

# Advanced Geometric Approach for Graphics and Visual Guided Robot Object Manipulation

Dietmar Hildenbrand  
*Interactive Graphics Systems Group*  
*University of Technology*  
*Darmstadt, Germany*  
*dhilden@gris.informatik.tu-darmstadt.de*

Eduardo Bayro-Corrochano and Julio Zamora  
*Computer Science Department*  
*Centro de Investigación y de Estudios Avanzados*  
*Guadalajara, Jalisco 44550, Mexico*  
*edb,jzamora@gdl.cinvestav.mx*

**Abstract**—This paper presents an approach to deal with standard tasks of computer animations and robotics based on Conformal Geometric Algebra. We will show that this algebra is very well suitable for applications of all kind of robot manipulator kinematics, representation and visualization and object robot manipulation. Due to its geometric intuitiveness and simplicity Conformal Geometric Algebra appears to be a promising mathematical tool for building intelligent man-machine interfaces.

**Index Terms**—Inverse Kinematics, Geometric Algebra, Grasping

## I. FOUNDATIONS OF CONFORMAL GEOMETRIC ALGEBRA

**Blades** are the basic computational elements and the basic geometric entities of the Geometric Algebra. For example, the 5D Conformal Geometric Algebra provides a great variety of basic geometric entities to compute with. It consists of blades with **grades** 0, 1, 2, 3, 4 and 5, whereby a scalar is a **0-blade** (blade of grade 0). There exists only one element of grade five in the Conformal Geometric Algebra. It is therefore also called the pseudoscalar. A linear combination of k-blades is called a **k-vector** (also called vectors, bivectors, trivectors ... ). Furthermore, a linear combination of blades of different grades is called a multivector. Multivectors are the general elements of a Geometric Algebra.

### A. Spheres and Planes

The equation of a sphere of radius  $\rho$  centered at point  $p_e \in \mathbb{R}^n$  can be written as

$$(x_e - p_e)^2 = \rho^2. \quad (1)$$

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

$$x_c \cdot p_c = -\frac{1}{2}\rho^2. \quad (2)$$

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

$$x_c \cdot (p_c - \frac{1}{2}\rho^2 e_\infty) = 0. \quad (3)$$

Which finally yields the simplified equation for the sphere as  $s = p_c - \frac{1}{2}\rho^2 e_\infty$ . Alternatively, the dual of the sphere is represented as 4-vector  $s^* = sI^{-1}$ . The advantage

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

$$s^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge x_{c_4}. \quad (4)$$

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

$$\pi^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge e_\infty. \quad (5)$$

So that  $\pi$  becomes in the standard form

$$\pi = I^{-1}\pi^* = n + de_\infty \quad (6)$$

Where  $n$  is the normal vector and  $d$  represents the Hesse distance.

### B. Circles and Lines

A circle  $z$  can be regarded as the intersection of two spheres  $s_1$  and  $s_2$  as  $z = (s_1 \wedge s_2)$ . The dual form of the circle (in 3D) can be expressed by three points lying on it as

$$z^* = x_{c_1} \wedge x_{c_2} \wedge x_{c_3}. \quad (7)$$

Similar to the case of planes, lines can be defined by circles passing through the point at infinity as:

$$L^* = x_{c_1} \wedge x_{c_2} \wedge e_\infty. \quad (8)$$

The standard form of the line (in 3D) can be expressed by

$$L = l + e_\infty(t \cdot l), \quad (9)$$

the line in the standard form is a bivector, and it has six parameters (Plucker coordinates), but just four degrees of freedom.

