $\mathbf{P}_i$.

7 Conclusions

This paper presented the Clifford algebra in its geometric interpretation as a common language for the treatment of problems of robotics and computer vision. The authors focused in the geometry of 3D and 4D spaces which are necessary for the representation and manipulation of basic geometric entities required in those areas. In the first field we analyze the 3D and 4D modelling of motion in complicated mechanisms and in the hand-eye calibration problem. A 3D motion or general displacement is a nonlinear transformation, but linear if it is represented in 4D. This motivates us to use the 4D geometric algebra to solve in a linear manner problems involving 3D motions. An very interesting example is the hand-eye calibration. This requires a nonlinear solution strategy if it is treated in 3D. However when the 3D representation is extended to the 4D space using the motor algebra the problem of computing of the unknown motion becomes linear. In the second part of the paper it is shown how geometric algebra can be used for the algebra of incidence useful in the projective space. For intersections of planes, lines etc. and for the discussion of projective transformations we find that we do not need to invoke the standard concepts or machinery of classical projective geometry, all that is needed is the idea of the *projective split* relating the quantities in R^4 to quantities in our 3D world and the algebra of incidence.The case of computing invariants using n uncalibrated cameras is analyzed. Here the duality principle and projective split help to reduce the complexity of the computation. Finally using the cross-ratio of points lying in the epipolar lines of n cameras the projective reconstruction is addressed. The authors believe that the geometric algebra approach opens a new way to deal with problems in computer vision and differs with the standard approaches substantially due to its powerful algebraic and geometric properties.

Since PAC systems require different mathematical techniques for visual and motor signal processing, the construction of PAC systems demands of a framework where should be possible the fusion of the fields of signal theory, computer

vision, robotics and neural computing. It have been seen by the problems treated in this paper that geometric algebra indeed has a powerful representation capability and a strong geometric basis. That is why the authors believe that it is a competitive language to provide a unified approach for the design and implementation of real time PAC systems. At last it is easy to identify that the disparate mathematical techniques used nowadays in PAC are simply special cases of the wider class of mathematics provided by geometric algebra.

Acknowledgment

The authors are thankful to Gerald Sommer for his valuable suggestions and for the enriching disscussions. Eduardo Bayro-Corrochano is supported by the Deutsche Forschungsgemeinschaft project SO 320-2-1 "Geometrische Algebra ein Repräsentationsrahmen für den Wahrnehmungs-Handlungs-Zyklus" and Joan Lasenby by a Royal Society University Research Fellowship.

References

1. Bayro-Corrochano, E. and Lasenby, J. 1995. Object modelling and motion analysis using Clifford algebra. In Proceedings of Europe-China Workshop on *Geometric Modeling and Invariants for Computer Vision*, Ed. Roger Mohr and Wu Chengke, Xi'an, China, April, pp. 143-149.
2. Bayro-Corrochano, E., Daniilidis, K. and Sommer, G. 1997. Hand-Eye calibration in terms of motion of lines using Geometric Algebra. In *Proc. of the 10th Scandinavian Conference on Image Analysis SCIA'97*, Lappeenranta, Finland, June 9-11, Vol.I , pp. 397-404.
3. Bayro-Corrochano E., Lasenby J., Sommer G. Geometric Algebra: A framework for computing point and line correspondences and projective structure using n uncalibrated cameras *IEEE Proceedings of ICPR'96 Viena, Austria*, Vol. I, pages 334-338, August, 1996.
4. Bayro-Corrochano E., Buchholz S., Sommer G. 1996. Selforganizing Clifford neural network *IEEE ICNN'96 Washington, DC*, June, pp. 120-125.
5. Blaschke, W. 1960. Kinematik und Quaternionen. VEB Deutscher Verlag der Wissenschaften, Berlin 1960.
6. Carlsson, S. 1994. The Double Algebra: and effective tool for computing invariants in computer vision. *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science 825; Proceedings of 2nd-joint Europe-US Workshop, Azores, October 1993. Eds. Mundy, Zisserman and Forsyth. Springer-Verlag.
7. 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.
8. 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.
9. Clifford, W.K. 1878. Applications of Grassmann's extensive algebra. *Am. J. Math.* 1: 350–358.
10. Clifford., W.K. 1873. Preliminary sketch of bi-quaternions. *Proc. London Math. Soc.*, 4:381–395.

11. Csurka, G. and Faugeras, O. 1995. Computing three-dimensional projective invariants from a pair of images using the Grassmann-Cayley algebra. In Proceedings of Europe-China Workshop on *Geometric Modeling and Invariants for Computer Vision*, Ed. Roger Mohr and Wu Chengke, Xi'an, China, April, pp. 150-157.
12. Daniilidis K. and Bayro-Corrochano E. The dual quaternion approach to hand-eye calibration. *IEEE Proceedings of ICPR'96 Viena, Austria*, Vol. I, pages 318-322, August, 1996.
13. Grassmann, H. 1877. Der Ort der Hamilton'schen Quaternionen in der Ausdehnungslehre. *Math. Ann.*, 12: 375.
14. Hestenes, D. 1986. New Foundations for Classical Mechanics *D. Reidel*, Dordrecht.
15. Hestenes, D. 1991. The design of linear algebra and geometry. *Acta Applicandae Mathematicae*, 23: 65–93.
16. Hestenes, D. and Sobczyk, G. 1984. Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics. *D. Reidel*, Dordrecht.
17. Hestenes, D. and Ziegler, R. 1991. Projective geometry with Clifford algebra. *Acta Applicandae Mathematicae*, 23: 25–63.
18. Horaud R. and Dornaika F. Hand-eye calibration. *Intern. Journal of Robotics Research*, 14:195–210, 1995.
19. Lasenby, J., Fitzgerald, W.J., Lasenby, A.N. and Doran, C.J.L. 1997. New geometric methods for computer vision. To appear in *International Journal of Computer Vision*.
20. Lasenby, J., Bayro-Corrochano, E., Lasenby, A. and Sommer, G. 1996. A New Methodology for the Computation of Invariants in Computer Vision. Cambridge University Engineering Department Technical Report, CUED/F-INENG/TR.244.
21. Lasenby, J. , Bayro-Corrochano E.J., Lasenby, A. and Sommer G. 1996. A new methodology for computing invariants in computer vision. IEEE Proceedings of ICPR'96, Viena, Austria, Vol. I, pages 393-397, August, 1996.
22. Lasenby, J. , Bayro-Corrochano E.J. . 1997. Computing 3D projective invariants from points and lines. To appear in *Int. Conference on Analysis of Images and Patterns CAIP'97*, Kiel, Germany, September, 1997.
23. Longuet-Higgins, H.C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293: 133–138.
24. Luong, Q-T. and Faugeras, O.D. 1996. The fundamental matrix: theory, algorithms and stability analysis. *IJCV*, 17: 43–75.
25. McCarthy J.M. Dual orthogonal matrices in manipulator kinematics IJRR, Vol.5, Number 2, 1986.
26. Shashua, A. 1994. Projective structure from uncalibrated images: structure from motion and recognition PAMI, 16(8), 778:790.
27. 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.
28. Tsai R.Y. and Lenz R.K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Rob. and Autom.*, 5:345–358, 1989.
29. Sommer G., Bayro-Corrochano E. and Bülow T. 1997. Geometric algebra as a framework for the perception–action cycle. Workshop on Theoretical Foundation of Computer Vision, Dagstuhl, March 13-19, 1996, Springer Wien.