OBJECT MODELLING AND MOTION ANALYSIS USING CLIFFORD ALGEBRA

Eduardo Bayro-Corrochano
Computer Science Institute, Christian-Albrechts University, Kiel, Germany.
edb@informatik.uni-kiel.d400.de
Joan Lasenby
Department of Engineering, Trumpington Street, Cambridge, UK.
jl@eng.cam.ac.uk

Keywords: Computer vision, invariance theory, clifford algebra, coordinate-free approach, geometric modelling, collision detection, motion analysis.

Abstract

This paper discusses a coordinate-free geometric approach to object modelling and the analysis of motion in computer vision. The new technique used to analyse the 3-dimensional transformations involved will be that of Clifford algebra or geometric algebra. This is not an approach designed specifically for the task in hand, but rather a framework for almost all mathematical physics. Both object modelling and estimating camera motion from different scene projections have been heavily discussed in the literature, however the Clifford algebra based method allows a more elegant reformulation which provides greater geometrical insight. The hope is that one can then extend the scope of the techniques to more complicated problems.

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 efficiently. The authors believe that it could be a useful tool for object modelling and motion analysis where the 3-dimensional geometry of any given problem is of fundamental importance.

The next section gives a brief introduction to Clifford Algebra. The is followed by sections which present geometric equations for polyhedral object modelling and discuss the analysis of polyhedral objects in contact situations with an application in robotics. Camera motion from two scene projections and the conclusions are presented in the final sections.

2 An Outline of Clifford Algebra

Clifford algebras are well-known to pure mathematicians. In this work we will use an interpretation called geometric algebra [1] which is a coordinate-free approach to geometry. In geometric algebra the elements are coordinate-independent objects called multivectors which can be multiplied together using a *geometric product*. It is thus very different to standard vector calculus.

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

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

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

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

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 where $\mathbf{a}_-$ 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

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

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 $\mathbf{a} \mapsto \mathbf{R}\mathbf{a}\tilde{\mathbf{R}}$ 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 $\mathbf{b} = \mathbf{R_1}\mathbf{a}\tilde{\mathbf{R}}_\mathbf{1}$ and $\mathbf{c} = \mathbf{R_2}\mathbf{b}\tilde{\mathbf{R}}_\mathbf{2}$, the rotors combine in a straightforward manner, i.e. $\mathbf{c} = \mathbf{R}\mathbf{a}\tilde{\mathbf{R}}$ where $\mathbf{R} = \mathbf{R_2}\mathbf{R_1}$.

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 [2] 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 $\sigma_i' = \mathbf{R}\sigma_i\tilde{\mathbf{R}}$ 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 $\mathbf{x_1'} = \mathbf{R}\mathbf{x_1}\tilde{\mathbf{R}} + \mathbf{t}$. 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 $\mathbf{e}'=\mathcal{R}(\mathcal{T}(\mathbf{e})) = \mathcal{R}(\mathbf{e}) = \mathbf{R}\mathbf{e}\tilde{\mathbf{R}}$, 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 $\mathbf{V_1'} = \mathbf{R}\mathbf{V_1}\tilde{\mathbf{R}} + \mathbf{t}$. 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

$$\begin{aligned} \mathbf{V_n'} &= \mathbf{R}\mathbf{x_n}\tilde{\mathbf{R}} + \mathbf{t} + \mathbf{R}\mathbf{v_n}\tilde{\mathbf{R}} = \mathbf{R}(\mathbf{V_n})\tilde{\mathbf{R}} + \mathbf{t} \\ &= \mathbf{R}\mathbf{x_n}\tilde{\mathbf{R}} + \mathbf{R}(\mathbf{v_1})\tilde{\mathbf{R}} + \mathbf{R}(\mathbf{v_2})\tilde{\mathbf{R}} + ... + \mathbf{R}(\mathbf{v_{n-1}})\tilde{\mathbf{R}} + \mathbf{t} \end{aligned} \tag{10}$$

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.

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

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

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

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

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

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

4 Detection of Polyhedral Contacts

In 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 the obstacle.