TABLE I  
STANDARD REPRESENTATION OF THE CONFORMAL GEOMETRIC ENTITIES

entity	standard representation	grade
Point	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$	1
Sphere	$s = P - \frac{1}{2}r^2 e_\infty$	1
Plane	$\pi = \mathbf{n} + de_\infty$	1
Circle	$z = s_1 \wedge s_2$	2
Line	$l = \pi_1 \wedge \pi_1$	2
Point Pair	$P_p = s_1 \wedge s_2 \wedge s_3$	3

Table I lists the standard representation of the geometric entities of the Conformal Geometric Algebra. Please find

details in [5]. The  $\{s_i\}$  represent different spheres and the  $\{\pi_i\}$  different planes. A sphere is represented with the help of its center point  $P$  and its radius  $r$ . Note that the representation of a point is simply a sphere with radius zero. Similarly, a plane is a sphere with infinite radius. A circle is generated with the help of the outer product  $\wedge$  of two spheres. In this table  $\mathbf{x}$  and  $\mathbf{n}$  are marked bold since they represent 3D entities (  $\mathbf{x}$  as the original 3D point that has to be extended to a 5D-vector with additional 2 base vectors  $e_\infty$  and  $e_0$  representing the point at infinity and the origin,  $\mathbf{n}$  as the 3D normal vector of the plane  $\pi$  ).

Table II lists the **dual representation** of the Conformal

TABLE II

DUAL REPRESENTATION OF THE CONFORMAL GEOMETRIC ENTITIES

entity	dual representation	grade
Sphere	$s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4$	4
Plane	$\pi^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty$	4
Circle	$z^* = x_1 \wedge x_2 \wedge x_3$	3
Line	$l^* = x_1 \wedge x_2 \wedge e_\infty$	3
Point Pair	$P_p^* = x_1 \wedge x_2$	2

Geometric Algebra. E. g. a sphere is represented with the help of 4 points that lie on it.

Both representations are dual to each other. It depends on the application which representation is more convenient to use.

## II. THE INVERSE KINEMATICS OF A HUMAN-ARM-LIKE KINEMATIC CHAIN

The following algorithm has been originally developed for a computer animation application with human figures but can be used for human-arm-like robots as well.

Our model of the human arm is a 7 DOF kinematic chain according to [7] with 3 degrees of freedom ( $\theta_1, \theta_2, \theta_3$ ) at the shoulder, 1 degree of freedom at the elbow ( $\theta_4$ ) and 3 degrees of freedom at the wrist ( $\theta_5, \theta_6, \theta_7$ ).

While in former analytic algorithms a lot of mathematical knowledge about trigonometry, rotation matrices etc. has to be available, in our advanced approach only some basic operations with basic entities are needed.

### A. compute the swivel plane

Our goal is to reach the chosen point  $p_w$  with the wrist. An arbitrary orientation of the gripper is not investigated in this paper. Please refer to the algorithm of paper [3] for the inverse kinematics a robot with 5 DOF also handling the orientation and grasping of the gripper.

According to [7], the swivel angle can be used as one free degree of redundancy. It is denoted by  $\phi$ .

The swivel plane is the plane rotated by  $\phi$  around the line  $l_{sw}$  through shoulder ( at the origin ) and  $p_w$ .

$$l_{sw} = (e_0 \wedge p_w \wedge e_\infty)^* \quad (10)$$

Note that the dual of a line is defined with the help of 2 points and the point at infinity ( see table II ).

The rotation operator  $R_{swivel}$  is defined by

$$R_{swivel} = e^{-\frac{1}{2}\phi l_{sw}} \quad (11)$$

For details please refer to [5].

Initially the swivel plane is defined with the help of the origin, the point  $p_w$ , the point  $P(e_2)$  and the point at infinity ( see table II ).

