We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

2

Visually Guided Robotics Using Conformal Geometric Computing

Eduardo Bayro-Corrochano, Luis Eduardo Falcon-Morales and Julio Zamora-Esquivel,
CINVESTAV, Department of Electrical Engineering and Computer Science. Unidad Guadalajara, Jalisco, Mexico

1. Abstract

Classical Geometry, as conceived by Euclid, was a plataform from which Mathematics started to build its actual form. However, since the XIX century, it was a language that was not evolving as the same pase as the others branches of Physics and Mathematics. In this way, analytic, non-Euclidean and projective geometries, matrix theory, vector calculus, complex numbers, rigid and conformal transformations, ordinary and partial differential equations, to name some, are different mathematical tools which are used nowadays to model and solve almost any problem in robotic vision, but the presence of the classical geometric theory in such solutions is only implicit. However, over the last four decades a new mathematical framework has been developed as a new lenguage where not only the classical geometry is included, but where many of these mathematical systems will be embedded too. Instead of using different notation and theory for each of those systems, we will simplify the whole study introducing the CGA, a unique mathematical framework where all those systems are embedded, gaining in principle clarity and simplicity. Moreover, incidence algebra operations as union and intersection of subspaces, are also included in this system through the meet and join operations. In this regard, CGA appears promising for dealing with kinematics, dynamics and projective geometric problems in one and only one mathematical framework.

In this chapter we propose simulated and real tasks for perception-action systems, treated in a unify way and using only operations and geometrical entities of this algebra. We propose applications to follow geometric primitives or ruled sufaces with an arm's robot for shape understanding and object manipulation, as well as applications in visual grasping. But we believe that the use of CGA can be of great advantage in visually guided robotics using stereo vision, range data, laser, omnidirectional or odometry based systems.

Keywords: Computer vision; Clifford (geometric) algebra; projective and affine geometry; spheres projective geometry; incidence algebra; 3D rigid motion; ruled surfaces; directed distance; visually guided robotics.

Source: Mobile Robots: Perception & Navigation, Book edited by: Sascha Kolski, ISBN 3-86611-283-1, pp. 704, February 2007, Plv/ARS, Germany

2. Introduction

The Occam's razor, a mediaeval logical principle, said that 'when you have two competing theories which make exactly the same predictions, the one that is simpler is the better'. From this perspective the CGA is a single mathematical framework that unify and include different systems as matrix algebra, projective geometry, conformal transformations and differential forms. This chapter is an introduction to the communities of computer vision and robotics of this novel computational framework, called Conformal Geometric Algebra (CGA). This subject has been also treated in a wide scope in [4].

Our mathematical approach appears promising for the development of perception action cycle systems, see Figure 1. The subjects of this chapter are an improvement to previous works [3, 5, 6, 7, 13], because using the CGA we are now including the group of transformations in our computations and expanding our study to more complex surfaces, the ruled surfaces. Other authors have used Grassmann–Cayley algebra in computer vision [14] and robotics [19], but while they can express in this standard mathematical system the key ideas of projective geometry, such as the meet, join, duality and projective split, it lacks of an inner (contractive) product and of the group of transformations, which cannot be included in a very simple and natural way to the system.

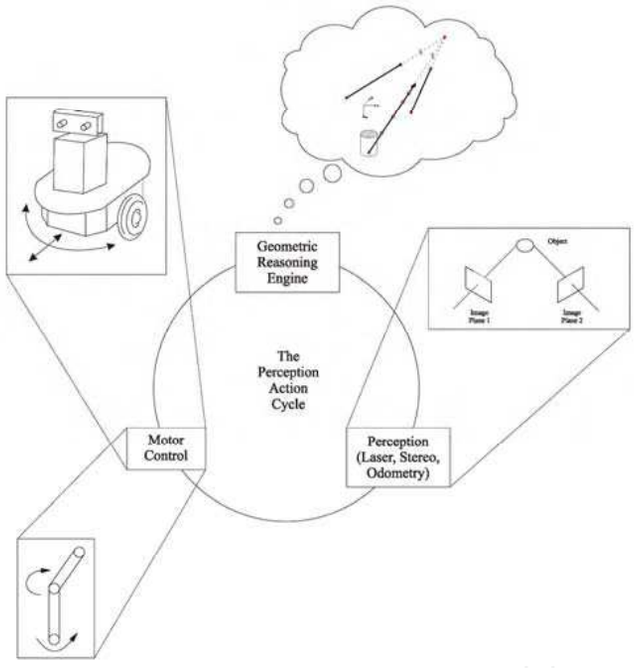

Fig. 1. Abstraction of the perception action cycle.

In fact, in the 1960's CGA take up again a proposal 'seeded' in the XIX century about build a global mathematical framework, which would include the main mathematical systems of that era: matrices and determinants; vector calculus; complex numbers; conformal transformations; Euclidean and projective spaces; differential forms; differential geometry; ordinary and partial differential equations.

In this chapter we put a lot of effort to explain clearly the CGA, illustrating the computations in great detail. Using the same ideas showed in this chapter, another practical tasks of visual guided robotics could be implemented for 3D motion estimation, *body − eye* calibration, 3D reconstruction, navigation, reaching and grasping 3D objects, etc. Thus, the idea is to introduce a suitable computational framework to the computer vision and robotic communities, which can be of great advantage for future applications in stereo vision, range data, laser, omnidirectional and odometry based systems.

CGA is the fusion of the Clifford Geometric Algebra (GA) and the non-Euclidean Hyperbolic Geometry. Historically, GA and CGA has not been taken into consideration seriously by the scientific community, but now and after the work of David Hestenes [10] and Pertti Lounesto [15] it has been taking a new scope of perspectives, not only theoretically, but for new and innovative applications to physics, computer vision, robotics and neural computing. One of the critics against CGA is the wrong idea that this system can manipulate only basic entities (points, lines, planes and spheres) and therefore it won't be useful to model general two and three dimensional objects, curves, surfaces or any other nonlinear entity required to solve a problem of a perception action system in robotics and computer vision. However, in this chapter we present the CGA, with its *algebra of incidence* [12] and rigid-motio*n* transformations, to obtain several practical techniques in the resolution of problems of perception action systems including ruled surfaces: 3D motion guidance of very non-linear curves; reaching and 3D object manipulation on very non-linear surfaces.

There are several interest points to study ruled surfaces: as robots and mechanisms are moving, any line attached to them will be tracing out a ruled surface or some other high nonlinear 3D-curve; the industry needs to guide the arm of robots with a laser welding to joint two ruled surfaces; reaching and manipulating 3D-objects is one of the main task in robotics, and it is usual that these objects have ruled surfaces or revolution surfaces; to guide a robot's arm over a critical 2D or 3D-curve or any other configuration constraint, and so forth.

The organization of this chapter paper is as follows: section two presents a brief introduction to conformal geometric algebra. Section three explains how the affine plane is embedded in the CGA. Section four shows how to generate the rigid transformations. In section five we present the way that several ruled surfaces or complex three dimensional curves can be generated in a very simple way using CGA. Section six shows how motors are usuful to obtain the Barret Hand™ forward kinematics. Section seven presents the real and simulated applications to follow geometric primitives and ruled surfaces for shape understanding and object manipulation, and section eight the applications to visual grasping identification. Conclusion are given in section nine.

2. Geometric Algebra

In general, a geometric algebra G_n is a 2^n-dimensional non-commutative algebra generated from a n-dimensional vector space V^n. Let us denote as $G_{p,q,r}$ this algebra where p, q, r denote the signature p, q, r of the algebra. If $p \neq 0$and $q = r = 0$, we have the standard Euclidean space and metric, if only $r \neq 0$ the metric is pseudoeuclidean and if $r \neq 0$ the metric is degenerate. See [17, 11] for a more detailed introduction to conformal geometric algebra.

We will use the letter e to denote the vector basis ei. In a geometric algebra Gp,q,r, the geometric product of two basis vectors is defined as

$$e_i e_j = \begin{cases} 1 & for\ i = j \in 1, ..., \mathrm{p} \\ -1 & for\ i = j \in \mathrm{p}+1, ..., \mathrm{p}+\mathrm{q} \\ 0 & for\ i = j \in \mathrm{p}+\mathrm{q}+1, ..., \mathrm{p}+\mathrm{q}+\mathrm{r} \\ e_i \wedge e_j & for\ i \neq j \end{cases} \tag{1}$$

2.1 Conformal Geometric Algebra

The geometric algebra of a 3D Euclidean space $G_{3,0,0}$ has a point basis and the motor algebra $G_{3,0,1}$ a line basis. In the latter geometric algebra the lines expressed in terms of Plücker coordinates can be used to represent points and planes as well. The reader can find a comparison of representations of points, lines and planes using $G_{3,0,0}$ and $G_{3,0,1}$ in [8].

