OBJECT MODELLING AND COLLISION AVOIDANCE USING CLIFFORD ALGEBRA

Eduardo Bayro-Corrochano and Gerald Sommer
Computer Science Institute, Christian-Albrechts University, Kiel, Germany.
edb,gs@informatik.uni-kiel.d400.de

Keywords: Computer vision, invariance theory, clifford algebra, coordinate-free approach, geometric modelling, object modelling, object contact detection, collision avoidance.

Abstract

In this paper the authors discuss a coordinate-free geometric approach using invariants to object modelling in computer vision. The new technique used to analyse the 3-dimensional transformations involved will be that of Clifford algebra or geometric algebra. Object modelling and collision avoidance has been heavily discussed in the literature, however the Clifford algebra based method allows a more elegant reformulation which provides greater geometrical insight. In this paper the approach helps greatly to identify and to formulate algebraic constraints useful for object collision detection, grasping and collision avoidance.

1 Introduction

Geometric algebra has already been successfully applied to many areas of mathematical physics and engineering. The system adopts a coordinate-free approach and deals with rotations in n-dimensional space very easily. Therefore the authors believe that it has a good approach to offer in the world of computer vision where the 3-dimensional geometry of any given problem is of fundamental importance.

In this work the coordinate-free Clifford Algebra and invariants are used for object modelling. The essential difference between geometric algebra, the standard vector, spinor or quaternion calculus lies in the way in which vectors are multiplied together. The geometric product can be defined for any 2 multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), etc... up to grade n. Projective geometry invariants were identified using geometric algebra. Diverse geometric decision rules were developed useful for object collision detection, grasping and path planning.

The next section introduces to Clifford Algebra. The modelling of polyhedral objects are treated in section 3. Geometric equations of object contact situations are presented in section 4. This is followed by the illustrating applications collision avoidance in section 5 and grasping in section 6. The last section is dedicated to the conclusions.

2 An Outline of Clifford Algebra

Clifford algebras are well-known to pure mathematicians. In this work it is used an interpretation called geometric algebra **[1]** which is a coordinate-free approach to geometry. In geometric algebra the vectors are represented using multivectors independent of coordinate basis and in this format they are multiplied together. This is the essential difference with the standard vector calculus. This outline is essentially based on the introduction to Clifford Algebra written by Hestenes **[1]** and Bayro-Corrochano and Lasemby **[2]**.

2.1 The Geometric Product and Multivectors

The geometric or Clifford product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is written $\mathbf{ab}$ and defined as

$$\mathbf{ab} = \mathbf{a}\cdot\mathbf{b} + \mathbf{a}\wedge\mathbf{b}. \tag{1}$$

Where the outer or *wedge* product, $\wedge$, of two vectors forms a *bivector* which is interpreted as a directed area. The geometric product $\mathbf{ab}$ is therefore the sum of a scalar, $\mathbf{a}\cdot\mathbf{b}$, and a bivector, $\mathbf{a}\wedge\mathbf{b}$. In 3 dimensions the the *trivector* $(\mathbf{a}\wedge\mathbf{b})\wedge\mathbf{c}$ is an oriented 3-dimensional volume obtained by sweeping the bivector $\boldsymbol{a}\wedge\boldsymbol{b}$ along the vector $\mathbf{c}$.

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

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

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

2.2 Geometric Algebra and Rotors in 3-D Space

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

$$1, \quad \{\sigma_i\}, \quad \{\sigma_i\wedge\sigma_j\}, \quad \{\sigma_i\wedge\sigma_j\wedge\sigma_k\}, \quad \ldots, \quad \sigma_1\wedge\sigma_2\wedge\ldots\wedge\sigma_n \tag{3}$$

Note that we shall not use bold symbols for these basis vectors. The highest grade element is called the *pseudoscalar* for the space. Any multivector can be expressed in terms of this basis, and while it is often useful to do so, we stress that the main strength of geometric algebra is the ability to carry out operations in a basis-free manner. The basis for the 3-D space has $2^3 = 8$ elements given by:

$$\underbrace{1}_{scalar}, \underbrace{\{\sigma_1, \sigma_2, \sigma_3\}}_{vectors}, \underbrace{\{\sigma_1\sigma_2, \sigma_2\sigma_3, \sigma_3\sigma_1\}}_{bivectors}, \underbrace{\{\sigma_1\sigma_2\sigma_3\} \equiv i}_{trivector}. \tag{4}$$

The reference vector frame $\{\sigma_1, \sigma_2, \sigma_3\}$ corresponds to the 3-D scene space XYZ coordinate basis. The trivector or pseudoscalar $\sigma_1\sigma_2\sigma_3$ squares to -1 and commutes with all multivectors in the 3-D space. Therefore it is given the symbol i. Note that this is not the uninterpreted commutative scalar imaginary j used in quantum mechanics and engineering.