$$\pi_{swivel} = (e_0 \wedge P(e_2) \wedge p_w \wedge e_\infty)^* \quad (12)$$

with  $P(e_2)$  being the 5D-vector representation of the 3D point  $\mathbf{x}=(0,1,0)$  ( see table I )

$$P(e_2) = e_2 + \frac{1}{2}e_\infty + e_0 \quad (13)$$

Its final rotated location is

$$\pi_{swivel} = R_{swivel} \pi_{swivel} R_{swivel}^{\sim} \quad (14)$$

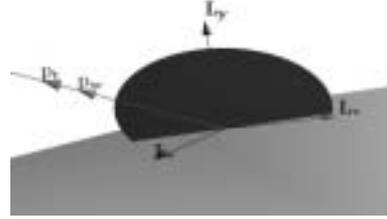


Fig. 1. swivel plane

### B. Step 1 : the elbow point $p_e$

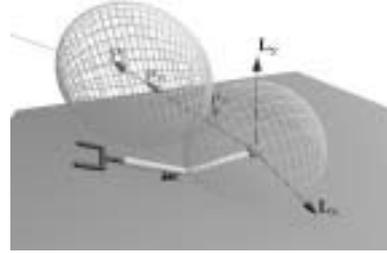


Fig. 2. Step 1

With the help of the two spheres

$$S_1 = p_w - \frac{1}{2}L_2^2 e_\infty \quad (15)$$

and

$$S_2 = e_0 - \frac{1}{2}L_1^2 e_\infty \quad (16)$$

with center points  $p_w$  and  $e_0$  and radii  $L_2, L_1$  we are able to compute the circle determining all the possible locations of the elbow as the intersection of the spheres ( see table I ).

$$C_e = S_1 \wedge S_2 \quad (17)$$

The intersection with the swivel plane delivers the point pair

$$P_p = C_e \wedge \pi_{swivel} \quad (18)$$

and we decide for one of the two possible elbow points. Please refer to [5] for details about extracting points out of point pairs.

### C. Step 2 : the elbow angle $\theta_4$

The elbow angle  $\theta_4$  is computed with the help of the line  $l_{se}$  through the shoulder and the elbow

$$l_{se} = (e_0 \wedge p_e \wedge e_\infty)^* \quad (19)$$

and the line  $l_{ew}$  through the shoulder and the wrist

$$l_{ew} = (p_e \wedge p_w \wedge e_\infty)^* \quad (20)$$

$$\theta_4 = \text{angle}(l_{se}^*, l_{ew}^*) \quad (21)$$

according to

$$\text{angle}(o_1^*, o_2^*) = \arccos \frac{o_1^* \cdot o_2^*}{|o_1^*| |o_2^*|} \quad (22)$$

### D. Step 3 : elevate to the rotation plane including the elbow point $p_e$

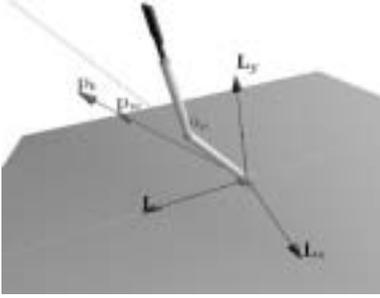


Fig. 3. Step 3

Now we have to elevate the arm in order that the rotation plane for the next angle includes the elbow point  $p_e$ . This is done with the help of the line in z direction

$$l_z = (e_0 \wedge P(e_3) \wedge e_\infty)^* \quad (23)$$

and the projection of the line  $l_{se}$  onto the plane in y and z direction ( with normal vector  $e_1$  and zero distance to the origin, see table II )

$$\pi_{yz} = e_1 \quad (24)$$

$$l_{proj} = \text{proj}(l_{se}, \pi_{yz}) \quad (25)$$

$$\theta_1 = \text{angle}(l_z^*, l_{proj}^*) \quad (26)$$

### E. Step 4 : rotate until the elbow position matches

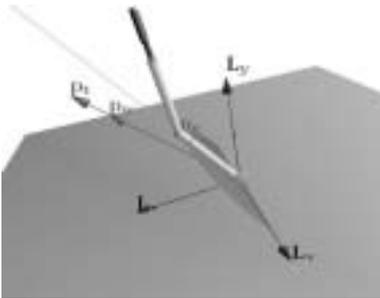


Fig. 4. Step 4

The angle  $\theta_2$  is computed with the help of the line in x direction and the line  $l_{se}$  through shoulder and elbow.