Interesting enough in the case of the conformal geometric algebra we find that the unit element is the sphere which allows us to represent the other geometric primitives in its terms. To see how this is possible we begin giving an introduction in conformal geometric algebra following the same formulation presented in [11] and show how the Euclidean vector space R^n is represented in $R^{n+1,1}$. Let $\{e_1,.., e_n,e_+,e_-\}$ be a vector basis with the following properties

$$e_i^2 = 1, \quad i = 1..,n; \tag{2}$$

$$e_\pm^2 = \pm 1, \tag{3}$$

$$e_i \cdot e_+ = e_i \cdot e_- = e_+ \cdot e_- = 0, \quad i = 1,..,n. \tag{4}$$

Note that this basis is not written in bold. A *null basis* $\{e_0,e_\infty\}$ can be introduced by

$$e_0 = \frac{(e_- - e_+)}{2}, \tag{5}$$

$$e_\infty = e_- + e_+, \tag{6}$$

with the properties

$$e_0^2 = e_\infty^2 = 0, \quad e_\infty \cdot e_0 = -1. \tag{7}$$

A unit pseudoscalar $E \in \mathbb{R}^{1,1}$ which represents the so-called Minkowski plane is defined by

$$\boldsymbol{E} = e_\infty \wedge e_0 = e_+ \wedge e_- = e_+ e_-. \tag{8}$$

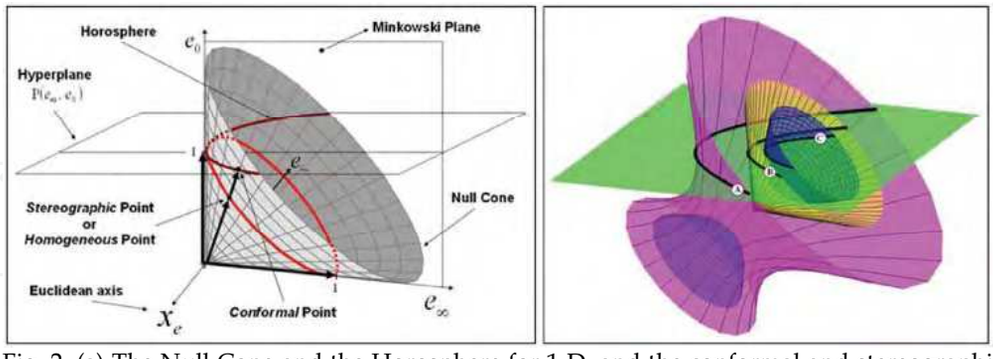

Fig. 2. (a) The Null Cone and the Horosphere for 1-D, and the conformal and stereographic representation of a 1-D vector. (b) Surface levels *A*, *B* and *C* denoting spheres of radius positive, zero and negative, respectively.

One of the results of the non-Euclidean geometry demonstrated by Nikolai Lobachevsky in the XIX century is that in spaces with hyperbolic structure we can find subsets which are

isomorphic to a Euclidean space. In order to do this, Lobachevsky introduced two constraints, to the now called *conformal point* $x_c \in \mathbb{R}^{n+1,1}$. See Figure 2(a). The first constraint is the *homogeneous* representation, *normalizing* the vector x_c such that

$$x_c \cdot e_\infty = -1, \tag{9}$$

and the second constraint is such that the vector must be a *null vector*,thatis,

$$x_c^2 = 0 \tag{10}$$

Thus, conformal points are required to lie in the intersection surface, denoted $\mathbb{N}_e^n$, between the *null cone* $\mathbb{N}^{n+1}$ and the *hyperplane* $\mathbb{P}(e_\infty, e_0)$:

$$\begin{aligned} \mathbb{N}_e^n &= \mathbb{N}^{n+1} \cap \mathbb{P}(e_\infty, e_0) \\ &= \{\mathbf{x}_c \in \mathbb{R}^{n+1,1} | \mathbf{x}_c^2 = 0, \mathbf{x}_c \cdot e_\infty = -1\}. \end{aligned} \tag{11}$$

The constraint (11) define an isomorphic mapping between the Euclidean and the Conformal space. Thus, for each conformal point $x_c \in \mathbb{R}^{n+1,1}$ there is a unique Euclidean point $\mathbf{x}_e \in \mathbb{R}^n$ and unique scalars *a*, *β* such that the mapping $\mathbf{x}_e \mapsto x_c = \mathbf{x}_e + \alpha e_0 + \beta e_\infty$. Then, the *standard form* of a conformal point x_c is

$$x_c = \mathbf{x}_e + \frac{1}{2}\mathbf{x}_e^2 e_\infty + e_0. \tag{12}$$

Note that a conformal point x_c and be splitted as

$$x_c = \mathbf{x}_e + \frac{1}{2}\mathbf{x}_e^2 e_\infty + e_0 = (x_c \wedge \boldsymbol{E})\boldsymbol{E} + (x_c \cdot \boldsymbol{E})\boldsymbol{E}. \tag{13}$$

We can gain further insight into the geometrical meaning of the null vectors by analyzing the isomorphism given by equation (13). For instance by setting $x_e = 0$ we find that e_0 represents the origin of R^n (hence the name). Similarly, dividing this equation by $x_c \cdot e_0 = -\frac{1}{2} x_e^{\,2}$ gives

$$\frac{x_c}{x_c \cdot e_0} = -\frac{2}{\mathbf{x}_e^2}(\mathbf{x}_e + \frac{1}{2}\mathbf{x}_e^2 e_\infty + e_0) = -\frac{2\mathbf{x}_e^2}{\mathbf{x}_e^2}(\frac{1}{\mathbf{x}_e} + \frac{1}{2}e_\infty + \frac{e_0}{\mathbf{x}_e^2}) = -2(\frac{1}{\mathbf{x}_e} + \frac{1}{2}e_\infty + \frac{e_0}{\mathbf{x}_e^2}) \underset{\mathbf{x}_e \to \infty}{\longrightarrow} e_\infty. \tag{14}$$

Thus we conclude that e_∞ represents the point at infinity.

The *dual* of a multivector $A \in Gn$ is defined by

$$\mathbf{A}^* = \mathbf{A}\, \mathbf{I}_n^{-1} \tag{16}$$

where $In \equiv e_{12\cdots n}$ is the unit *pseudoscalar* of *Gn* and the *inverse* of a multivector A*n*, if it exists, is defined by the equation $A^{-1}A=1$.

Duality give us the opportunity to define the *meet* $M \vee N$ between two multivectors *M* and *N* ,using one of the following equivalent expressions

$$meet(M, N) \equiv M \vee N = M^* \cdot N = M^* \wedge N^*. \tag{17}$$

Geometrically, this operation will give us the intersection between geometric primitives through the intersection of their generated subspaces. See [12].

2.2 Spheres and planes

The equation of a sphere of radius ρ centered at point p$e \in R^n$ can be written as

$$(\mathbf{x}_e - \mathbf{p}_e)^2 = \rho^2. \tag{18}$$

Since $\boldsymbol{x}_c \cdot \boldsymbol{y}_c = -\frac{1}{2}(\mathbf{x}_e - \mathbf{y}_e)^2$, we can rewrite the formula above in terms of homogeneous coordinates as

$$\boldsymbol{x}_c \cdot \boldsymbol{p}_c = -\frac{1}{2}\rho^2. \tag{19}$$

Since $x_c \cdot e_\infty = -1$ we can factor the expression above and then

$$\boldsymbol{x}_c \cdot (\boldsymbol{p}_c - \frac{1}{2}\rho^2 e_\infty) = 0, \tag{20}$$

which finally yields the simplified equation for the sphere as

$$\boldsymbol{x}_c \cdot \boldsymbol{s} = 0, \tag{21}$$

where

$$\boldsymbol{s} = \boldsymbol{p}_c - \frac{1}{2}\rho^2 e_\infty = \mathbf{p}_e + e_0 + \frac{\mathbf{p}_e^2 - \rho^2}{2} e_\infty \tag{22}$$

is the equation of the sphere. From this equation and (13) we can see that a conformal point is just a sphere with zero radius. The vector *s* has the properties

$$\boldsymbol{s}^2 = \rho^2 > 0, \tag{23}$$

$$e_\infty \cdot \boldsymbol{s} = -1. \tag{24}$$

From these properties, we conclude that the sphere *s* is a point lying on the hyperplane $xc \cdot e\infty = -1$, but *outside* the null cone $x^2 = 0$. In particular, all points on the hyperplane outside the horosphere determine spheres with positive radius, points lying on the horosphere define spheres of zero radius (i.e. points), and points lying inside the horosphere have imaginary radius. Finally, note that spheres of the same radius form a surface which is parallel to the horosphere.

Alternatively, spheres can be dualized and represented as $(n + 1)$-vectors $s^* = sI^{-1}$ and then using the *main convolution* $\tilde{I}$ of I defined as

$$\tilde{I} = (-1)^{\frac{1}{2}(n+2)(n+1)} I = -I^{-1}, \tag{25}$$

we can express the constraints of equations (23) and (24) as

$$\begin{aligned} \boldsymbol{s}^2 &= -\tilde{\boldsymbol{s}}^* s^* = \rho^2, \\ e_\infty \cdot \boldsymbol{s} &= e_\infty \cdot (\boldsymbol{s}^* I) = (e_\infty \wedge \boldsymbol{s}^*) I = -1. \end{aligned} \tag{26}$$

The equation for the sphere now becomes

$$\boldsymbol{x}_c \wedge \boldsymbol{s}^* = 0. \tag{27}$$

The advantage of the dual form is that the sphere can be directly computed from four points (in 3D) as

$$\boldsymbol{s}^* = \boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge \boldsymbol{x}_{c_3} \wedge \boldsymbol{x}_{c_4}. \tag{28}$$

If we replace one of these points for the point at infinity we get

$$\pi^* = \boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge \boldsymbol{x}_{c_3} \wedge e_\infty, \tag{29}$$

Developing the products, we get

$$\pi^* = \boldsymbol{x}_{c_3} \wedge \boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge e_\infty = \mathbf{x}_{e_3} \wedge \mathbf{x}_{e_1} \wedge \mathbf{x}_{e_2} \wedge e_\infty + ((\mathbf{x}_{e_3} - \mathbf{x}_{e_1}) \wedge (\mathbf{x}_{e_2} - \mathbf{x}_{e_1}))\boldsymbol{E}, \tag{30}$$

which is the equation of the plane passing through the points x_{e1}, x_{e2} and x_{e3}. We can easily see that $x_{e1} \wedge x_{e2} \wedge x_{e3}$ is a pseudoscalar representing the volume of the parallelepiped with sides xe_1, xe_2 and xe_3. Also, since $(xe_1 - xe_2)$ and $(xe_3 - xe_2)$ are two vectors on the plane, the expression $((x_{e1} - x_{e2}) \wedge (x_{e3} - x_{e2}))^*$ is the normal to the plane. Therefore planes are spheres passing through the point at infinity.