By straightforward multiplication it can be easily seen that the three bivectors can also be written as

$$\sigma_2\sigma_3 = i\sigma_1 = \boldsymbol{i}, \qquad \sigma_1\sigma_3 = -i\sigma_2 = \boldsymbol{j}, \qquad \sigma_1\sigma_2 = i\sigma_3 = \boldsymbol{k}. \tag{5}$$

These simple bivectors are spinors, as they rotate vectors in their own plane by $90°$, e.g. $(\sigma_1\sigma_2)\sigma_2 = \sigma_1$, $(\sigma_2\sigma_3)\sigma_2 = -\sigma_3$ etc. Since $(i\sigma_1)^2 = -1$, $(-i\sigma_2)^2 = -1$, $(i\sigma_3)^2 = -1$ and $(i\sigma_1)(-i\sigma_2)(i\sigma_3) = i\sigma_1\sigma_2\sigma_3 = -1$, the famous Hamilton relations

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \tag{6}$$

are easily recovered. Interpreting the $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ as bivectors, it can be seen that they indeed represent $90°$ rotations in orthogonal directions and will therefore provide a system particularly suited for the representation of 3-D rotations. Now, using nothing other than the above simple bivectors one can show that the quaternion algebra of Hamilton is simply a subset of the geometric algebra of space. If a quaternion $\mathcal{A}$ is represented by $[a_0, a_1, a_2, a_3]$, then there exists a one-to-one mapping between quaternions and rotors given by

$$\mathcal{A} = [a_0, a_1, a_2, a_3] \leftrightarrow a_0 + a_1(i\sigma_1) + a_2(i\sigma_2) + a_3(i\sigma_3) \tag{7}$$

In order to find out more about rotors in the geometric algebra we note that any rotation can be represented by a pair of reflections. It can be easily shown that the result of reflecting a vector $\mathbf{a}$ in the plane perpendicular to a unit vector $\mathbf{n}$ is

$$\mathbf{a}_{\perp} - \mathbf{a}_{\|} = -\mathbf{nan} \tag{8}$$

where $\mathbf{a}_{\perp}$ and $\mathbf{a}_{\|}$ respectively denote parts of $\mathbf{a}$ perpendicular and parallel to $\mathbf{n}$. Thus, a reflection of $\mathbf{a}$ in the plane perpendicular to $\mathbf{n}$, followed by a reflection in the plane perpendicular to $\mathbf{m}$ results in a new vector

$$-\mathbf{m}(-\mathbf{nan})\mathbf{m} = (\mathbf{mn})\mathbf{a}(\mathbf{nm}) = \mathbf{Ra}\tilde{\mathbf{R}}. \tag{9}$$

The multivector $\mathbf{R} = \mathbf{mn}$ is called a rotor. It contains only even-grade elements and satisfies $\mathbf{R}\tilde{\mathbf{R}} = 1$. The transformation $\mathbf{a} \mapsto \mathbf{Ra}\tilde{\mathbf{R}}$ is a very general way of handling rotations of multivectors of any grade unlike the quaternion calculus. In 3-D we use the term 'rotor' for those even elements of the space that represent rotations. Any rotor can be written in the form $\mathbf{R} = \pm e^{\mathbf{B}/2}$, where $\mathbf{B}$ is a bivector. In particular, in 3-D we write $\mathbf{R} = e^{(-i\frac{\theta}{2}\mathbf{n})} = \cos\frac{\theta}{2} - i\mathbf{n}\sin\frac{\theta}{2}$ which represents a rotation of θ radians anticlockwise about an axis parallel to the unit vector $\mathbf{n}$. If $\mathbf{b} = \mathbf{R_1 a \tilde{R}_1}$ and $\mathbf{c} = \mathbf{R_2 b \tilde{R}_2}$, the rotors combine in a straightforward manner, i.e. $\mathbf{c} = \mathbf{Ra\tilde{R}}$ where $\mathbf{R} = \mathbf{R_2 R_1}$.