$$l_x = (e_0 \wedge P(e_1) \wedge e_\infty)^* \quad (27)$$

$$\theta_2 = \text{angle}(l_x^*, l_{se}^*) \quad (28)$$

### F. Step 5 : rotate until the wrist location is reached

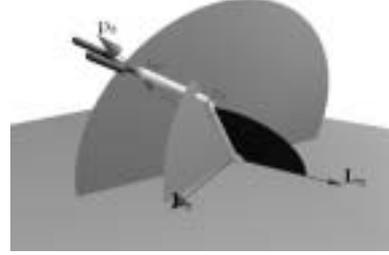


Fig. 5. Step 5

$\theta_3$  is computed with the help of the y-z-plane rotated by the two angles  $\theta_1$  and  $\theta_2$  and the swivel plane.

$$\pi_{yz2} = R_{\theta_1, \theta_2} \pi_{yz} \tilde{R}_{\theta_1, \theta_2} \quad (29)$$

$$\theta_3 = \text{angle}(\pi_{yz2}^*, \pi_{swivel}^*) \quad (30)$$

## III. THE INVERSE KINEMATICS OF A PAN-TILT UNIT.

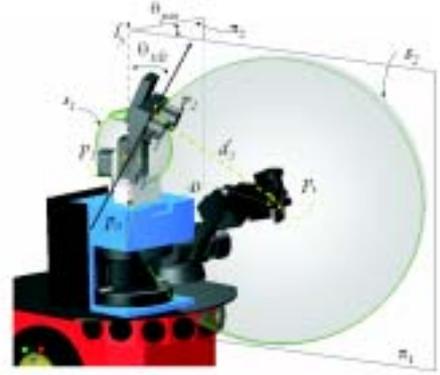


Fig. 6. Point  $p_2$  given by intersection of the plane  $\pi_1$  and the spheres  $s_1$  and  $s_2$ .

The problem consists in determining the angles  $\theta_{tilt}$  and  $\theta_{pan}$  of stereo-head, so that the cameras fix at the point  $p_t$ . We will now show how we find the values of  $\theta_{pan}$  and  $\theta_{tilt}$  using the conformal approach. The problem will be divided in three steps to be solved.

**Step 1:** Determine the point  $p_2$ .

When the  $\theta_{tilt}$  rotates and the base rotates ( $\theta_{pan}$ ) around the  $l_y$  (see Fig.6), the point  $p_2$  describes a sphere  $s_1$ . This sphere has center at the point  $p_1$  and radius  $d_2$ .

$$S_1 = p_1 - \frac{d_2^2}{2} e_\infty \quad (31)$$

Also the point  $p_t$  can be locked from every point around it, that means the point  $p_2$  is in the sphere:

$$S_2 = p_t - \frac{d_3^2}{2} e_\infty \quad (32)$$

Where  $d_3$  is the distance between point  $p_t$  and the cameras, and we can calculate  $d_3$  using a Pythagorean theorem  $d_3^2 = D^2 - d_2^2$ , where  $D$  is the direct distance between  $p_t$  and  $p_1$ .

We have restricted the position of the point  $p_2$ , but there is another restriction: the vector going from the  $p_2$  to the point  $p_t$  must lie on the plane  $\pi_1$  generated by the  $l_y$  axis ( $l_y^* = p_0 \wedge p_1 \wedge e_\infty$ ) and the point  $p_t$ , as we can see in Fig. 6. So that  $p_2$  can be determined by intersecting the plane  $\pi_1$  with the spheres  $s_1$  and  $s_2$  as follows

$$\pi_1^* = l_y^* \wedge p_t, \quad (33)$$

$$P_{p2} = s_1 \wedge \pi_1 \wedge s_2. \quad (34)$$

**Step 2:** Determine the lines and planes.

Once  $p_2$  have been determined, the line  $l_2$  and the plane  $\pi_2$  can be defined. This line and plane will be useful to calculate the angles  $\theta_{tilt}$  and  $\theta_{pan}$ .