2.3 Geometric identities, duals and incidence algebra operations

A circle z can be regarded as the intersection of two spheres s_1 and s_2. This means that for each point on the circle $x_c \in z$ they lie on both spheres, that is, $x_c \in s_1$and $x_c \in s_2$. Assuming that s_1 and s_2 are linearly independent, we can write for $x_c \in z$

$$(\boldsymbol{x}_c \cdot \boldsymbol{s}_1)\boldsymbol{s}_2 - (\boldsymbol{x}_c \cdot \boldsymbol{s}_2)\boldsymbol{s}_1 = x_c \cdot (\boldsymbol{s}_1 \wedge \boldsymbol{s}_2) = \boldsymbol{x}_c \cdot \boldsymbol{z} = 0, \tag{31}$$

this result tells us that since x_c lies on both spheres, z=($s1 \wedge s1$) should be the intersection of the spheres or a circle. It is easy to see that the intersection with a third sphere leads to a point pair. We have derived algebraically that the wedge of two linearly independent spheres yields to their intersecting circle (see Figure 3), this topological relation between two spheres can be also conveniently described using the dual of the meet operation, namely

$$\boldsymbol{z} = (\boldsymbol{z}^*)^* = (\boldsymbol{s}_1^* \vee \boldsymbol{s}_2^*)^* = \boldsymbol{s}_1 \wedge \boldsymbol{s}_2, \tag{32}$$

this new equation says that the dual of a circle can be computed via the meet of two spheres in their dual form. This equation confirms geometrically our previous algebraic computation of equation (31).

The dual form of the circle (in 3D) can be expressed by three points lying on it as

$$\boldsymbol{z}^* = \boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge \boldsymbol{x}_{c_3}, \tag{33}$$

seeFigure3.a.

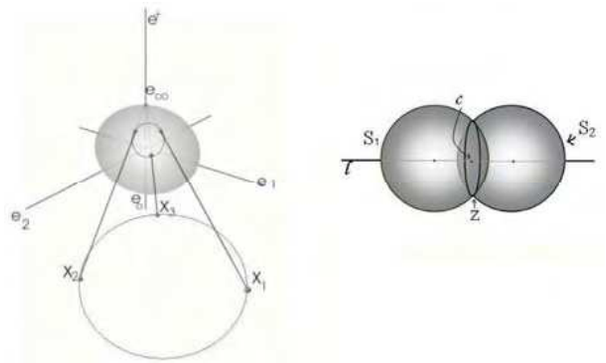

Fig. 3. a) Circle computed using three points, note its stereographic projection. b) Circle computed using the meet of two spheres.

Similar to the case of planes show in equation (29), lines can be defined by circles passing through the point at infinity as

$$\boldsymbol{l}^* = \boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge e_\infty. \tag{34}$$

This can be demonstrated by developing the wedge products as in the case of the planes to yield

$$\boldsymbol{x}_{c_1} \wedge \boldsymbol{x}_{c_2} \wedge e_\infty = \mathbf{x}_{e_1} \wedge \mathbf{x}_{e_2} \wedge e_\infty + (\mathbf{x}_{e_2} - \mathbf{x}_{e_1}) \wedge \boldsymbol{E}, \tag{35}$$

from where it is evident that the expression $x_{e1} \wedge x_{e2}$ is a bivector representing the plane where the line is contained and (x_{e2}− x_{e1}) is the direction of the line.

The dual of a point p is a sphere s. The intersection of four spheres yields a point, see Figure 4.b . The dual relationships between a point and its dual, the sphere, are:

$$\boldsymbol{s}^* = \boldsymbol{p}_1 \wedge \boldsymbol{p}_2 \wedge \boldsymbol{p}_3 \wedge \boldsymbol{p}_4 \leftrightarrow \boldsymbol{p}^* = \boldsymbol{s}_1 \wedge \boldsymbol{s}_2 \wedge \boldsymbol{s}_3 \wedge \boldsymbol{s}_4, \tag{36}$$

where the points are denoted as p_i and the spheres s_i for i =1, 2, 3, 4. A summary of the basic geometric entities and their duals is presented in Table 1.

There is another very useful relationship between a $(r-2)$-dimensional sphere A_r and the sphere s (computed as the dual of a point s). If from the sphere A_r we can compute the hyperplane $\boldsymbol{A}_{r+1} \equiv \boldsymbol{e}_\infty \wedge \boldsymbol{A}_r \neq 0$, we can express the meet between the dual of the point s(a sphere) and the hyperplane A_{r+1} getting the sphere A_r of one dimension lower

$$(-1)^\epsilon \boldsymbol{s}^* \cap \boldsymbol{A}_{r+1} = (\boldsymbol{s}^* I) \cdot \boldsymbol{A}_{r+1} = \boldsymbol{s}\boldsymbol{A}_{r+1} = \boldsymbol{A}_r. \tag{37}$$

This result is telling us an interesting relationship: that the sphere A_r and the hyperplane A_{r+1} are related via the point s (dual of the sphere s^*), thus we then rewrite the equation (37) as follows

$$\boldsymbol{s} = \boldsymbol{A}_r \boldsymbol{A}_{r+1}^{-1}. \tag{38}$$

Using the equation (38) and given the plane π (A_{r+1}) and the circle z(A_r) we can compute the sphere

$$s = z\pi^{-1}. \tag{39}$$

Similarly we can compute another important geometric relationship called the *pair of points* using the equation (38) directly

$$s = PPL^{-1}. \tag{40}$$

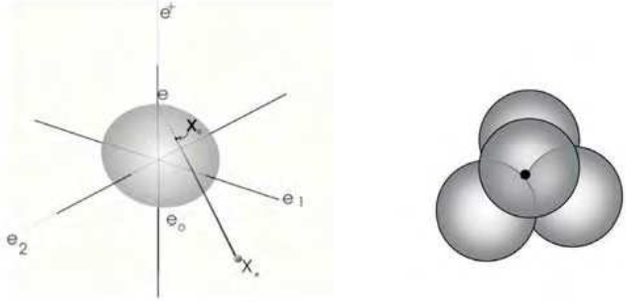

Fig. 4. a) Conformal point generated by projecting a point of the affine plane to the unit sphere. b) Point generated by the meet of four spheres.

Entity	Representation	Grade	Dual Representation	Grade
Sphere	$\boldsymbol{s} = \mathbf{p} + \frac{1}{2}(\mathbf{p}^2 - \rho^2)e_\infty + e_0$	1	$\boldsymbol{s}^* = \boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}\wedge\boldsymbol{d}$	4
Point	$\boldsymbol{x} = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$	1	$\boldsymbol{x}^* = \boldsymbol{s}_1\wedge\boldsymbol{s}_2\wedge\boldsymbol{s}_3\wedge\boldsymbol{s}_4$	4
Plane	$\pi = \boldsymbol{n}I_E - de_\infty$ $\boldsymbol{n} = (\boldsymbol{a}-\boldsymbol{b})\wedge(\boldsymbol{a}-\boldsymbol{c})$ $d = (\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c})I_E$	1	$\pi^* = e_\infty\wedge\boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}$	4
Line	$\boldsymbol{L} = \pi_1\wedge\pi_2$ $\boldsymbol{L} = \boldsymbol{n}I_E - e_\infty\boldsymbol{m}I_E$ $\boldsymbol{n} = (\boldsymbol{a}-\boldsymbol{b})$ $\boldsymbol{m} = (\boldsymbol{a}\wedge\boldsymbol{b})$	2	$\boldsymbol{L}^* = e_\infty\wedge\boldsymbol{a}\wedge\boldsymbol{b}$	3
Circle	$\boldsymbol{z} = \boldsymbol{s}_1\wedge\boldsymbol{s}_2$	2	$\boldsymbol{z}^* = \boldsymbol{a}\wedge\boldsymbol{b}\wedge\boldsymbol{c}$	3
Point Pair	$\boldsymbol{PP} = \boldsymbol{s}_1\wedge\boldsymbol{s}_2\wedge\boldsymbol{s}_3$ $\boldsymbol{PP} = \boldsymbol{s}\wedge\boldsymbol{L}$	3 2	$\boldsymbol{PP}^* = \boldsymbol{a}\wedge\boldsymbol{b}$	2

Table 1. Entities in conformal geometric algebra.

Now using this result given the line $L$and the sphere s we can compute the pair of points PP (see Figure 5.b)

$$PP = sL = s \wedge L. \tag{41}$$

3. The 3D Affine Plane

In the previous section we described the general properties of the conformal framework. However, sometimes we would like to use only the projective plane of the conformal framework but not the null cone of this space. This will be the case when we use only rigid transformations and then we will limit ourselves to the *Affine Plane* which is a $n + 1$ dimensional subspace of the Hyperplane of reference $P(e_\infty, e_0)$.
We have chosen to work in the algebra $G_{4,1}$. Since we deal with homogeneous points the particular choice of null vectors does not affect the properties of the conformal geometry. Points in the affine plane $x \in \mathbb{R}^{4,1}$ are formed as follows

$$\boldsymbol{x}^a = \mathbf{x}_e + e_0, \tag{42}$$

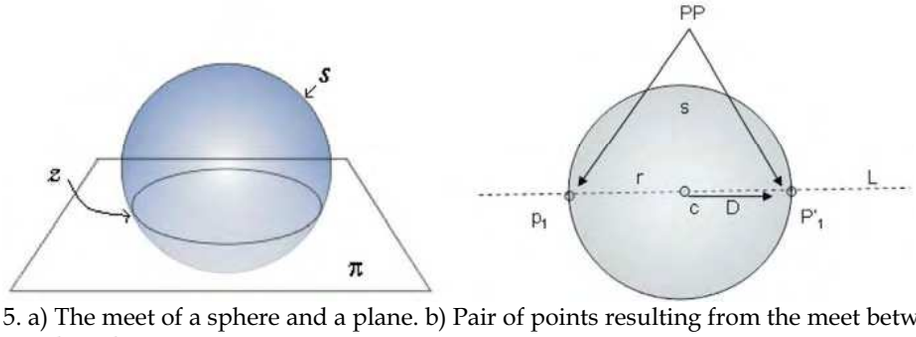

Fig. 5. a) The meet of a sphere and a plane. b) Pair of points resulting from the meet between a line and a sphere.