3 Polyhedral Object Modelling

In this section the analysis is restricted to the case of polyhedral objects which can appear in any position and can also be partially occluded. This approach cannot currently deal with objects which have many curved surfaces. Canny **[3]** used quaternions for object modelling. Quaternion algebra is a subset of Clifford Algebra, hence, the Clifford algebra approach for object modelling generalizes and extends the scope of standard techniques to cope with more complicated problems.

Suppose an object undergoes a displacement from position 1 to position 2. Such a general displacement ($\mathcal{D}$) will consist of a translation ($\mathcal{T}$) expressed by the vector $\mathbf{t}$ and a rotation ($\mathcal{R}$) represented by the angle θ with respect to some axis $\mathbf{n}$ described by the rotor $\mathbf{R}$. In the analysis of this section the reference frame $\{\sigma_1, \sigma_2, \sigma_3\}$ is attached to the XYZ coordinate system at some chosen origin. The rotor $\mathbf{R}$ takes this frame to $\{\sigma_1', \sigma_2', \sigma_3'\}$ where $\sigma_i' = \mathbf{R}\sigma_i\mathbf{\tilde{R}}$ for $i = 1, 2, 3$. Let us represent the object points by position vectors relative to the origin.

A point $\mathbf{x_1}$ maps to the new point $\mathbf{x_1'}$ given by $\mathbf{x_1'} = \mathbf{Rx_1\tilde{R}} + \mathbf{t}$. An edge of the object is specified by a unit vector $\mathbf{e}$ indicating the edge direction and by a vertex lying on the edge. After a displacement the new edge is $\mathbf{e}' = \mathcal{R}(\mathcal{T}(\mathbf{e})) = \mathcal{R}(\mathbf{e}) = \mathbf{Re\tilde{R}}$, since the edge is a property within the body and is therefore unaffected by the translation. Any point on the edge can be specified by a vector $\mathbf{V_1} = \mathbf{x_1} + \lambda\mathbf{v_1}$, where λ is a variable parameter and $\mathbf{v_1}$ is a vector connecting two points, $\mathbf{x_1}$ and $\mathbf{x_2}$, on the edge such that $\mathbf{v_1} = \mathbf{x_1} - \mathbf{x_2}$. After a general displacement this goes to $\mathbf{V_1'} = \mathbf{RV_1\tilde{R}} + \mathbf{t}$. Now consider a polygon of N corners as an object, the connecting vectors $\{\mathbf{v_i}\}$ satisfy: $\mathbf{v_n} = \mathbf{v_1} + \mathbf{v_2} + .. + \mathbf{v_3} + .. + \mathbf{v_{n-1}}$. This polygon can be specified completely using these connecting vectors and one of the vertices, $\mathbf{V_n} = \mathbf{x_n} + \mathbf{v_n} = \mathbf{x_n} + (\mathbf{v_1} + \mathbf{v_2} + ... + \mathbf{v_{n-1}})$. Therefore, after a displacement the polygon is specified by $\mathbf{V_n'} = \mathbf{x_n'} + \mathbf{v_n'}$, which can be written as

$$\mathbf{V_n'} = \mathbf{Rx_n\tilde{R}} + \mathbf{t} + \mathbf{Rv_n\tilde{R}} = \mathbf{R(V_n)\tilde{R}} + \mathbf{t} = \mathbf{Rx_n\tilde{R}} + \mathbf{R(v_1)\tilde{R}} + \mathbf{R(v_2)\tilde{R}} + ... + \mathbf{R(v_{n-1})\tilde{R}} \quad (10)$$

Collinear points represented as vectors on a planar surface can be detected using the constraint equation $\mathbf{x_{n1}} \wedge \mathbf{x_{n2}} \wedge \mathbf{x_{n3}} = 0$, for any three such points. Points on a plane are in a "general position" if three of them are not collinear. The last equation can be used for selecting a set of points in some general position.