$$l_2^* = p_1 \wedge p_2 \wedge e_\infty, \quad (35)$$

$$\pi_2^* = l_2^* \wedge e_3. \quad (36)$$

**Step 3:** Find the angles  $\theta_{tilt}$  and  $\theta_{pan}$ .

Once we have all the geometric entities, the computation of the angles is a trivial step.

$$\cos(\theta_{pan}) = \frac{\pi_1^* \cdot \pi_2^*}{|\pi_1^*| |\pi_2^*|}, \quad \cos(\theta_{tilt}) = \frac{l_1^* \cdot l_y^*}{|l_1^*| |l_y^*|}. \quad (37)$$

#### IV. 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 straight lines when no points on the line are available. This is the problem to solve in the following experiment.

If we do not have points on the line of interest, then we find this line via the intersection of two planes (the welding planes). In order to determine each plane, we need three points. These points are triangulated by standard means yielding a configuration like the one shown in Fig. 7.



Fig. 7. Images acquired by the binocular system of the robot "Geometer" showing the points on each plane.

Once the points in space have been triangulated, we can find each plane with  $\pi^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty$ , and  $\pi'^* = x'_1 \wedge x'_2 \wedge x'_3 \wedge e'_\infty$ . The line of intersection is computed with  $l = \pi' \wedge \pi$ . In Fig. 8 we show a simulation of the arm

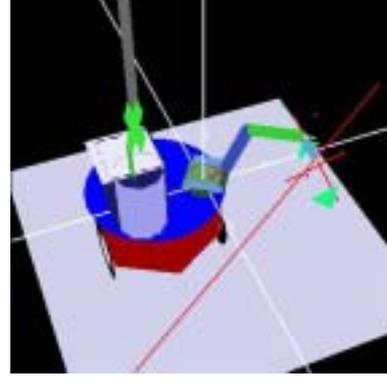


Fig. 8. Simulation of the arm following the path of a line produced by the intersection of two planes.

following the line produced by the intersection of these two planes.

Once the line of intersection  $l$  is computed, it suffices with translating it on the plane  $\psi = l^* \wedge e_2$  (see Fig. 9) using the translator  $T_1 = 1 + \gamma e_2 e_\infty$ , in the direction of  $e_2$  (the  $y$  axis) a distance  $\gamma$ . Furthermore, we build the translator  $T_2 = 1 + d_3 e_2 e_\infty$  with the same direction ( $e_2$ ), but with a separation  $d_3$  which corresponds to the size of the gripper. Once the translators have been computed, we find the lines  $l'$  and  $l''$  by translating the line  $l$  with  $l' = T_1 l T_1^{-1}$ , and  $l'' = T_2 l' T_2^{-1}$ .

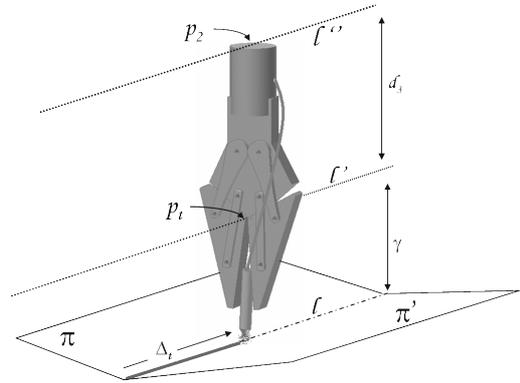


Fig. 9. Guiding lines for the robotic arm produced by the intersection of planes and vertical translation.

The next step after computing the lines, is to find the points  $p_t$  and  $p_2$  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  $l'$  and  $l''$  with a plane that defines the desired depth. In order to make the motion over the line, we build a translator  $T_L = 1 - \Delta_L l e_\infty$  with the same direction as  $l$  as shown in Fig. 8.b. Then, this translator is applied to the points  $p_2 = T_L p_2 T_L^{-1}$  and  $p_t = T_L p_t T_L^{-1}$  in an iterative fashion to yield a displacement  $\Delta_L$  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 Fig. 10.