where $x_e \in \mathbb{R}^3$. From this equation we note that e_0 represents the origin (by setting $x_e = 0$), similarly, $e\infty$ represents the point at infinity. Then the normalization property is now expressed as

$$e_\infty \cdot \boldsymbol{x}^a = -1. \tag{43}$$

In this framework, the conformal mapping equation is expressed as

$$\boldsymbol{x}_c = \mathbf{x}_e + \frac{1}{2}\mathbf{x}_e^2 e_\infty + e_0 = \boldsymbol{x}^a + \frac{1}{2}\mathbf{x}_e^2 e_\infty. \tag{44}$$

For the case when we will be working on the affine plane exclusively, we will be mainly concerned with a simplified version of the *rejection*. Noting that $E = e_\infty \wedge e_0 = e_\infty \wedge e$, we write a equation for rejection as follows

$$\begin{aligned} P_E^{\perp}(\boldsymbol{x}_c) &= (\boldsymbol{x}_c \wedge \boldsymbol{E})\boldsymbol{E} = (\boldsymbol{x}_c \wedge \boldsymbol{E}) \cdot \boldsymbol{E} = (e_\infty \wedge e_0) \cdot e_0 + (\boldsymbol{x}_c \wedge e_\infty) \cdot e_0, \\ \mathbf{x}_e &= -e_0 + (\boldsymbol{x}_c \wedge e_\infty) \cdot e_0. \end{aligned} \tag{45}$$

Now, since the points in the affine plane have the form $x^a = xe + e0$, we conclude that

$$\boldsymbol{x}^a = (\boldsymbol{x}_c \wedge e_\infty) \cdot e_0, \tag{46}$$

is the mapping from the horosphere to the affine plane.

3.1 Lines and Planes

The lines and planes in the affine plane are expressed in a similar fashion to their conformal counterparts as the *join* of 2 and 3 points, respectively

$$\boldsymbol{L}^a = \boldsymbol{x}_1^a \wedge \boldsymbol{x}_2^a, \tag{47}$$

$$\Pi^a = \boldsymbol{x}_1^a \wedge \boldsymbol{x}_2^a \wedge \boldsymbol{x}_3^a. \tag{48}$$

Note that unlike their conformal counterparts, the line is a *bivector* and the plane is a *trivector*. As seen earlier, these equations produce a moment-direction representation thus

$$\boldsymbol{L}^a = e_\infty \mathbf{d} + \boldsymbol{B}, \tag{49}$$

where d is a vector representing the direction of the line and B is a bivector representing the moment of the line. Similarlywehavethat

$$\Pi^a = e_\infty \mathrm{n} + \delta e_{123} \tag{50}$$

where n is the normal vector to the plane and δ is a scalar representing the distance from the plane to the origin. Note that in any case, the direction and normal can be retrieved with $\mathrm{d} = e_\infty \cdot L^a$ and $\mathrm{n} = e_\infty \cdot \Pi^a$, respectively.

In this framework, the intersection or *meet* has a simple expression too. Let $\boldsymbol{A}^a = a_1^a \wedge \ldots \wedge a_r^a$ and $B^a = b_1^a \wedge \ldots \wedge b_s^a$, then the meet is defined as

$$\boldsymbol{A}^a \cap \boldsymbol{B}^a = \boldsymbol{A}^a \cdot (\boldsymbol{B}^a \cdot \bar{I}_{\boldsymbol{A}^a \cup \boldsymbol{B}^a}), \tag{51}$$

where $\Gamma_{A^a \cup B^a}$ is either $e_{12}e_\infty$, $e_{23}e_\infty$, $e_{31}e_\infty$,or $e_{123}e_\infty$, according to which basis vectors span the largest common space of A^a and B^a.

3.2 Directed distance

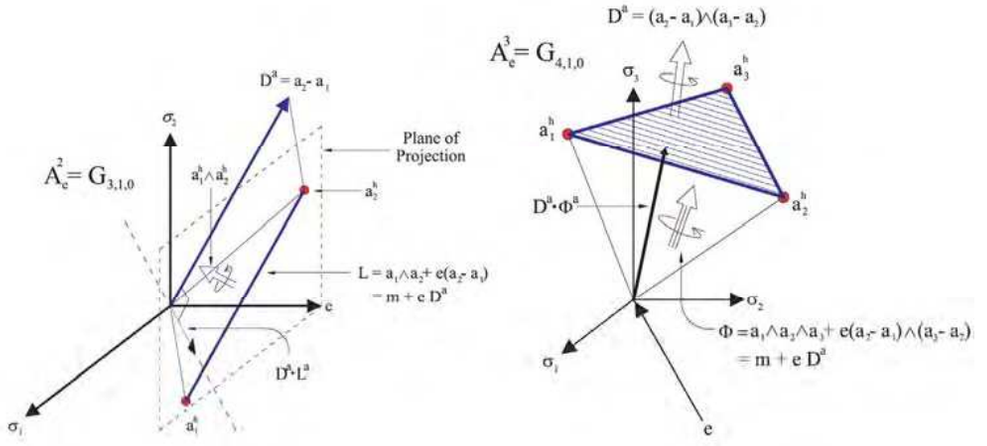

Fig. 6. a) Line in 2D affine space. b) Plane in the 3D affine space (note that the 3D space is "lifted" by a null vector e.

It is well known from vector analysis the so-called *Hessian normal form*, a convenient representation to specify lines and planes using their distance from the origin (the Hesse distance or Directed distance). In this section we are going to show how CGA can help us to obtain the Hesse distance for more general simplexes and not only for lines and planes. Figure 6(a) and (b) depict a line and a plane, respectively, that will help us to develop our equations. Let A^k be a k-line (or plane), then it consist of a *momentum* M^k of degree k and of a *direction* $D^{k-1}$of degree $k - 1$. For instance, given three Euclidean points a_1,a_2,a_3 their 2-simplex define a dual 3-plane in CGA that canbe expressedas

$$\boldsymbol{A}^k \equiv \Phi = M^3 + D^2 e_0 = a_1 \wedge a_2 \wedge a_3 + (a_2 - a_1) \wedge (a_3 - a_1) e_0. \tag{52}$$

Then, the directed distance of this plane, denoted as p^k , can be obtained taking the inner product between the unit direction $D^{k-1}$and the moment M^k . Indeed, from (52) and using expressions (1) to (7), we get the direction
from $\Phi\ e_\infty = D^{k-1}$and then its unitary expression D^{k-1} dividing D^{k-1} by its magnitude. Schematically,

$$\boldsymbol{A}^k \longrightarrow \boldsymbol{A}^k \cdot e_\infty = \boldsymbol{D}^{k-1} \longrightarrow \boldsymbol{D}_u^{k-1} = \frac{\boldsymbol{D}^{k-1}}{\left|\boldsymbol{D}^{k-1}\right|}. \tag{53}$$

Finally the directed distance p^k of A^k is

$$\boldsymbol{p}^k = \boldsymbol{D}_u^{k-1} \cdot \boldsymbol{A}^k, \tag{54}$$

where the dot operation basically takes place between the direction $D^{k-1}$and the momentum of A^k . Obviously, the directed distance vector p touches orthogonally the k-plane A^k, and as we mentioned at the beginning of k this subsection, the magnitude p equals the Hesse distance. For sake of simplicity, in Figures (6.a) and (6.b) only $D^{k-1} \cdot L^k$ and $D^{k-1} \cdot \Phi^k$ are respectively shown.
Now, having this point from the first object, we can use it to compute the directed distance from the k-plane A^k parallel to the object B^k as follows

$$d[\boldsymbol{A}^k, \boldsymbol{B}^k] = d[\boldsymbol{D}^{k-1} \cdot \boldsymbol{A}^k, \boldsymbol{B}^k] = d[(e_\infty \cdot \boldsymbol{A}^k) \cdot \boldsymbol{A}^k, \boldsymbol{B}^k]. \tag{55}$$

4. Rigid Transformations

We can express rigid transformations in conformal geometry carrying out reflections between planes.

4.1 Reflection

The reflection of conformal geometric entities help us to do any other transformation. The reflection of a point x respect to the plane π is equal x minus twice the direct distance between the point and plane see the image (7), that is $x = x - 2(\pi \cdot x)\pi^{-1}$ to simplify this expression recalling the property of Clifford product of vectors $2(b \cdot a) = ab + ba$.

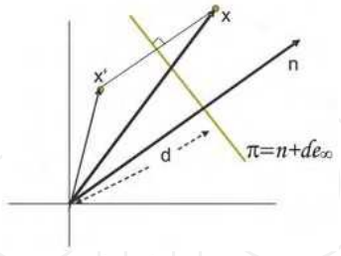

Fig. 7. Reflection of a point x respect to the plane π.

The reflection could be written

$$x' = x - (\pi x - x\pi)\pi^{-1}, \tag{56}$$

$$x' = x - \pi x\pi^{-1} - x\pi\pi^{-1} \tag{57}$$

$$x' = -\pi x\pi^{-1}. \tag{58}$$

For any geometric entity Q, the reflection respect to the plane π is given by

$$Q' = \pi Q\pi^{-1} \tag{59}$$

4.2 Translation

The translation of conformal entities can be by carrying out two reflections in parallel planes π_1 and π_2 see the image (8), that is

$$Q' = T_a\underbrace{(\pi_2\pi_1)}Q\widetilde{T_a}\underbrace{(\pi_1^{-1}\pi_2^{-1})} \tag{60}$$

$$T_a = (n+de_\infty)n = 1+\frac{1}{2}ae_\infty = e^{-\frac{a}{2}e_\infty} \tag{61}$$

With a =2dn.

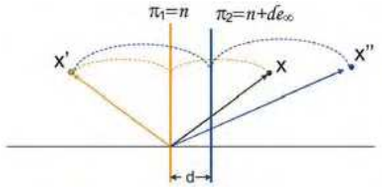

Fig. 8. Reflection about parallel planes.