Now consider four points $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}$ in general position on an object's planar surface. The description of the planar surface can be based on terms of its outward normal vector and its directed distance. For that, first let us compute the intersections of the gravity positions of the triangles $\triangle\mathbf{x_1x_2x_4}$ and $\triangle\mathbf{x_2x_3x_4}$, namely $\mathbf{r_1} = \frac{1}{3}(\mathbf{x_1} + \mathbf{x_2} + \mathbf{x_4})$ and $\mathbf{r_2} = \frac{1}{3}(\mathbf{x_2} + \mathbf{x_3} + \mathbf{x_4})$. Their unit vectors are $\mathbf{u_{r_1}} = \frac{\mathbf{r_1}}{|\mathbf{r_1}|}$ and $\mathbf{u_{r_2}} = \frac{\mathbf{r_2}}{|\mathbf{r_2}|}$. Then the unit vectors between the points $\mathbf{x_2}$ and $\mathbf{r_1}$ and $\mathbf{x_2}$ and $\mathbf{r_2}$ are computed using $\mathbf{u_{x_2}} = \frac{\mathbf{x_2}}{|\mathbf{X_2}|}$ as follows $\mathbf{u_{y_1}} = \mathbf{u_{x_2}} - \mathbf{u_{r_1}}$ and $\mathbf{u_{y_2}} = \mathbf{u_{x_2}} - \mathbf{u_{r_2}}$. The outward normal unit vector to the planar surface will be $\mathbf{n} = \mathbf{u_{y_1}} \wedge \mathbf{u_{y_2}}$. Considering any point $\mathbf{x}$ lying on the surface, the directed distance to the planar surface is $d = \langle \mathbf{nx} \rangle$.

In this way polyhedral faces can be individually specified by an outward normal unit vector $\mathbf{n}_F$ and the distance from the origin to the face d_F for any point $\mathbf{x_F}$ lying on the face. Alternatively a face can be specified by the homogeneous normal

$$\mathbf{H_F} = d_F + \mathbf{n_F}. \tag{11}$$

Note that this multivector consists of a scalar and a vector and can simplify the equation of the plane through the face. For example, any point $\mathbf{x}$ will be on the face if

$$\mathbf{x} \cdot \mathbf{n_F} - d_F = -\langle \mathbf{H_F}(1 - \mathbf{x}) \rangle = 0. \tag{12}$$

The multivector $\mathbf{H_F}$ transforms as follows under a general displacement $\mathcal{D}$; $\mathbf{H_F}' = \mathcal{D}(\mathbf{H_F}) = (\mathbf{R x_F \tilde{R}} + \mathbf{t}) \cdot (\mathbf{R n_F \tilde{R}}) + (\mathbf{R n_F \tilde{R}})$. Since $(\mathbf{R x_F \tilde{R}}) \cdot (\mathbf{R n_F \tilde{R}}) = \mathbf{x_F} \cdot \mathbf{n_F}$, this becomes $\mathbf{H_F}' = \mathbf{x_F} \cdot \mathbf{n_F} + \mathbf{R n_F \tilde{R}} + \mathbf{t} \cdot (\mathbf{R n_F \tilde{R}})$ which can then be written as

$$\mathbf{H_F}' = \mathcal{D}(\mathbf{H_F}) = \mathbf{R H_F \tilde{R}} + \langle \mathbf{R H_F \tilde{R} t} \rangle. \tag{13}$$

We will use this characterization of the displaced face in what follows.

4 Detection of Polyhedral Contacts

Regarding particular situations, the equations of the previous section can be used for defining a set of geometric rules useful for polyhedral modelling, contact detection, collision avoidance and path planning. Let us consider a moving object 1 and a static object 2 as obstacle.

Situation 1: A displaced object touches with its face F_1 a vertex $\mathbf{x_2}$ of an obstacle. The vertex must lie on the face F_1. The equation $\langle \mathbf{H}'_{\mathbf{F_1}}(1 - \mathbf{x_2}) \rangle = 0$ has to be satisfied. After replacing $\mathbf{H}'_{\mathbf{F_1}} = \mathbf{R H_{F_1} \tilde{R}} + \langle \mathbf{R H_{F_1} \tilde{R} t} \rangle$ the equation for the situation 1 is

$$\langle \mathbf{R H_{F_1} \tilde{R}} (\mathbf{1} - \mathbf{x_2} + \mathbf{t}) \rangle = \mathbf{0}. \tag{14}$$

Situation 2: A displaced object touches with its vertex $\mathbf{x_1}$ a face F_2 of an obstacle.