Fig. 10. Image sequence of a linear-path motion.

## V. 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 Fig. 11).



Fig. 11. Points over the sphere as seen by the robot "Geometer".

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  $p_t$  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 Fig. 12.



Fig. 12. Guiding spheres for the arm's motion.

The dilators are produced with

$$D_\gamma = e^{-\frac{1}{2} \ln(\frac{\gamma+\rho}{\rho})E}, \text{ and} \quad (38)$$

$$D_d = e^{-\frac{1}{2} \ln(\frac{d_3+\gamma+\rho}{\rho})E}. \quad (39)$$

The spheres  $S_1$  and  $S_2$  are computed by dilating  $S_t$ :

$$S_1 = D_\gamma S_t D_\gamma^{-1}, \quad (40)$$

$$S_2 = D_d S_t D_d^{-1}. \quad (41)$$

We decompose each sphere in their parametric form as

$$p_t = M_1(\varphi)M_1(\phi)p_{s_1}M_1^{-1}(\phi)M_1^{-1}(\varphi), \quad (42)$$

$$p_2 = M_2(\varphi)M_2(\phi)p_{s_2}M_2^{-1}(\phi)M_2^{-1}(\varphi). \quad (43)$$

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  $\varphi$  and  $\phi$  and compute the corresponding  $p_t$  and  $p_2$  using Eqs. 42 and 43. The results of the simulation are shown in Fig. 13, whereas the results of the actual experiment can be seen in Figs. 14.

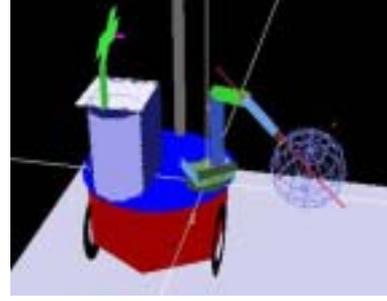


Fig. 13. Simulation of the motion over a sphere.



Fig. 14. Two of the images in the sequence of the actual experiment.

## VI. GRASPING AN OBJECT

We begin with four non-coplanar points belonging to the corners of the object and use them to build a circle. With this circle, we can make  $\pi_t$  and  $p_t$  to take the object. The procedure is described next.

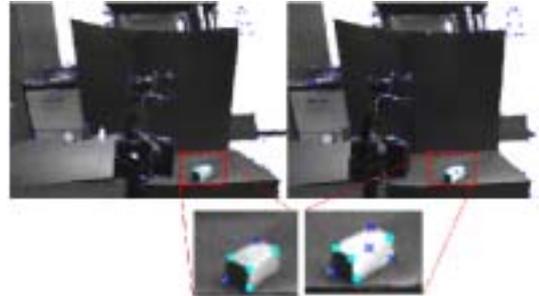


Fig. 15. Points on the object as seen from the robot.

- 1) Take a calibrated stereo pair of images of the object.
- 2) Extract four non-coplanar points from these images (see for example Fig. 15).

- 3) Compute the corresponding 3D points  $x_i, i = 1, \dots, 4$  using triangulation.
- 4) Briefly the direct distance is the vector which has the minimum Euclidean distance between two entities (see [1]). Compute the directed distances:

$$\begin{aligned} d_1 &= \text{Dist}(x_1, x_2 \wedge x_3 \wedge x_4 \wedge e_\infty), \\ d_2 &= \text{Dist}(x_2, x_1 \wedge x_3 \wedge x_4 \wedge e_\infty), \\ d_3 &= \text{Dist}(x_3, x_2 \wedge x_1 \wedge x_4 \wedge e_\infty), \\ d_4 &= \text{Dist}(x_4, x_2 \wedge x_3 \wedge x_1 \wedge e_\infty). \end{aligned}$$

- 5) Select the point with the greatest distance as the apex  $x_a$  and label the rest  $x_{b_1}, x_{b_2}, x_{b_3}$  as belonging to the base of the object.
- 6) Compute the circle  $z_b = x_{b_1} \wedge x_{b_2} \wedge x_{b_3}$ .
- 7) Compute the directed distance  $d_a$  from  $z_b$  to  $x_a$ .
- 8) Translate the circle  $z$  in the direction and magnitude of  $d_a$  to produce the grasping plane  $\pi_t$ .
- 9) Compute the inverse kinematics



Fig. 16. The robot “Geometer” grasping a wooden cube.