4.3 Rotation

The rotation is the product of two reflections between nonparallel planes see image (9)

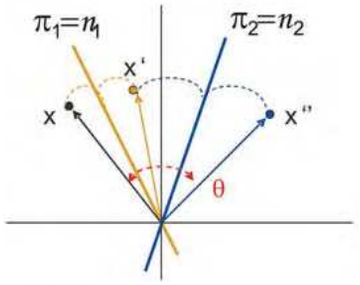

Fig. 9. Reflection about nonparallel planes.

$$Q' = R_\theta\underbrace{(\pi_2\pi_1)}Q\widetilde{R_\theta}\underbrace{(\pi_1^{-1}\pi_2^{-1})} \tag{62}$$

Or computing the conformal product of the normals of the planes.

$$R_\theta = n_2n_1 = Cos(\frac{\theta}{2}) - Sin(\frac{\theta}{2})l = e^{-\frac{\theta}{2}l} \tag{63}$$

With l = $n2$ ^ $n1$,and θ twice the angle between the planes π_2 and π_1. The screw motion *called motor* related to an arbitrary axis L is M = TRT

4.4 Kinematic Chains

The direct kinematics for serial robot arms is a succession of motors and it is valid for points, lines, planes, circles and spheres.

$$Q' = ni = 1\prod M_iQni = 1\prod\widetilde{M}_{n-i+1} \tag{64}$$

5. Ruled Surfaces

Conics, ellipsoids, helicoids, hyperboloid of one sheet are entities which can not be directly described in CGA, however, can be modeled with its multivectors. In particular, a ruled

surface is a surface generated by the displacement of a straight line (called generatrix) along a directing curve or curves (called directrices). The plane is the simplest ruled surface, but now we are interested in nonlinear surfaces generated as ruled surfaces. For example, a circular cone is a surface generated by a straight line through a fixed point and a point in a circle. It is well known that the intersection of a plane with the cone can generate the conics. See Figure 10. In [17] the cycloidal curves can be generated by two coupled twists. In this section we are going to see how these and other curves and surfaces can be obtained using only multivectors of CGA.

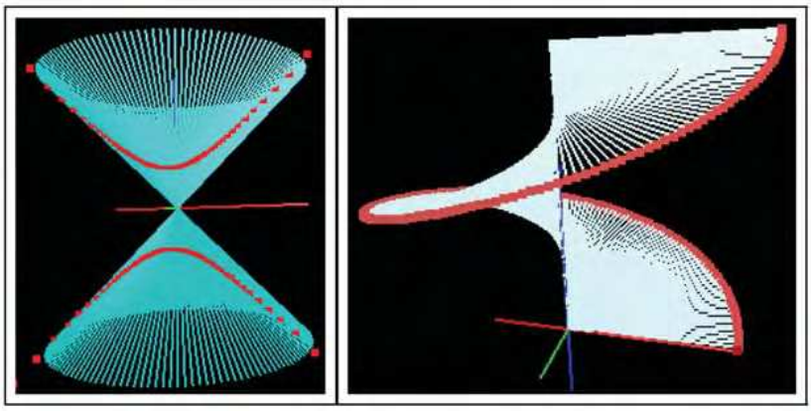

Fig. 10. (a) Hyperbola as the meet of a cone and a plane. (b) The helicoid is generated by the rotation and translation of a line segment. In CGA the *motor* is the desired multivector.

5.1 Cone and Conics

A circular cone is described by a fixed point v_0 (*vertex*), a dual circle $z_0 = a_0 \wedge a_1 \wedge a_2$ (*directrix*) and a rotor $R(\theta, l)$, $\theta \in [0, 2\pi)$ rotating the straight line $L(v_0,a_0) = v_0 \wedge a_0 \wedge e_\infty$, (*generatrix*) along the axis of the cone $l_0 = z_0 \cdot e_\infty$. Then, the cone w is generated as

$$w = R(\theta, l_0)\, L(v_0, a_0) \tilde{R}(\theta, l_0)\,,\ \theta \in [0, 2\pi) \tag{67}$$

A conic curve can be obtained with the *meet* (17) of a cone and a plane. See Figure 10(a).

5.2 Cycloidal Curves

The family of the cycloidal curves can be generated by the rotation and translation of one or two circles. For example, the cycloidal family of curves generated by two circles of radius $r_0$and r_1 are expressed by, see Figure 11, the motor

$$M = TR_1T^4R_2 \tag{68}$$

where

$$T = T\left((r_0 + r_1)(sin(\theta)e_1 + cos(\theta)e_2)\right) \tag{69}$$

$$R_1 = R_1\left(\frac{r_0}{r_1}\theta\right) \tag{70}$$

$$R_2 = R_2(\theta) \tag{71}$$

Then, each conformal point x is transformed as MxM.

5.3 Helicoid

We can obtain the ruled surface called helicoid rotating a ray segment in a similar way as the spiral of Archimedes. So, if the axis e_3 is the directrix of the rays and it is orthogonal to them, then the translator that we need to apply is a multiple of θ, the angle of rotation. See Figure 10(b).

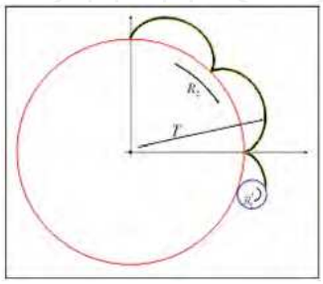

Fig. 11. The *motor* $M = TR_1T^*R_2$, defined by two rotors and one translator, can generate the family of the cycloidal curves varying the multivectors R_i and T.

5.4 Sphere and Cone

Let us see an example of how the algebra of incidence using CGA simplify the algebra. The intersection of a cone and a sphere in general position, that is, the axis of the cone does not pass through the center of the sphere, is the three dimensional curve of all the euclidean points (x, y, z) such that x and y satisfy the quartic equation

$$[x^2(1+\frac{1}{c^2}) - 2x_0x + y^2(1+\frac{1}{c^2}) - 2y_0y + x_0^2 + y_0^2 + z_0^2 - r^2]^2 = 4z_0^2(x^2+y^2)/c^2 \tag{72}$$

and x, y and z the quadratic equation

$$(x - x_0)^2+(y - y_0)^2+(z - z_0)^2= r. \tag{73}$$

See Figure 12. In CGA the set of points q of the intersection can be expressed as the meet (17) of the dual sphere s and the cone w, (67), defined in terms of its generatrix L,that is

$$q = (s^*) \cdot [R(\theta, l_0)\, L(v_0, a_0) \tilde{R}(\theta, l_0)],\ \theta \in [0, 2\pi)\,. \tag{74}$$

Thus, in CGA we only need (74) to express the intersection of a sphere and a cone, meanwhile in euclidean geometry it is necessary to use (72) and (73).

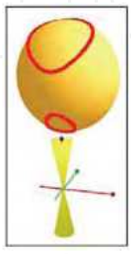

Fig. 12. Intersection as the *meet* of a sphere and a cone.

5.5 Hyperboloid of one sheet

The rotation of a line over a circle can generate a hyperboloid of one sheet. Figure 13(a).

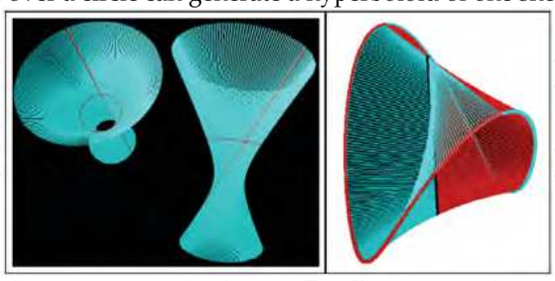

Fig. 13. (a) Hyperboloid as the rotor of a line. (b) The Plücker conoid as a ruled surface.

5.6 Ellipse and Ellipsoid

The ellipse is a curve of the family of the cycloid and with a translator and a dilator we can obtain an ellipsoid.

5.7 Plücker Conoid

The cylindroid or Pl¨ucker conoid is a ruled surface. See Figure 13(b). This ruled surface is like the helicoid where the translator parallel to the axis e_3 is of magnitude, a multiple of $cos(\theta)sin(\theta)$. The intersection curve of the conoid with a sphere will be obtained as the *meet* of both surfaces. Figure 14(a) and (b).

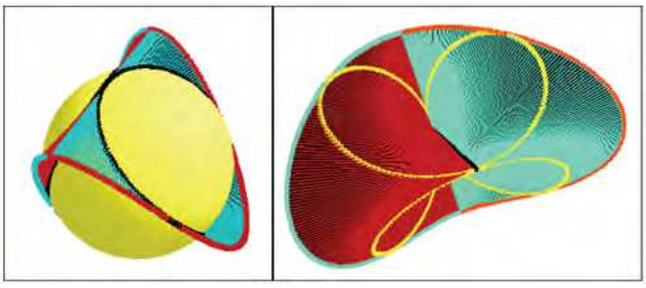

Fig. 14. The intersection between the Plücker conoid and a sphere.

6. Barrett Hand Forward Kinematics

The direct kinematics involves the computation of the position and orientation of the robot end-effector given the parameters of the joints. The direct kinematics can be easily computed if the lines of the screws' axes are given [2].

In order to introduce the kinematics of the Barrett Hand^{TM} we will show the kinematic of one finger, assuming that it is totally extended. Note that such an hypothetical position is not reachable in normal operation.

Let x_{1o}, x_{2o} be points-vectors describing the position of each joint of the finger and x_{3o} the end of the finger in the Euclidean space, see the Figure 15. If A_w, $A_{1,2,3}$ and D_w are denoting the dimensions of the finger's components

$$x_{1o} = A_{we1} + A_{1e2} + D_{we3}, \tag{75}$$

$$x_{2o} = A_{we1} + (A_1 + A_2)e_2 + D_{we3}, \tag{76}$$

$$x_{3o} = A_{we1} + (A_1 + A_2 + A_3)e_2 + D_{we3}. \tag{77}$$

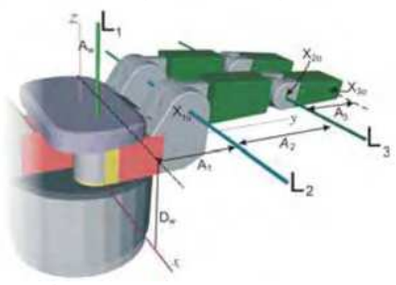

Fig. 15. Barrett hand hypothetical position.