This means $\langle \mathbf{H_{F_2}}(1 - \mathbf{x}'_1) \rangle = 0$. Substituting $\mathbf{x}' = \mathbf{R X_1 \tilde{R}} - \tilde{\mathbf{t}}$ the equation for situation 2 is

$$\langle \mathbf{H_{F_2}} (\mathbf{1} - \mathbf{R x_1 \tilde{R}} + \mathbf{t}) \rangle = \mathbf{0}. \tag{15}$$

Situation 3: Contact occurs when an edge of a displaced object touches an edge of an obstacle. If the edges intersect at a point, all points of both edges belong to the same plane. The edge directions and the vector joining $\mathbf{x}'_1$ (on the edge of the displaced object) and $\mathbf{x_2}$ (on the edge of the obstacle) are coplanar. If the edge vectors are coplanar they either intersect at some point or they are parallel. This condition is true if $(\mathbf{x_1'} - \mathbf{x_2}) \wedge \mathbf{e_2} \wedge \mathbf{e_1'} = 0$ – if the edges are parallel then obviously we have $\mathbf{e_1'} \wedge \mathbf{e_2} = 0$.

Since we are working in 3-dimensions, $(\mathbf{x}_1' - \mathbf{x}_2) \wedge \mathbf{e}_2 \wedge \mathbf{x}_1'$ is a trivector and can therefore be written as αi, where α is a scalar. Thus, $\alpha i = 0$ is equivalent to saying that $\langle i(i\alpha)\rangle = 0$. The co-planarity condition can then be written as $\langle i(\mathbf{x}_1' - \mathbf{x}_2) \wedge \mathbf{e}_2 \wedge \mathbf{x}_1' \rangle = 0$ Since the quantity in the angled brackets is made up of vector and trivector parts we can write

$$\langle i\mathbf{R}\mathbf{x_1}\mathbf{e_1}\tilde{\mathbf{R}}\mathbf{e_2}\rangle + \langle i(\mathbf{t} - \mathbf{x_2})\mathbf{R}\mathbf{e_1}\tilde{\mathbf{R}}\mathbf{e_2}\rangle = 0. \tag{16}$$

Situation 4: A line lies on a face when a face of an object is in contact with the edge on an obstacle. This situation can be geometrically represented as: $\mathbf{H_1'} \wedge \mathbf{e_2}$=0. Substituting the expression for a displaced homogeneous normal $\mathbf{H_F'}$ follows

$$\mathbf{H_1'} \wedge \mathbf{e_2} = \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} + \langle \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}}\mathbf{t}\rangle \wedge \mathbf{e_2} = \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} \wedge \mathbf{e_2} = \mathbf{0}. \tag{17}$$

Situation 5: An object's face approaches an obstacle's face. The orientation difference between the faces can be detected using: $\mathbf{H'_{F_1}} \wedge \mathbf{H_{F_2}} = (\mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} + \langle \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}}\mathbf{t}\rangle) \wedge \mathbf{H_{F_2}} = (\mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} \wedge \mathbf{H_{F_2}} + \langle \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}}\mathbf{t}\rangle \wedge \mathbf{H_{F_2}}$=0.

$$\mathbf{H'_{F_1}} \wedge \mathbf{H_{F_2}} = \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} \wedge \mathbf{H_{F_2}} = \mathbf{0}. \tag{18}$$

The distance between the two faces during the approach is: $\mathbf{H'_{F_1}} - \mathbf{H_{F_2}} \geq \mathbf{0}$. Replacing the expression for $\mathbf{H'_{F_1}}$ follows

$$\mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}} + \langle \mathbf{R}\mathbf{H_{F_1}}\tilde{\mathbf{R}}\mathbf{t}\rangle - \mathbf{H_{F_2}} \geq \mathbf{0}. \tag{19}$$

These equations can be used for collision avoidance and also for detection of overlapped polyhedral objects. Note that manipulations using the multivector $\mathbf{H_F}$ do not require a coordinate basis and therefore provide us with greater geometric insight and transparency. As a mode of illustration two simple examples of applications will be given in the next sections.