Some points of the previous algorithm can be explained in more detail. For example, for the object in Fig. 17.a, the base circle is  $z_b^* = x_1 \wedge x_2 \wedge x_3$ , whereas the main axis of the object is computed with  $j_{z_b} = z_b \wedge e_\infty$ . The translator that moves  $z_b$  is produced as  $T = 1 + \frac{1}{4}d_4e_\infty$ . The grasping circle can be computed with  $z_t^* = Tz_b^*T^{-1}$ . The point of contact is the closest point from the circle to the  $y$  axis. Finally, the grasping plane is  $\pi^* = z_t^* \wedge e_\infty$ . Note that this last algorithm may take the object regardless of its being in a horizontal or vertical position. We illustrate this algorithm with a simulation shown in Fig. 17.b.

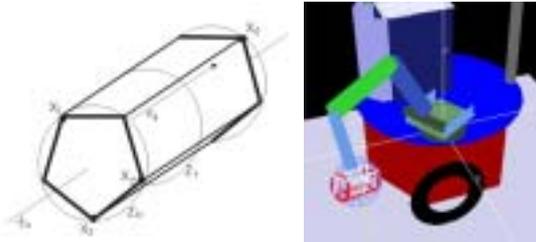


Fig. 17. a) Regular prism with height  $d_4$  and main axis  $j_{z_b}$ , b) Simulation of the algorithm showing the robotic arm grasping the object.

Conformal Geometric Algebra is very well suitable for describing physical parameters like forces. The dynamical

equation for combined rotational and translational motion takes a very compact form. For details please refer to [4].

## VII. CONCLUSION

In the previous sections we could see examples highlighting some nice properties of the Conformal Geometric Algebra. Since we are able to easily compute with geometric entities like spheres, planes and circles a lot of geometric problems in all kind of engineering can be handled in a straightforward and intuitive way.

## REFERENCES

- [1] Bayro-Corrochano E. 2001. *Geometric Computing for Perception Action Systems*, Springer Verlag, Boston.
- [2] Bayro-Corrochano E. and Banarer V. 1999. Computing depth, shape and motion using invariants and incidence algebra. Proc. Int. Conf on Pattern Recognition, ICPR'2000, Barcelona 2000
- [3] Julio Zamora and Bayro-Corrochano E., Inverse kinematics, fixation and grasping using conformal geometric algebra. CINVESTAV, Unidad Guadalajara, GEOVIS Lab., Mexico, IROS 2004, Sendai, Japan.
- [4] Hestenes D. 2001. Old Wine in New Bottles: A New Algebraic Framework for Computational Geometry, in *Geometric Algebra with applications in science and engineering* Editors Bayro-Corrochano E. and Sobczyk G. Birkhauser, Boston, 2001
- [5] Hildenbrand D., Fontijne D., Perwass Ch., Dorst L. , *Geometric Algebra and its Application to Computer Graphics*. Tutorial notes of the EUROGRAPHICS conference 2004 in Grenoble.
- [6] Lasenby, J., Lasenby, A.N., Lasenby, Doran, C.J.L and Fitzgerald, W.J. ‘New geometric methods for computer vision – an application to structure and motion estimation’. *International Journal of Computer Vision*, 26(3), 191-213. 1998.
- [7] Tolani D., Goswami A., Badler N. , Real-time inverse kinematics techniques for anthropomorphic limbs. University of Pennsylvania, 2000