Once we have defined these points it is quite simple to calculate the axes $L_{1o,2o,3o}$, which will be used as motor's axis. As you can see at the Figure 15,

$$L_{1o} = -A_w(e_2 \wedge e_\infty) + e_{12}, \tag{78}$$

$$L_{2o} = (x_{1o} \wedge e_1 \wedge e_\infty)\, I_c, \tag{79}$$

$$L_{3o} = (x_{2o} \wedge e_1 \wedge e_\infty)\, I_c. \tag{80}$$

When the hand is initialized the fingers moves away to the home position, this is the angle $\Phi 2 = 2.46^o$ by the joint two and the angle $\Phi 3 = 50^o$ degrees by the joint three. In order to move the finger from this hypothetical position to its home position the appropriate transformation is as follows:

$$M_{2o} = cos\,(\Phi 2/2) - sin(\Phi_2/2)L_{2o} \tag{81}$$

$$M_{3o} = cos\,(\Phi 3/2) - sin(\Phi_3/2)L_{3o}. \tag{82}$$

Once we have gotten the transformations, then we apply them to the points x_{2o} and x_{3o} in order to get the points x_2 and x_3 that represents the points in its home position, also the line L_3 is the line of motor axis in home position.

$$x_2 = M_{2o}x_{2o}\widetilde{M}_{2o}, \tag{83}$$

$$x_3 = M_{2o}M_{3o}x_{3o}\widetilde{M}_{3o}\widetilde{M}_{2o}, \tag{84}$$

$$L_3 = M_{2o}L_{3o}\widetilde{M}_{2o}. \tag{85}$$

The point $x_1 = x_{1o}$ is not affected by the transformation, the same for the lines $L_1 = L_{1o}$ and $L_2 = L_{2o}$ see Figure 16. Since the rotation angles of both axis $L2$ and $L3$ are related, we will use fractions of the angle q_1 to describe their individual rotation angles. The motors of each joint are computed using $\frac{2}{35}q_4$ to rotate around L_1, $\frac{1}{125}q_1$ around L_2 and $\frac{1}{375}q_1$ around $L3$, these specific angle coefficients where taken from the Barrett Hand user manual.

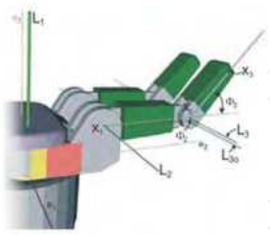

Fig. 16. Barrett hand at home position.

$$M_1 = cos(q_4/35) + sin(q_4/35)L_1, \tag{86}$$

$$M_2 = cos(q_1/250) - sin(q_1/250)L_2, \tag{87}$$

$$M_3 = cos(q_1/750) - sin(q_1/750)L_3. \tag{88}$$

The position of each point is related to the angles q_1 and q_4 as follows:

$$x_1' = M_1 x_1 \widetilde{M}_1, \tag{89}$$

$$x_2' = M_1 M_2 x_2 \widetilde{M}_2 \widetilde{M}_1, \tag{90}$$

$$x_3' = M_1 M_2 M_3 x_3 \widetilde{M}_3 \widetilde{M}_2 \widetilde{M}_1. \tag{91}$$

7. Application I: Following Geometric Primitives and Ruled Surfaces for Shape Understanding and Object Manipulation

In this section we will show how to perform certain object manipulation tasks in the context of conformal geometric algebra. First, we will solve the problem of positioning the gripper of the arm in a certain position of space disregarding the grasping plane or the gripper's alignment. Then, we will illustrate how the robotic arm can follow linear paths.

7.1 Touching a point

In order to reconstruct the point of interest, we make a back-projection of two rays extended from two views of a given scene (see Figure 17). These rays will not intersect in general, due to noise. Hence, we compute the directed distance between these lines and use the the middle point as target. Once the 3D point *pt* is computed with respect to the cameras' framework, we transform it to the arm's coordinate system.

Once we have a target point with respect to the arm's framework, there are three cases to consider. There might be several solutions (see Figs. 18.a and 19.a), a single solution (see Figure 18.b), or the point may be impossible to reach (Figure 19.b).

In order to distinguish between these cases, we create a sphere $\boldsymbol{S}_t = \boldsymbol{p}_t - \frac{1}{2}d_3^2 e_\infty$ centered at the point $\boldsymbol{p}_t$ and intersect it with the bounding sphere $\boldsymbol{S}_e = \boldsymbol{p}_0 - \frac{1}{2}(d_1 + d_2)^2 e_\infty$ of the other joints (see Figures 18.a and 18.b), producing the circle $z_s = \boldsymbol{S}_e \wedge \boldsymbol{S}_t$.

If the spheres $\boldsymbol{S}_t$ and $\boldsymbol{S}_e$ intersect, then we have a solution circle z_s which represents all the possible positions the point $\boldsymbol{p}_2$ (see Figure 18) may have in order to reach the target. If the spheres are tangent, then there is only one point of intersection and a single solution to the problem as shown in Figure 18.b.

Fig. 17. Point of interest in both cameras (*pt*).

If the spheres do not intersect, then there are two possibilities. The first case is that S_t is outside the sphere S_e. In this case, there is no solution since the arm cannot reach the point p_t as shown in Figure 19.b. On the other hand, if the sphere St is inside Se, then we have a sphere of solutions. In other words, we can place the point p_2 anywhere inside S_t as shown in Figure 19.a. For this case, we arbitrarily choose the upper point of the sphere St.

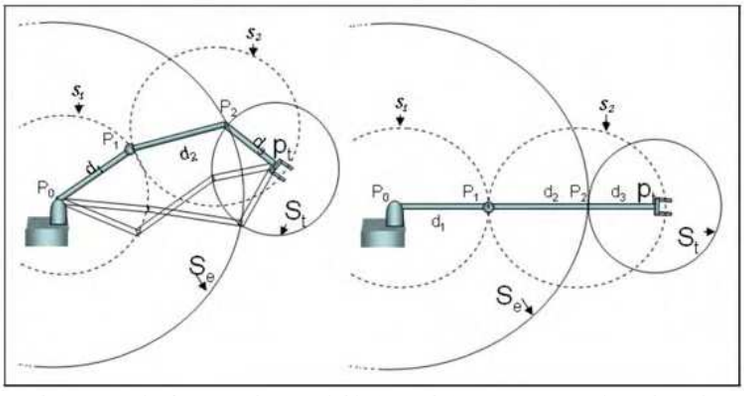

Fig. 18. a) S_e and S_t meet (infinite solutions) b) S_e and S_t are tangent (single solution).

In the experiment shown in Figure 20.a, the sphere St is placed inside the bounding sphere Se, therefore the point selected by the algorithm is the upper limit of the sphere as shown in Figures 20.a and 20.b. The last joint is completely vertical.

7.2 Line of intersection of two planes

In the industry, mainly in the sector dedicated to car assembly, it is often required to weld pieces. However, due to several factors, these pieces are not always in the same position, complicating this task and making this process almost impossible to automate. In many cases the requirement is to weld pieces of straight lines when no points on the line are available. This is the problem to solve in the following experiment.

Since we do not have the equation of the line or the points defining this line we are going to determine it via the intersection of two planes (the welding planes). In order to determine each plane, we need three points. The 3D coordinates of the points are

triangulated using the stereo vision system of the robot yielding a configuration like the one shown in Figure 21.

Once the 3D coordinates of the points in space have been computed, we can find now each plane as $\pi^* = \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 \wedge e_\infty$, and $\pi'^* = \boldsymbol{x}'_1 \wedge \boldsymbol{x}'_2 \wedge \boldsymbol{x}'_3 \wedge e'_\infty$. The line of intersection is computed via the *meet* operator $\boldsymbol{l} = \pi' \cap \pi$. In Figure 22.a we show a simulation of the arm following the line produced by the intersection of these two planes.

Once the line of intersection $\boldsymbol{l}$ is computed, it suffices with translating it on the plane $\psi = \boldsymbol{l}^* \wedge e_2$ (see Figure 22.b) using the translator $\boldsymbol{T}_1 = 1+\gamma e_2 e_\infty$, in the direction of e_2 (the y axis) a distance γ. Furthermore, we build the translator $\boldsymbol{T}_2 = 1+d_3 e_2 e_\infty$ with the same direction ($e2$), but with a separation d_3 which corresponds to the size of the gripper. Once the translators have been computed, we find the lines $\boldsymbol{l}'$ and $\boldsymbol{l}''$ by translating the line $\boldsymbol{l}$ with $\boldsymbol{l}' = \boldsymbol{T}_1 \boldsymbol{l} \boldsymbol{T}_1^{-1}$, and $\boldsymbol{l}'' = \boldsymbol{T}_2 \boldsymbol{l}' \boldsymbol{T}_2^{-1}$.

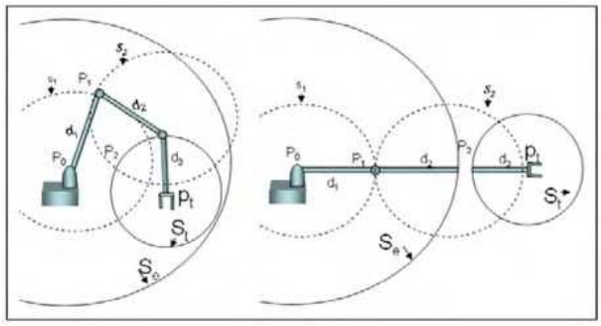

Fig. 19. a) *St* inside *Se* produces infinite solutions, b) *St* outside *Se*, no possible solution.