5 Collision Avoidance

This section presents a collision avoidance approach for a mobile robot (object 1). The general equation for avoiding collisions considering intersection of N volumes Vol_j can be simply formulated as:

$$(Vol_1' \bigcap Vol_2') \bigvee (Vol_1' \bigcap Vol_3') \bigvee ... \bigvee (Vol_1' \bigcap Vol_N') = 0. \tag{20}$$

The movement is a displacement, hence the equation of the moving robot and any threaten object is $\mathtt{C}_2 \bigwedge \mathtt{C}_3 \bigwedge ... \bigwedge \mathtt{C}_\mathtt{N} = \mathtt{0}$ with

$$C_j : \mathcal{D}(Vol_1) \bigcap \mathcal{D}(Vol_j) \tag{21}$$

for j=2,3,...,N. Note that especially for non-polyhedral objects this equation only can be written as a complicated and non-transparent expression in terms of the standard vector, spinor or quaternion calculus (see Canny [**3**]). Using the geometric algebra approach the equation will be expressed in a clear and easier way as follows. Firstly, consider a couple of vectors **a**, **b** and a point $\mathbf{x_1}$ lying in the back surface of the moving robot. Any point $\mathbf{x_j}$ of an external object j lying on a virtual internal surface of the object 1 can be detected using an extension of the equation of situation 3 a trivector based condition for co-planarity

$$\langle(\mathbf{a}-\mathbf{p_j'})((\mathbf{a}+\mathbf{b})'-\mathbf{p_j'})(\mathbf{x_1'}-\mathbf{p_j'}-\mathbf{x_j'})\rangle=\mathbf{0} \tag{22}$$

and a approach distance limit condition

$$||((\mathbf{x_{center}a}-\mathbf{p_j'})-\mathbf{x_j'})||<\mathbf{d_{max}} \tag{23}$$

where the distance to the touching point $\mathbf{x_j}$ is $\mathbf{p_j'}\approx\langle(\mathbf{R_j x_j \tilde{R}_j}-\tilde{\mathbf{t_j}})-(\mathbf{R_1 x_1 \tilde{R}_1}-\tilde{\mathbf{t_1}})\cdot\mathbf{R_1 n_1 \tilde{R}_1}\rangle$ (here $\mathbf{n}_1$ corresponds to the unit vector of the orientation of the movement of object 1) and $\mathbf{a}'=\mathbf{R_1 a \tilde{R}_1}-\tilde{\mathbf{t_1}}$, $\mathbf{x_1'}=\mathbf{R_1 x_1 \tilde{R}_1}-\tilde{\mathbf{t_1}}$, $(\mathbf{a}+\mathbf{b})'=\mathbf{R_1}(\mathbf{a}+\mathbf{b})\mathbf{\tilde{R}_1}-\tilde{\mathbf{t_1}}$, $\mathbf{x_j'}=\mathbf{R_j x_j \tilde{R}_j}-\tilde{\mathbf{t_j}}$ for j=2,3,...,N (threatening objects).
The fulfillment of this equation requires a control of the robot movement. This can be formulated in terms of a trivector. Considering the control volume V_c of the trajectory of the robot within time t follows

$$V_c=\int_{\mathbf{x_1}(t=0)}^{\mathbf{x_1}(T)}\mathbf{a}\wedge\mathbf{b}\wedge d\mathbf{x}=\int_0^T\mathbf{a}\wedge\mathbf{b}\wedge\frac{d\mathbf{x}}{dt}dt=\int_0^T\mathbf{a}\wedge\mathbf{b}\wedge\mathbf{x^{(1)}}(t)\,dt \tag{24}$$

Observe that the rate how the volume is swept is $V_c^{(1)}(t)=\mathbf{a}\wedge\mathbf{b}\wedge\mathbf{x^{(1)}}(t)$, where $\mathbf{x^{(1)}}(t)$ can be computed using an interpolation polynom utilizing the points $\mathbf{x_1},\mathbf{x_2},...,\mathbf{x_m}$ of the security distance for avoiding a collision with the obstacles. Now the trajectory volume has to allow smooth movements of the robot. That means the volume should have almost a constant cross shape everywhere. This requires that the sweeping area should also be smoothly rotated. Then the control equation is

$$V_c(t)=\int_0^T\mathbf{R(t)a}\wedge\mathbf{b\tilde{R}(t)})\wedge\mathbf{x^{(1)}(t)}\,dt \tag{25}$$

The robot has to move perpendicular to the obstacle's face. Using the normals to the obstacle surfaces the required orientation angles can be derived and using a interpolation polynom the rotor $\mathbf{R}(t)=e^{(-\mathbf{i}\frac{\theta(t)}{2}\mathbf{n})}$ can be computed. Note that this is an adaptive control of the trajectory volume. Whenever a threatening obstacle moves the interpolating polynoms will adapt the equation.

6 Grasping

A simple example of a grasping application will now be given. The symbols used are: $\wedge$ for the geometric outer product, $\bigwedge$ for the Boolean AND operation and $\bigvee$ for the Boolean OR operation. The example considers the positioning of a two finger grasper in front of a static object. Consider two points $\mathbf{g_1}$ and $\mathbf{g_2}$ which are the closest corners of the finger tips and two points $\mathbf{x_1}$, $\mathbf{x_2}$ lying on the extremes of the object. These points lie on the adequate grasping surface defined during the previous object recognition process. A geometric rule for good grasping is that three simple constraints have to be simultaneously fulfilled. This can be written as $\mathtt{C}_1 \bigvee \mathtt{C}_2 \bigvee \mathtt{C}_3 \approx 0$, where the conditions $\mathtt{C}_1$ for aperture, $\mathtt{C}_2$ for attitude and $\mathtt{C}_3$ for alignment, are given by

$$
\begin{aligned}
&\mathtt{C}_1 : \mathbf{R}(\mathbf{g_1} - \mathbf{g_2})\tilde{\mathbf{R}} - (\mathbf{x_1} - \mathbf{x_2}) \approx 0 \\
&\mathtt{C}_2 : \langle i\mathbf{e}_{12}\mathbf{e}_{21}\mathbf{e}_{14}\rangle \bigvee \langle i\mathbf{e}_{12}\mathbf{e}_{21}\mathbf{e}_{23}\rangle \approx 0 \\
&\mathtt{C}_3 : \left(\frac{\mathbf{R}(\mathbf{g_1} + \mathbf{g_2})\tilde{\mathbf{R}} + 2\mathbf{t} - (\mathbf{x_1} + \mathbf{x_2})}{2}\right) \cdot \left(\mathbf{R}(\mathbf{g_1} - \mathbf{g_2})\tilde{\mathbf{R}}\right) \approx 0
\end{aligned}
\tag{26}
$$

Here, $\mathbf{x}_3$ and $\mathbf{x}_4$ are points on the far side of the object, such that $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ are in a plane which is parallel to the floor. $\mathbf{e}_{12} = \frac{\mathbf{x}_1 - (\mathbf{R}\mathbf{g}_2\tilde{\mathbf{R}} + \mathbf{t})}{|\mathbf{x}_1 - (\mathbf{R}\mathbf{g}_2\tilde{\mathbf{R}} + \mathbf{t})|}$ is the unit edge vector between the points $\mathbf{x_1}$ and $\mathbf{g_2'}$, $\mathbf{e}_{21} = \frac{\mathbf{x}_2 - (\mathbf{R}\mathbf{g}_1\tilde{\mathbf{R}} + \mathbf{t})}{|\mathbf{x}_2 - (\mathbf{R}\mathbf{g}_1\tilde{\mathbf{R}} + \mathbf{t})|}$ is the unit edge vector between the points $\mathbf{x_2}$ and $\mathbf{g_1'}$ and $\mathbf{e}_{14}$ and $\mathbf{e}_{23}$ are respectively the unit edge vectors $\frac{\mathbf{x}_1 - \mathbf{x}_4}{|\mathbf{x}_1 - \mathbf{x}_4|}$ and $\frac{\mathbf{x}_2 - \mathbf{x}_3}{|\mathbf{x}_2 - \mathbf{x}_3|}$.

7 Conclusion

The authors have shown that the use of geometric algebra helps greatly to the identification and the formulation of algebraic constraints useful for object modelling, collision avoidance and grasping. Instead of laborious matrix operations, the geometric algebra based method offers stability and avoids redundant elements present in matrix calculus. Finally, the author believe that the reformulation of the problem in purely geometric terms give greater intuitive insight and will enable more complicated problems to be successfully addressed.

References

1. Hestenes D and Sobczyk G. Clifford Algebra to Geometric Calculus. A Unified Language for Mathematics and Physics. D Reidel Publishing Company, Dordrecht, Holland, 1984.

2. Bayro-Corrochano EJ and Lasenby J. Object modelling and motion analysis using Clifford Algebra. Europe-China workshop on Geometrical Modelling and Invariants for Computer Vision, Xi'an, China, pp. 143-149, 27-29 April 1995.

3. Canny J. Collision detection for moving polyhedra. PAMI-8, No. 2, pp. 200-209, March 1986.

This article was processed using the LaTeX macro package with LLNCS style