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

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

Situation 2: A displaced object touches a face F_2 of an obstacle with its vertex $\mathbf{x_1}$. This means $\langle \mathbf{H_{F_2}}(1-\mathbf{x}'_1)\rangle = 0$. Substituting $\mathbf{x}'_1 = \mathbf{R}\mathbf{x_1}\tilde{\mathbf{R}} + \mathbf{t}$ the equation for situation 2 becomes

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

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 coplanarity 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

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

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.

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 $\mathtt{C}_1 \bigwedge \mathtt{C}_2 \bigwedge \mathtt{C}_3 = 0$, where the conditions $\mathtt{C}_1$ for

$$
\begin{aligned}
\mathtt{C}_1 &: \mathbf{R}(\mathbf{g_1}-\mathbf{g_2})\tilde{\mathbf{R}} \geq \mathbf{x_1}-\mathbf{x_2} \\
\mathtt{C}_2 &: \langle i\mathbf{e}_{12}\mathbf{e}_{21}\mathbf{e}_{14}\rangle \bigvee \langle i\mathbf{e}_{12}\mathbf{e}_{21}\mathbf{e}_{23}\rangle = 0 \\
\mathtt{C}_3 &: \left(\frac{\mathbf{R}(\mathbf{g_1}+\mathbf{g_2})\tilde{\mathbf{R}}+2\mathbf{t}-(\mathbf{x_1}+\mathbf{x_2})}{2}\right)\cdot\left(\mathbf{R}(\mathbf{g_1}-\mathbf{g_2})\tilde{\mathbf{R}}\right) = 0
\end{aligned} \tag{17}
$$

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 $\frac{\mathbf{x}_2-\mathbf{x}_3}{|\mathbf{x}_2-\mathbf{x}_3|}$.

5 Camera motion from two scene projections

The 'eight-point algorithm' of Longuet-Higgins [3] computes the 3D structure of a scene from a correlated pair of perspective projections when the spatial relationship between the projections is unknown. This paper presents a method of obtaining the eight simultaneous equations in [3] via a totally geometric approach using only vectors and rotations. Suppose the camera undergoes a displacement taking it from position 1 to position 2. Such a general displacement will consist of a translation plus a rotation. At each position the camera observes some point P in the scene. The image planes of the camera are α_1 and α_2. Let O_1 be the optical centre of the camera at position 1 and consider a frame with origin O_1 and axes $(\sigma_1, \sigma_2, \sigma_3)$, where σ_3 is perpendicular to the plane α_1. Let $\mathbf{X}_1 = \overrightarrow{O_1P}$ be the position vector of P relative to O_1, and $\mathbf{x}_1 = \overrightarrow{O_1M_1}$ be the position vector of M_1 relative to O_1, where M_1 is the projection of the point P onto the plane α_1. Let the translation be given by the vector $\mathbf{t}$ such that $\overrightarrow{O_1O_2} = \mathbf{t}$, where O_2 is the translation of O_1. A general rotation of θ about some axis $\mathbf{n}$ will be described by the rotor $\mathbf{R}$, where $\mathbf{R} = \exp(-i\frac{\theta}{2}\mathbf{n})$. Thus, the frame $(\sigma_1, \sigma_2, \sigma_3)$ is rotated to a frame $(\sigma_1', \sigma_2', \sigma_3')$ at O_2 where $\sigma_i' = \mathbf{R}\sigma_i\tilde{\mathbf{R}}$ for $i = 1, 2, 3$. At position 2 we have a new image plane α_2, and we let M_2 be the projection of P onto α_2. Relative to the frame at O_1 we define $\tilde{\mathbf{X}} = \overrightarrow{O_2P}$ and $\tilde{\mathbf{x}} = \overrightarrow{O_2M_2}$. If the position vectors of P and M_2 relative to O_2 are given by $\mathbf{X}_2$ and $\mathbf{x}_2$, then it is clear that the relation, $\mathbf{X}_2 = \mathbf{R}(\mathbf{X}_1 - \mathbf{t})\tilde{\mathbf{R}}$, holds.
In the geometric algebra approach the condition for coplanarity of $\tilde{\mathbf{X}}, \mathbf{X}_1$ and $\mathbf{t}$ is just $\tilde{\mathbf{X}}\wedge\mathbf{X}_1\wedge\mathbf{t} = 0$. As in the previous section, we can now write

$$
\langle i\tilde{\mathbf{X}}\wedge\mathbf{X}_1\wedge\mathbf{t}\rangle = \langle i\tilde{\mathbf{X}}\mathbf{X}_1\mathbf{t}\rangle = \langle i\tilde{\mathbf{R}}\mathbf{X}_2\mathbf{R}\mathbf{X}_1\mathbf{t}\rangle = 0. \tag{18}
$$

Using the definition of the geometric product and the fact that $\langle ABC\rangle = \langle BCA\rangle = \langle CAB\rangle$, the coplanarity condition can now be given by

$$
\langle \mathbf{X}_2\mathbf{R}(i\mathbf{X}_1\wedge\mathbf{t})\tilde{\mathbf{R}})\rangle = 0 \tag{19}
$$

With $\mathbf{X}_1 = X_1^1\sigma_1 + X_1^2\sigma_2 + X_1^3\sigma_3$ it is then possible to write $i\mathbf{X}_1\wedge\mathbf{t} = X_1^j(i\sigma_j\wedge\mathbf{t}) \equiv X_1^j\boldsymbol{m}_j$ where the vector $\boldsymbol{m}_j$ is defined by $\boldsymbol{m}_j = i\sigma_j\wedge\mathbf{t}$. $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$ therefore lie in the plane perpendicular to $\mathbf{t}$. If we apply the rotor $\mathbf{R}$ to the vectors $\boldsymbol{m}_j$ we obtain a set of rotated vectors which we shall call $\tilde{\boldsymbol{m}}_j = \mathbf{R}\boldsymbol{m}_j\tilde{\mathbf{R}}$. The coplanarity condition is now reduced to $\langle \mathbf{X}_2X_1^j\tilde{\boldsymbol{m}}_j\rangle = 0$. Evaluating the scalar part of $\mathbf{X}_2X_1^j\tilde{\boldsymbol{m}}_j$ gives

$$
X_1^1(\mathbf{X}_2\cdot\tilde{\boldsymbol{m}}_1) + X_1^2(\mathbf{X}_2\cdot\tilde{\boldsymbol{m}}_2) + X_1^3(\mathbf{X}_2\cdot\tilde{\boldsymbol{m}}_3) = 0 \tag{20}
$$

(X_1^1, X_1^2, X_1^3) and (X_2^1, X_2^2, X_2^3) are the true 3-dimensional cartesian coordinates of the point P relative to the two viewpoints 1 and 2. Projected onto the image planes α_1 and α_2, the image coordinates of

Following the notation of Longuet-Higgins we write $\mathbf{x}_1 = \frac{\mathbf{X}_1}{X_1^3}$ and $\mathbf{x}_2 = \frac{\mathbf{X}_2}{X_2^3}$. Dividing equation (20) by $X_1^3 X_2^3$ then gives us

$$x_1^j \mathbf{x}_2 \cdot \tilde{\boldsymbol{m}}_j = 0 \tag{21}$$

The vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are known from measurements in the two image planes and equation (21) therefore provides us with a linear equation in the 9 unknowns $\{\tilde{m}_j^i\}$.

If $\tilde{\boldsymbol{M}} = (\tilde{m}_1^1, \tilde{m}_1^2, \tilde{m}_1^3, \tilde{m}_2^1, \tilde{m}_2^2, \tilde{m}_2^3, \tilde{m}_3^1, \tilde{m}_3^2, \tilde{m}_3^3)$ is a 9-element vector we can write equation (21) as $\boldsymbol{A}_i \cdot \tilde{\boldsymbol{M}}^{\mathrm{T}} = 0$. For each point P_i in the scene for which we have a match, there is a corresponding equation $\boldsymbol{A}_i \cdot \tilde{\boldsymbol{M}}^{\mathrm{T}} = 0$. As described by Longuet-Higgins, the *ratios* of the 9 unknowns $\{\tilde{m}_j^i\}$ can be obtained by solving 8 simultaneous equations. We assume that such a solution has been found and that the vectors $\tilde{\boldsymbol{m}}_i$ are known. For inner products between the $\tilde{\boldsymbol{m}}_i$'s we note that

$$\tilde{\boldsymbol{m}}_i \cdot \tilde{\boldsymbol{m}}_j = \langle \tilde{\boldsymbol{m}}_i \tilde{\boldsymbol{m}}_j \rangle = \langle \mathbf{R}\boldsymbol{m}_i \tilde{\mathbf{R}} \mathbf{R} \boldsymbol{m}_j \tilde{\mathbf{R}} \rangle = \langle \mathbf{R} \boldsymbol{m}_i \boldsymbol{m}_j \tilde{\mathbf{R}} \rangle = \langle \boldsymbol{m}_i \boldsymbol{m}_j \rangle = \boldsymbol{m}_i \cdot \boldsymbol{m}_j . \tag{22}$$

The above is simply a statement of the fact that the inner product of two vectors remains invariant under a rotation. It can easily be shown that

$$\tilde{\boldsymbol{m}}_i \cdot \tilde{\boldsymbol{m}}_j = (i\sigma_i \wedge \mathbf{t}) \cdot (i\sigma_k \wedge \mathbf{t}) = -\tfrac{1}{2} \langle \sigma_i \mathbf{t} \sigma_j \mathbf{t} \rangle + \tfrac{1}{2} \delta_{ij} \tag{23}$$

¿From this it follows that $\tilde{\boldsymbol{m}}_i \cdot \tilde{\boldsymbol{m}}_i = 1 - t_i^2$ for $i = 1, 2, 3$. It is therefore possible to reconstruct $\mathbf{t}$ easily from a knowledge of the $\tilde{\boldsymbol{m}}_i$'s.

$$t_i = \pm \sqrt{1 - \tilde{\boldsymbol{m}}_i \cdot \tilde{\boldsymbol{m}}_i} \tag{24}$$

for $i = 1, 2, 3$, no summation implied. The rotor $\mathbf{R}$ is such that it rotates $\boldsymbol{m}_i$ into $\tilde{\boldsymbol{m}}_i$ for $i = 1, 2, 3$, so that $\mathbf{R}$ can be recovered using the following standard procedure. Given any two distinct vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ (because the $\{\boldsymbol{m}_i\}$ are coplanar we need deal only with any two) which are rotated under $\mathbf{R}$ to give $\tilde{\boldsymbol{m}}_1$ and $\tilde{\boldsymbol{m}}_2$, the rotor $\mathbf{R}$ which gives this is $\mathbf{R} = \exp(-i\frac{\theta}{2}\mathbf{n})$ where θ and $\mathbf{n}$ are given by

$$\mathbf{n} = \frac{i(\tilde{\boldsymbol{m}}_1 - \boldsymbol{m}_1) \wedge (\tilde{\boldsymbol{m}}_2 - \boldsymbol{m}_2)}{|i(\tilde{\boldsymbol{m}}_1 - \boldsymbol{m}_1) \wedge (\tilde{\boldsymbol{m}}_2 - \boldsymbol{m}_2)|} \tag{25}$$

$$\theta = \cos^{-1} \left\{ \frac{(\boldsymbol{m}_1 - (\boldsymbol{m}_1 \cdot \mathrm{n})\mathbf{n}) \cdot (\tilde{\boldsymbol{m}}_1 - (\tilde{\boldsymbol{m}}_1 \cdot \mathrm{n})\mathbf{n})}{|(\boldsymbol{m}_1 - (\boldsymbol{m}_1 \cdot \mathrm{n})\mathbf{n}) \cdot (\tilde{\boldsymbol{m}}_1 - (\tilde{\boldsymbol{m}}_1 \cdot \mathrm{n})\mathbf{n})|} \right\} \tag{26}$$

To reconstruct the scene we find the true position vector $\mathbf{X}_1$ of some point P. If we write $\mathbf{X}_1 = \mathbf{x}_1 X_1^3$ and $\mathbf{X}_2 = \mathbf{x}_2 X_2^3$ and substitute in $\mathbf{X}_2 = \mathbf{R}(\mathbf{X}_1 - \mathbf{t})\tilde{\mathbf{R}}$ we have

$$\begin{aligned} X_2^3 \mathbf{x}_2 &= X_1^3 \mathbf{R} \mathbf{x}_1 \tilde{\mathbf{R}} - \mathbf{R} \mathbf{t} \tilde{\mathbf{R}} \\ &= X_1^3 \tilde{\mathbf{x}}_2 - \tilde{\mathbf{t}} \end{aligned} \tag{27}$$

Now consider some vector orthogonal to $\mathbf{x}_2$, say $\mathbf{p} = i\sigma_1 \wedge \mathbf{x}_2$ and take the inner product of equation (27) with $\mathbf{p}$:

$$X_2^3 \mathbf{x}_2 \cdot \mathbf{p} = X_1^3 \tilde{\mathbf{x}}_2 \cdot \mathbf{p} - \tilde{\mathbf{t}} \cdot \mathbf{p} \tag{28}$$

As $\mathbf{p}$ is orthogonal to $\mathbf{x}_2$, this gives an explicit expression for X_1^3

$$X_1^3 = \frac{\tilde{\mathbf{t}} \cdot \mathbf{p}}{\tilde{\mathbf{x}}_2 \cdot \mathbf{p}} \tag{29}$$

Having obtained X_1^3 in this way the other coordinates X_1^1 and X_1^2 are simply found using

$$X_1^1 = x_1^1 X_1^3 \qquad \text{and} \qquad X_1^2 = x_1^2 X_1^3 \tag{30}$$

As was pointed out by Longuet-Higgins, the ambiguity in the choice of signs for t_1 is resolved by demanding that the forward coordinates of any point (i.e. X_1^3 and X_2^3) must be positive.

The authors hope to have shown that the use of geometric algebra simplifies object modelling and the geometric interpretation of object displacement. It deals with real vectors and all quantities involved in the methods have a definite geometric interpretation. The approach provides a simple formulation of algebraic constraints useful for either object detection or path planning. Instead of laborious matrix operations, the geometric algebra based method offers stability and avoids the redundant elements present in matrix calculus.

For the motion analysis it is useful to compare the geometrical quantities occurring in the geometric algebra based method with the original version of Longuet-Higgins. In [3] a rotation matrix R is used for the unknown camera rotation; this is equivalent to our rotor $\mathbf{R}$ where the row vectors of R, which in [3] are called $\boldsymbol{R}_\alpha$ are given by $\boldsymbol{R}_\alpha = \tilde{\mathbf{R}}\sigma_\alpha\mathbf{R}$. Similarly, the column vectors of R, $\boldsymbol{R}_\beta$ are given by $\boldsymbol{R}_\beta = \mathbf{R}\sigma_\beta\tilde{\mathbf{R}}$ and therefore represent the way in which the basis vectors are rotated. The elements of the rotation matrix R_{ij} can therefore be written as $\mathrm{R}_{ij} = \langle\tilde{\mathbf{R}}\sigma_i\mathbf{R}\sigma_j\rangle$. The skew-symmetric matrix S is given no geometrical interpretation in [3], but from the previous sections it is clear that the column vectors of S are the three coplanar vectors $-\boldsymbol{m}_i = -i\sigma_i\wedge\mathbf{t}$, where $\boldsymbol{m}_i$ is a vector perpendicular to σ_i and $\mathbf{t}$. Their matrix $Q = \mathrm{R}S$ has elements Q_{ij} which are given, in our notation by $Q_{ij} = \langle i\mathbf{R}\sigma_i\tilde{\mathbf{R}}\sigma_j\mathbf{t}\rangle$. Once $\mathbf{t}$ and $\tilde{\boldsymbol{m}}_i$ are known (and therefore also $\boldsymbol{m}_i$) one is able to unwrap the rotation. It is in the unwrapping of $\mathbf{R}$ that our method differs from that in [3], and it is shown in the previous section that the geometric algebra method gives a more efficient and transparent unwrapping.

In terms of pure computational efficiency the geometric algebra based method is similar to the standard algorithm of Longuet-Higgins. However, the presented method allows us to unwrap the translation, rotation and scene coordinates using purely geometric techniques, in a more efficient way. A more general geometric algebra least-squares approach to the problem of estimating camera motion in the absence of range data will be presented elsewhere.

References

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

[2] J. Canny. Collision detection for moving polyhedra. *PAMI-8.* 2, 200:209, 1986.

[3] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 133:138, Sept. 1981.