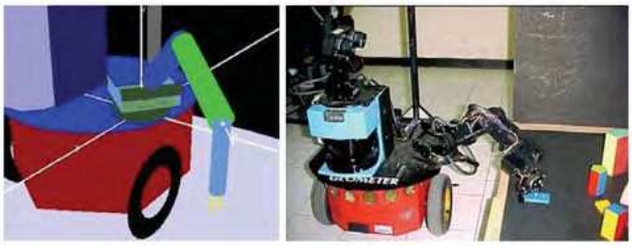

Fig. 20. a) Simulation of the robotic arm touching a point. b) Robot "Geometer" touching a point with its arm.

The next step after computing the lines, is to find the points $\boldsymbol{p}_t$ and $\boldsymbol{p}_2$ which represent the places where the arm will start and finish its motion, respectively. These points were given manually, but they may be computed with the intersection of the lines $\boldsymbol{l}'$ and $\boldsymbol{l}''$ with a plane that defines the desired depth. In order to make the motion over the line, we build a translator $\boldsymbol{T}_L = 1-\Delta_L l e_\infty$ with the same direction as l as shown in Figure 22.b. Then, this translator is applied to the points $\boldsymbol{p}_2 = \boldsymbol{T}_L \boldsymbol{p}_2 \boldsymbol{T}_L^{-1}$ and $\boldsymbol{p}_t = \boldsymbol{T}_L \boldsymbol{p}_t \boldsymbol{T}_L^{-1}$ in an iterative fashion to yield a displacement ΔL on the robotic arm.

By placing the end point over the lines and p_2 over the translated line, and by following the path with a translator in the direction of l we get a motion over l as seen in the image sequence of Figure 23.

7.3 Following a spherical path

This experiment consists in following the path of a spherical object at a certain fixed distance from it. For this experiment, only four points on the object are available (see Figure 24.a).

After acquiring the four 3D points, we compute the sphere $S^* = x^1 \wedge x^2 \wedge x^3 \wedge x^4$. In order to place the point p_2 in such a way that the arm points towards the sphere, the sphere was expanded using two different dilators. This produces a sphere that contains S^* and ensures that a fixed distance between the arm and S^* is preserved, as shown in Figure 24.b.

The dilators are computed as follows

$$\boldsymbol{D}_\gamma = e^{-\frac{1}{2}\ln(\frac{\gamma+\rho}{\rho})\boldsymbol{E}}, \tag{92}$$

$$\boldsymbol{D}_d = e^{-\frac{1}{2}\ln(\frac{d_3+\gamma+\rho}{\rho})\boldsymbol{E}}. \tag{93}$$

The spheres S_1 and S_2 are computed by dilating St:

$$\boldsymbol{S}_1 = \boldsymbol{D}_\gamma \boldsymbol{S}_t \boldsymbol{D}_\gamma^{-1}, \tag{94}$$

$$\boldsymbol{S}_2 = \boldsymbol{D}_d \boldsymbol{S}_t \boldsymbol{D}_d^{-1}. \tag{95}$$

Guiding lines for the robotic arm produced by the intersection, *meet*, of the planes and vertical translation.

We decompose each sphere in its parametric form as

$$\boldsymbol{p}_t = \boldsymbol{M}_1(\varphi)\boldsymbol{M}_1(\phi)\boldsymbol{p}_{s_1}\boldsymbol{M}_1^{-1}(\phi)\boldsymbol{M}_1^{-1}(\varphi), \tag{96}$$

$$\boldsymbol{p}_2 = \boldsymbol{M}_2(\varphi)\boldsymbol{M}_2(\phi)\boldsymbol{p}_{s_2}\boldsymbol{M}_2^{-1}(\phi)\boldsymbol{M}_2^{-1}(\varphi). \tag{97}$$

Where p_s is any point on the sphere. In order to simplify the problem, we select the upper point on the sphere. To perform the motion on the sphere, we vary the parameters φ and φ and compute the corresponding pt and p_2 using equations (96) and (97). The results of the simulation are shown in Figure 25.a, whereas the results of the real experiment can be seen in Figures 25.b and 25.c.

Fig. 21. Images acquired by the binocular system of the robot "Geometer" showing the points on each plane.

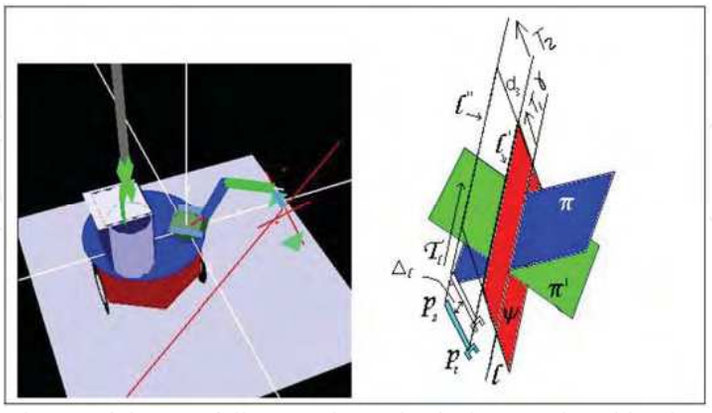

Fig. 22. a. Simulation of the arm following the path of a line produced by the intersection of two planes. b.

7.4 Following a 3D-curve in ruled surfaces

As another simulated example using ruled surfaces consider a robot arm laser welder. See Figure 26. The welding distance has to be kept constant and the end-effector should follow a 3D-curve w on the ruled surface guided only by the directrices d_1, d_2 and a guide line L. From the generatrices we can always generate the nonlinear ruled surface, and then with the meet with another surface we can obtain the desired 3D-curve. We tested our simulations with several ruled surfaces, obtaining expressions of high nonlinear surfaces and 3D-curves, that with the standard vector and matrix analysis it would be very difficult to obtain them.

Fig. 23. Image swquence of a linear-path motion.

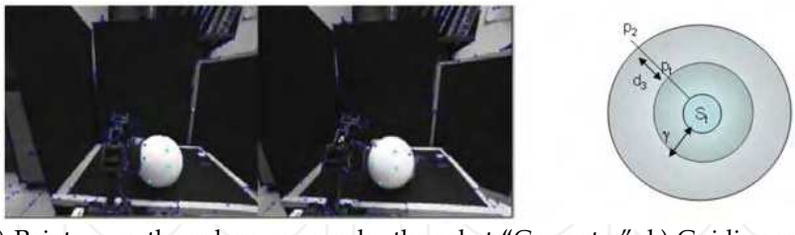

Fig. 24. a) Points over the sphere as seen by the robot "Geometer". b) Guiding spheres for the arm's motion.

8. Aplications II: Visual Grasping Identification

In our example considering that the cameras can only see the surface of the observed objects, thus we will consider them as bi-dimensional surfaces which are embedded in a 3D space, and are described by the function

$$\vec{H}(s,t) = h_x(s,t)e_1 + h_y(s,t)e_2 + h_z(s,t)e_3 \tag{98}$$

where s and t are real parameters in the range [0, 1]. Such parameterization allows us to work with different objects like points, conics, quadrics, or even more complex objects like cups, glasses, etc. The table 2 shows some parameterized objects.

Particle	$\vec{H} = 3e_1 + 4e_2 + 5e_3$
Cylinder	$\vec{H} = cos(t)e_1 + sin(t)e_2 + se_3$
Plane	$\vec{H} = te_1 + se_2 + (3s + 4t + 2)e_3$

Table 2. Parameterized Objects.

Due to that our objective is to grasp such objects with the Barrett Hand, we must consider that it has only three fingers, so the problem consists in finding three "touching points" for which the system is in equilibrium during the grasping; this means that the sum of the forces equals to zero, and also the sum of the moments. For this case, we consider that there exists friction in each "touching point".

If the friction is being considered, we can claim that over the surface $H(s, t)$ a set of forces exist which can be applied. Such forces are inside a cone which have the normal $N(s, t)$ of the surface as its axis (as shown in Fig. 27). Its radius depends on the friction's coefficient $\|F - F_n\| \leq -\mu(|F_n|)$, where $Fn = (F \cdot N(s, t))N(s, t)$ is the normal component of F. The angle θ for the incidence of F with respect to the normal can be calculated using the wedge product, and should be smaller than a fixed $\theta\mu$

$$\frac{\|F \wedge N(s,t)\|}{F \cdot N(s,t)} \leq tan(\theta_\mu) \tag{99}$$

Fig. 25. a) Simulation of the motion over a sphere. b) and c) Two of the images in the sequence of the real experiment.

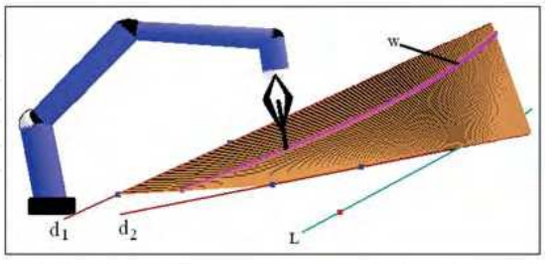

Fig. 26. A laser welding following a 3D-curve w on a ruled surface defined by the directrices d_1 and d_2. The 3D-curve w is the *meet* between the ruled surface and a plane containing the line L.

We know the surface of the object, so we can compute its normal vector in each point using

$$N(s,t) = \left(\frac{\partial \vec{H}(s,t)}{\partial s} \wedge \frac{\partial \vec{H}(s,t)}{\partial t} \right) I_e \tag{100}$$

In surfaces with lower friction, the angle θ is very small, then the value of F tends to its projection over the normal ($F \approx Fn$). To maintain equilibrium, the sum of the forces must be zero $(\sum_{i=1}^{3} \|F_n\| N(s_i,t_i) = 0)$. This fact restricts the points over the surface in which it can be applied the forces. This number of points is even more reduced if we are confronted with the case when considering the unit normal $(\sum_{i=1}^{3} N(s_i,t_i) = 0)$ the forces over the object are equal. Additionally, to maintain the equilibrium, it must be accomplished that the sum of the moments is zero

$$\sum_{i=1}^{3} H(s,t) \wedge N(s,t) = 0 \tag{101}$$

The points on the surface having the same directed distance to the center of mass of the object fulfill $H(s, t) \wedge N(s, t) = 0$. Due to the normal in such points crosses the center of mass (C_m), it does not produce any moment. Before determining the external and internal points, we must compute the center of mass as follows

$$C_m = \int_0^1 \int_0^1 \vec{H}(s,t) ds dt \tag{102}$$

Once that C_m is calculated we can establish next constraint

$$(H(s, t) - C_m) \wedge N(s, t) = 0 \tag{103}$$

The values s and t satisfying (103) form a subspace called *grasping space*. They accomplish that the points represented by $H(s, t)$ are critical on the surface (being maximums, minimums or inflections). In this work we will not consider other grasping cases like when they do not utilize extreme points other when friction cones are being considered. This issues will be treated in future work. The equation (103) is hard to fulfill due to the noise, and it is necessary to consider a cone of vectors. So, we introduce an angle called a,

$$\frac{\|(H(s,t) - C_m) \wedge N(s,t)\|}{(H(s,t) - C_m) \cdot N(s,t)} \leq tan(\alpha) \tag{104}$$

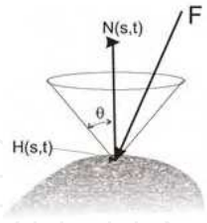

Fig. 27. The friction cone.

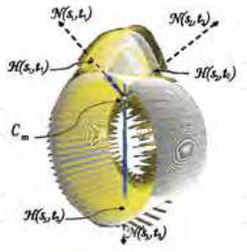

Fig. 28. Object and his normal vectors.

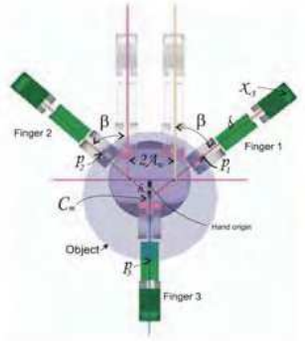

Fig. 29. Object relative position.

We use equation (104) instead of (103), because it allows us to deal with errors or data lost. The constraint imposing that the three forces must be equal is hard to fulfill because it implies that the three points must be symmetric with respect to the mass center. When such points are not present, we can relax the constraint to allow that only two forces are equal in order to fulfill the hand's kinematics equations. Then, the normals $N(s_1, t_1)$ and $N(s_2, t_2)$ must be symmetric with respect to $N(s_3, t_3)$.

$$N(s_3, t_3)N(s_1, t_1)N(s_3, t_3)^{-1} = N(s_2, t_2) \tag{105}$$

Once the three grasping points ($P_1 = H(s_1, t_1)$, $P_2 = H(s_2, t_2)$, $P_3 = H(s_3, t_3)$) are calculated, it is really easy to determine the angles at the joints in each finger. To determine the angle of the spread ($q_4 = \beta$) for example we use

$$\cos\beta = \frac{(p_1 - C_m)\cdot(C_m - p_3)}{|p_1 - c_m|\,|C_m - p_3|},\ \sin\beta = \frac{|(p_1 - C_m)\wedge(C_m - p_3)|}{|p_1 - c_m|\,|C_m - p_3|}, \tag{106}$$

or it is possible to implement a control law which will allow to move the desired finger without the need of solving any kind of inverse kinematics equations [1]. Given the differential kinematics equation

$$\dot{X}'_3 = \begin{bmatrix} \frac{1}{125}X'_3\cdot L'_2 + \frac{1}{375}X'_3\cdot L'_3 & -\frac{2}{35}X'_3\cdot L'_1 \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_4, \end{bmatrix} \tag{107}$$

If we want to reach the point $H(s_1, t_1)$, we require that the suitable velocity at the very end of the finger should be proportional to the error at each instance $V_i = -0.7(X'_3 - H(s_1, t_1))$. This velocity is mapped into the phase space by means of using the Jacobian inverse. Here we use simply the pseudo-inverse with $j_1 = \frac{1}{125}X'_3\cdot L'_2 + \frac{1}{375}X'_3\cdot L'_3$ and $j_2 = -\frac{2}{35}X'_3\cdot L'_1$

Applying this control rule, one can move any of the fingers at a desired position above an object, so that an adequate grasp is accomplish.

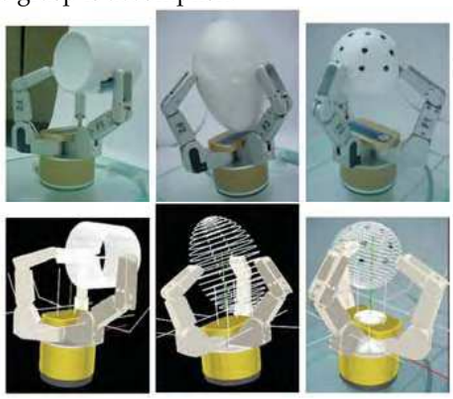

Fig. 30. Grasping some objects.

9. Conclusion

In this chapter the authors have used a single non–standard mathematical framework, the Conformal Geometric Algebra, in order to simplify the set of data structures that we usually use with the traditional methods. The key idea is to define and use a set of products in CGA that will be enough to generate conformal transformations, manifolds as ruled surfaces and develop incidence algebra operations, as well as solve equations and obtain directed distances between different kinds of geometric primitives. Thus, within this approach, all those different mathematical entities and tasks can be done simultaneously, without the necessity of abandoning the system.

Using conformal geometric algebra we even show that it is possible to find three grasping points for each kind of object, based on the intrinsic information of the object. The hand's kinematic and the object structure can be easily related to each other in order to manage a natural and feasible grasping where force equilibrium is always guaranteed. These are only some applications that could show to the robotic and computer vision communities the useful insights and advantages of the CGA, and we invite them to adopt, explore and implement new tasks with this novel framework, expanding its horizon to new possibilities for robots equipped with stereo systems, range data, laser, omnidirectional and odometry.

10. References

[1] Carlos Canudas de Wit, Georges Bastin, Bruno Siciliano. (1996) Theory of Robot Control, Springer.

[2] Bayro-Corrochano E. and Kähler D (2000). Motor Algebra Approach for Computing the kinematics of robot manipulators. *Journal of Robotics Systems*, 17(9):495-516.

[3] Bayro-Corrochano E. 2001. *Geometric Computing for Perception Action Systems*, Springer Verlag, Boston.

[4] Bayro-Corrochano E. 2005. Robot perception and action using conformal geometric algebra. In Handbook of Geometric Computing. Applications in Pattern

Recognition, Computer Vision, Neuralcomputing and Robotics. Eduardo Bayro-Corrochano (Ed.), Springer Verlag, Heidelberg, Chap.13, pp. 405-458.

[5] 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 27-29, pp. 143-149.

[6] Bayro-Corrochano, E., Lasenby, J. and Sommer, G. 1996. Geometric Algebra: a framework for computing point and line correspondences and projective structure using n–uncalibrated cameras. *Proceedings of the International Conference on Pattern Recognition (ICPR'96), Vienna, August 1996.*, Vol.I, pp. 393-397.

[7] Laseby J. and Bayro–Corrochano E. 1999. Analysis and Computation of Projective Invariants from Multiple Views in the Geometric Algebra Framework. In Special Issue on Invariants for Pattern Recognition and Classification, ed. M.A. Rodrigues. Int. Journal of Pattern Recognition and Artificial Intelligence, Vol 13, No 8, December 1999, pp. 1105–1121.

[8] Bayro–Corrochano E., Daniilidis K. and Sommer G. 2000. Motor algebra for 3D kinematics. The case of the hand–eye calibration. *Journal of Mathematical Imaging and Vision*, vol. 13, pag. 79-99, 2000.

[9] Hestenes, D. 1966. Space–Time Algebra. *Gordon and Breach*.

[10] Hestenes, D. and Sobczyk, G. 1984. Clifford Algebra to Geometric Calculus: A unified language for mathematics and physics. *D. Reidel*, Dordrecht.

[11] Hestenes D., Li H. and Rockwood A. 2001. New algebraic tools for classical geometry. *In Geometric Computing with Clifford Algebra*, G. Sommer (Ed.). Springer-Verlag, Berlin Heidelberg, Chap.I, pp. 3-23.

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

[13] Lasenby, J., Lasenby, A.N., Lasenby, Doran, C.J.L and Fitzgerald, W.J. 1998. 'New geometric methods for computer vision – an application to structure and motion estimation'. *International Journal of Computer Vision*, 26(3), 191-213.

[14] Csurka G. and Faugeras O. Computing three dimensional project invariants from a pair of images using the Grassmann–Cayley algebra. 1998. *Journal of Image and Vision Computing*, 16, pp. 3-12.

[15] Lounesto P. *Clifford Algebra and Spinors*, 2nd ed. Cambridge University Press, Cambridge, UK, 2001.

[16] Needham T. *Visual Complex Analysis*. Oxford University Press. Reprinted 2003.

[17] Rosenhahn B., Perwass C., Sommer G. *Pose estimation of 3D free-form contours*. Technical Report 0207, University Kiel, 2002.

[18] J.M. Selig. *Clifford algebra of points, lines and planes*. South Bank University, School of Computing, Information Technology and Maths, Technical Report SBU–CISM–99–06, 1999.

[19] White N. Geometric applications of the Grassman–Cayley algebra. In J.E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*, CRC Press, Floridam 1997.

Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

ISBN 3-86611-283-1

Hard cover, 704 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, February, 2007

Published in print edition February, 2007

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well es those handling global navigation aspects of a single robot and even for a group of robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Eduardo Bayro-Corrochano, Luis Eduardo Falcon-Morales and Julio Zamora-Esquivel (2007). Visually Guided Robotics Using Conformal Geometric Computing, Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), ISBN: 3-86611-283-1, InTech, Available from: http://www.intechopen.com/books/mobile_robots_perception_navigation/visually_guided_robotics_using_conformal_geometric_computing

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen