ANALYSIS OF POINT CLOUDS
Using Conformal Geometric Algebra

Dietmar Hildenbrand
Research Center of Excellence for Computer Graphics, University of Technology, Darmstadt, Germany

Eckhard Hitzer
Department of Applied Physics, University of Fukui, Japan

Keywords: Geometric algebra, geometric computing, point clouds, osculating circle, fitting of spheres, bounding spheres.

Abstract: This paper presents some basics for the analysis of point clouds using the geometrically intuitive mathematical framework of conformal geometric algebra. In this framework it is easy to compute with osculating circles for the description of local curvature. Also methods for the fitting of spheres as well as bounding spheres are presented. In a nutshell, this paper provides a starting point for shape analysis based on this new, geometrically intuitive and promising technology.

1 INTRODUCTION

The main contribution of this paper is its collection of properties and basic algorithms that make it a very promising tool for the analysis of point clouds. Please refer for instance to (Schnabel et al., 2007) for current research results concerning this application.

Conformal geometric algebra has shown some advantages in recent years. It is very easy to calculate directly with geometric objects like spheres, circles and planes. Also transformations can be handled very easily. This mathematical framework is able to unify a lot of mathematical systems like vector algebra, projective geometry, quaternions or Plücker coordinates.

While geometric algebra formerly had the problem of efficiency, now there are approaches available so that algorithms written in this framework can even be faster than conventional algorithms (see (Hildenbrand et al., 2006)).

Like implementations of quaternions can be more robust than rotation matrices geometric algebra promises to deliver more robust algorithms. As an example, planes can be represented as specific spheres allowing to fit spheres or planes into point sets (see (Hildenbrand, 2005)).

2 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 blades is called a **k-vector**. So a bivector is a linear combination of blades with grade 2. Other k-vectors are vectors (grade 1), trivectors (grade 3) and quadvectors (grade 4). Furthermore, a linear combination of blades of different grades is called a multivector. Multivectors are the general elements of a Geometric Algebra.

Table 1 presents the basic geometric entities of conformal geometric algebra, points, spheres, planes, circles, lines and point pairs. The s_i represent different spheres and the π_i different planes. The two representations are dual to each other. In order to switch between the two representations, the dual operator which is indicated by '*' (division by the pseudoscalar), can be used. For example in the stan-

Hildenbrand D. and Hitzer E. (2008).
ANALYSIS OF POINT CLOUDS - Using Conformal Geometric Algebra.
In *Proceedings of the Third International Conference on Computer Graphics Theory and Applications*, pages 99-106
DOI: 10.5220/0001094100990106
Copyright © SciTePress

Table 1: Representations of the conformal geometric entities.

entity	standard repr.	direct repr.
Point	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$	
Sphere	$s = P - \frac{1}{2}r^2 e_\infty$	$s^* = x_1 \wedge x_2 \wedge x_3 \wedge x_4$
Plane	$\pi = \mathbf{n} + d e_\infty$	$\pi^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty$
Circle	$z = s_1 \wedge s_2$	$z^* = x_1 \wedge x_2 \wedge x_3$
Line	$l = \pi_1 \wedge \pi_1$	$l^* = x_1 \wedge x_2 \wedge e_\infty$
P-Pair	$P_p = s_1 \wedge s_2 \wedge s_3$	$P_p^* = x_1 \wedge x_2$

Table 2: The geometric meaning of the inner product of conformal vectors U (1st column) and V (2nd column, rows 1,5, and 9).

$U \cdot V$	**plane**
plane	angle between planes
sphere	Euclidean distance to center
point	Euclidean distance
$U \cdot V$	**sphere**
plane	Euclidean distance to center
sphere	distance measure
point	distance measure
$U \cdot V$	**point**
plane	Euclidean distance
sphere	distance measure
point	Euclidean distance

dard representation a sphere is represented with the help of its center point P and its radius r, while in the direct representation it is constructed by the outer product '$\wedge$' of four points x_i that lie on the surface of the sphere ($x_1 \wedge x_2 \wedge x_3 \wedge x_4$). In standard representation the dual meaning of the outer product is the intersection of geometric entities. For example a circle is defined by the intersection of two spheres ($s_1 \wedge s_2$). Please notice that in this paper we indicate 3D vectors by bold letters, e.g. **n** means the 3D normal vector of a plane.

For the foundations of conformal geometric algebra and its application to computer graphics refer for instance to (L.Dorst et al., 2007), (Rosenhahn, 2003), (Fontijne and Dorst, 2003) and to the tutorials (Dorst and Mann, 2002), (Mann and Dorst, 2002), (Hildenbrand et al., 2004) and (Hildenbrand, 2005).

3 DISTANCES AND ANGLES

In conformal geometric algebra distances and angles are expressible easily with the help of the inner product. Table 2 summarizes the geometric meaning of the inner product of conformal vectors U and V.

The inner product of vectors in Conformal Geometric Algebra results in a scalar and can be used as a measure for distances between basic objects. The inner product $P \cdot S$ of two vectors P and S can be used for tasks like

- the Euclidean distance between two points
- the distance between one point and one plane
- the decision whether a point is inside or outside of a sphere.

A vector in Conformal Geometric Algebra can be written as

$$V = v_1 e_1 + v_2 e_2 + v_3 e_3 + v_4 e_\infty + v_5 e_0 \quad (1)$$

The meaning of the two additional coordinates e_0 and e_∞ is as follows :

	$\mathbf{v_5 = 0}$	$\mathbf{v_5 \neq 0}$
$\mathbf{v_4 = 0}$	plane through origin	sphere/point through origin
$\mathbf{v_4 \neq 0}$	plane	sphere/point

The multiplication with a constant $k \neq 0$ leads always to the same geometric object (like in projective space).
Division by $v_5 \neq 0$ leads to (normalized form)

$$S = s_1 e_1 + s_2 e_2 + s_3 e_3 + s_4 e_\infty + e_0 \quad (2)$$

representing a sphere S with center point $\mathbf{s}$ and radius r as

$$S = \mathbf{s} + s_4 e_\infty + e_0 \quad (3)$$

with

$$s_4 = \frac{1}{2}(\mathbf{s}^2 - r^2) = \frac{1}{2}(s_1^2 + s_2^2 + s_3^2 - r^2)$$

Points are degenerate spheres with radius $r = 0$.

$$P = \mathbf{s} + \frac{1}{2}\mathbf{s}^2 e_\infty + e_0 \quad (4)$$

Planes are degenerate spheres with infinite radius. They are represented as a vector with $v_5 = 0$

$$V = v_1 e_1 + v_2 e_2 + v_3 e_3 + v_4 e_\infty = \mathbf{v} + v_4 e_\infty \quad (5)$$

The inner product of 3D vectors corresponds to the well-known scalar product. The 3D basis vectors e_1, e_2, e_3 square to 1

$$e_1^2 = e_2^2 = e_3^2 = 1 \quad (6)$$

Because of the specific metric of the conformal space, the additional basis vectors e_o^2, e_∞^2 square to 0 and their inner product results in $e_\infty \cdot e_o = -1$. Based on these specific properties the inner product between a Conformal vector U and a Conformal vector V is defined by

$$U \cdot V = (\mathbf{u} + u_4 e_\infty + u_5 e_o) \cdot (\mathbf{v} + v_4 e_\infty + v_5 e_o)$$

also expressible as

$$U \cdot V = \mathbf{u} \cdot \mathbf{v} - u_5 v_4 - u_4 v_5 \tag{7}$$

or

$$U \cdot V = u_1 v_1 + u_2 v_2 + u_3 v_3 - u_5 v_4 - u_4 v_5$$

In the case of U and V both being points we get

$$U \cdot V = -\frac{1}{2}(\mathbf{s} - \mathbf{p})^2$$

We recognize that the square of the Euclidean distance of the inhomogenous points corresponds to the inner product of the homogenous points multiplied by -2.

$$(\mathbf{s} - \mathbf{p})^2 = -2(U \cdot V) \tag{8}$$

For a vector U representing a point and a vector V representing a plane with normal vector $\mathbf{n}$ and distance d we get according to equation (7)

$$U \cdot V = P \cdot \pi = \mathbf{p} \cdot \mathbf{n} - d \tag{9}$$

representing the Euclidean distance of the point and the plane. Note that the scalar product $\mathbf{p} \cdot \mathbf{n}$ describes the distance of the plane from the origin. Subtraction of d results in the Euclidean plane to point distance.
For a vector U representing a plane with normal vector $\mathbf{n}$ and origin distance d and a vector V representing a sphere we get according to equation (7)

$$U \cdot V = \pi \cdot S = \mathbf{n} \cdot \mathbf{s} - d \tag{10}$$

representing the Euclidean distance of the sphere center from the plane.
For two vectors S_1 and S_2 representing two spheres we get

$$2(S_1 \cdot S_2) = r_1^2 + r_2^2 - (\mathbf{s_2} - \mathbf{s_1})^2 \tag{11}$$

This means that twice the inner product of two spheres equals the sum of the square of the radii minus the square of the Euclidean distance of the sphere centers.

We will see now that the inner product of a point and a sphere can be used for the decision of whether a point is inside or outside of a sphere. For a vector P representing a point (sphere with radius 0) and a vector S representing a sphere with radius r we get according to equation (11)

$$2(P \cdot S) = r^2 - (\mathbf{s} - \mathbf{p})^2 \tag{12}$$

That is equal to the square of the radius minus the square of the distance between the point and the center point of the sphere.
Based on this observation we can see that for

$P \cdot S > 0$: $\mathbf{p}$ is inside of the sphere
$P \cdot S = 0$: $\mathbf{p}$ is on the sphere
$P \cdot S < 0$: $\mathbf{p}$ is outside of the sphere

Angles between two objects o_1, o_2 like two lines or two planes can be computed using the inner product of the normalized direct representation of the objects.

$$cos(\theta) = \frac{o_1^* \cdot o_2^*}{|o_1^*|\,|o_2^*|} \tag{13}$$

or

$$\theta = \angle(o_1, o_2) = \arccos \frac{o_1^* \cdot o_2^*}{|o_1^*|\,|o_2^*|} \tag{14}$$

Please refer to (Doran and Lasenby, 2003) for more details.

Let us derive as one example an expression for the angle between two planes based on the observation of equation (7). For a vector π_1 representing a plane with normal vector $\mathbf{n_1}$ and distance d_1 we get

$$\mathbf{u} = \mathbf{n_1}, \;\; u_4 = d_1, \;\; u_5 = 0$$

For a vector π_2 representing another plane we get

$$\mathbf{v} = \mathbf{n_2}, \;\; v_4 = d_2, \;\; v_5 = 0$$

The inner product of the two planes is

$$\pi_1 \cdot \pi_2 = \mathbf{n_1} \cdot \mathbf{n_2} \tag{15}$$

representing the scalar product of the two normals of the planes.
Based on this observation the angle θ between two planes can be computed as follows

$$cos(\theta) = \pi_1 \cdot \pi_2 \tag{16}$$

This corresponds to equation (14) taking into account that the planes are normalized and that the dualization operation only switches between the two possible angles between planes.

Please notice that the same is also true for two circles. In this case the inner product describes the cosine of the respective carrier planes of the two circles.

4 DIFFERENTIAL GEOMETRY

In order to analyze point clouds some properties of conformal geometric algebra are very helpful. Assuming some kind of curvature information of point clouds the easy handling of geometric objects like circles and lines can be used for the local description of curvature. Based on these local properties the existence of geometric objects like cylinder, sphere, cone or torus can be investigated.

(Adamson, 2007) already developed algorithms to estimate local curvature information including principal curvatures. These can be very easily be transferred into algebraic expressions describing locally fitting objects. These objects include

- osculating circle (see figure 2)
- line describing vanishing curvature (see figure 3)
- osculating circle with vanishing radius (see figure 1)

and can be treated very consistently in conformal geometric algebra since all these objects have the same algebraic structure. They are bivectors easy to compute with.

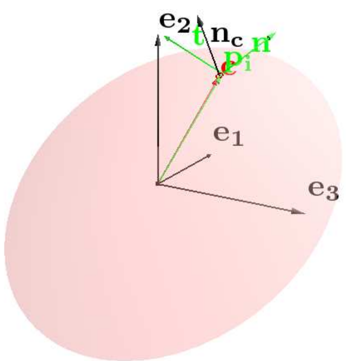

Figure 1: Local coordinate system at point p_i based on the tangent vector **t** and the normal vector **n**.

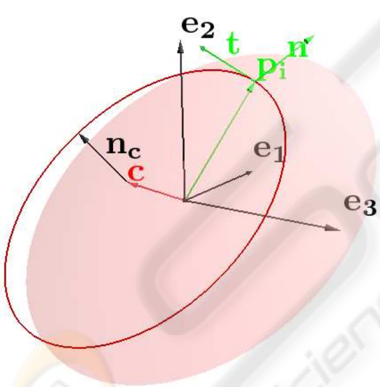

Figure 2: Osculating circle describing the curvature at point p_i.

Circles can be described with the help of the outer product of 3 points lying on the the circle or as the intersection of a sphere and a plane resulting in the following formula

$$Z = (\mathbf{c} \times \mathbf{n_c})e_{123} - \mathbf{n_c} \wedge e_0 - (\mathbf{c} \cdot \mathbf{n_c})e_\infty \wedge e_0 \\ +[(\mathbf{c} \cdot \mathbf{n_c})\mathbf{c} - \frac{1}{2}(\mathbf{c}^2 - r^2)\mathbf{n_c}] \wedge e_\infty$$

Lines can be described with the help of the outer product of 2 planes or as

$$L = \mathbf{u}e_{123} + \mathbf{m} \wedge e_\infty \tag{17}$$

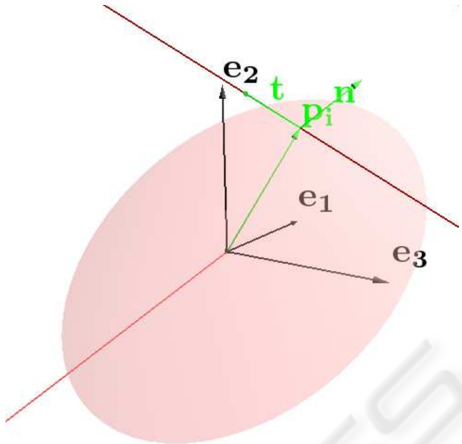

Figure 3: Line describing vanishing curvature at point p_i.

with $\mathbf{u} = \mathbf{b} - \mathbf{a}$ as Euclidean direction vector and $\mathbf{m} = \mathbf{a} \times \mathbf{b}$ as the moment vector. The corresponding six Plücker coordinates are

$$(\mathbf{u} : \mathbf{m}) = (u_1 : u_2 : u_3 : m_1 : m_2 : m_3). \tag{18}$$

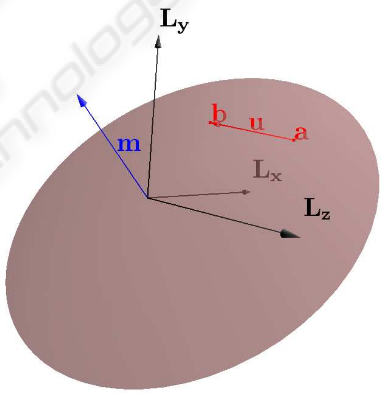

Figure 4: Plücker parameters **u** and **m** of the line through **a** and **b**.

5 FITTING OF POINTS WITH THE HELP OF A SPHERE

While in (Hildenbrand, 2005) fitting of spheres or planes into point clouds is described, in this section a point cloud $\mathbf{p_i} \in \mathbb{R}^3$, $i \in \{1, \ldots, n\}$ will be approximated specifically with the help of a sphere. The inhomogenous points $\mathbf{p_i}$ are represented as

$$P_i = \mathbf{p_i} + \frac{1}{2}\mathbf{p_i}^2 e_\infty + e_0 \tag{19}$$

and the sphere S with inhomogenous center point **s** and radius r is represented according to equation (3).

5.1 Approach

In order to solve the approximation problem we

- define a distance measure between point and sphere with the help of the inner product.
- make a least squares approach to minimize the squares of the distances between the points and the sphere.
- solve the resulting linear system of equations.

5.2 Distance Measure

According to section 3 the inner product between a point P_i and the sphere S

$$P_i \cdot S = (\mathbf{p_i} + \frac{1}{2}\mathbf{p_i}^2 e_\infty + e_0) \cdot (\mathbf{s} + s_4 e_\infty + e_0) \quad (20)$$

can be used as a measure for their distance. This results in

$$P_i \cdot S = \mathbf{p_i} \cdot \mathbf{s} - \frac{1}{2}\mathbf{p_i}^2 - s_4$$

according to equation (7) which can be written as

$$P_i \cdot S = w_{i,1}s_1 + w_{i,2}s_2 + w_{i,3}s_3 + w_{i,4}s_4 + w_{i,5}$$

or

$$P_i \cdot S = \sum_{j=1}^{4}(w_{i,j}s_j) + w_{i,5} \quad (21)$$

with

$$w_{i,k} = \begin{cases} p_{i,k} & : \; k \in \{1,2,3\} \\ -1 & : \; k = 4 \\ -\frac{1}{2}\mathbf{p_i}^2 & : \; k = 5 \end{cases}$$

5.3 Least Squares Approach

In the least-squares sense we consider the minimum of the squares of the distances between all the points and the sphere

$$min \sum_{i=1}^{n} (P_i \cdot S)^2 \quad (22)$$

In order to obtain the minimum we have the following 4 necessary conditions

$$\forall k \in \{1..4\} : \frac{\partial(\sum_{i=1}^{n}(P_i \cdot S)^2)}{\partial s_k} = \sum_{i=1}^{n} \frac{\partial(P_i \cdot S)^2}{\partial s_k} = 0 \quad (23)$$

With the help of

$$\frac{\partial(P_i \cdot S)^2}{\partial s_k} = 2(P_i \cdot S) \cdot \frac{\partial(P_i \cdot S)}{\partial s_k}$$

and

$$\frac{\partial(P_i \cdot S)}{\partial s_k} = \frac{\partial(\sum_{j=1}^{4}(w_{i,j}s_j) + w_{i,5})}{\partial s_k} = w_{i,k}$$

we obtain

$$\forall k \in \{1..4\} : \sum_{i=1}^{n} \frac{\partial(P_i \cdot S)^2}{\partial s_k} = 2\sum_{i=1}^{n}(\sum_{j=1}^{4}(w_{i,j}s_j w_{i,k}) + w_{i,5}w_{i,k}) = 0$$

or

$$\forall k \in \{1..4\} : \sum_{i=1}^{n}\sum_{j=1}^{4}(w_{i,j}w_{i,k}s_j) = -\sum_{i=1}^{n}(w_{i,5}w_{i,k})$$

which is the same as

$$\forall k \in \{1..4\} : \sum_{j=1}^{4}\sum_{i=1}^{n}(w_{i,j}w_{i,k}s_j) = -\sum_{i=1}^{n}(w_{i,5}w_{i,k})$$

or

$$\forall k \in \{1..4\} : \sum_{j=1}^{4} s_j \sum_{i=1}^{n}(w_{i,j}w_{i,k}) = -\sum_{i=1}^{n}(w_{i,5}w_{i,k})$$

leading to the following linear equation system

$$\begin{pmatrix} \sum w_{i,1}w_{i,1} & \sum w_{i,2}w_{i,1} & \sum w_{i,3}w_{i,1} & \sum w_{i,4}w_{i,1} \\ \sum w_{i,1}w_{i,2} & \sum w_{i,2}w_{i,2} & \sum w_{i,3}w_{i,2} & \sum w_{i,4}w_{i,2} \\ \sum w_{i,1}w_{i,3} & \sum w_{i,2}w_{i,3} & \sum w_{i,3}p_{i,3} & \sum w_{i,4}w_{i,3} \\ \sum w_{i,1}w_{i,4} & \sum w_{i,2}w_{i,4} & \sum w_{i,3}w_{i,4} & \sum w_{i,4}w_{i,4} \end{pmatrix} \cdot s = \begin{pmatrix} -\sum w_{i,5}w_{i,1} \\ -\sum w_{i,5}w_{i,2} \\ -\sum w_{i,5}w_{i,3} \\ -\sum w_{i,5}w_{i,4} \end{pmatrix}$$

For a fitting sphere, the result of the least squares approach is as follows :

$$\begin{pmatrix} \sum p_{i,1}p_{i,1} & \sum p_{i,2}p_{i,1} & \sum p_{i,3}p_{i,1} & -\sum p_{i,1} \\ \sum p_{i,1}p_{i,2} & \sum p_{i,2}p_{i,2} & \sum p_{i,3}p_{i,2} & -\sum p_{i,2} \\ \sum p_{i,1}p_{i,3} & \sum p_{i,2}p_{i,3} & \sum p_{i,3}p_{i,3} & -\sum p_{i,3} \\ -\sum p_{i,1} & -\sum p_{i,2} & -\sum p_{i,3} & \sum 1 \end{pmatrix} \cdot s = \begin{pmatrix} \frac{1}{2}\sum \mathbf{p_i}^2 p_{i,1} \\ \frac{1}{2}\sum \mathbf{p_i}^2 p_{i,2} \\ \frac{1}{2}\sum \mathbf{p_i}^2 p_{i,3} \\ -\frac{1}{2}\sum \mathbf{p_i}^2 \end{pmatrix}$$

with $p_{i,1}, p_{i,2}, p_{i,3}$ as inhomogenous coordinates of the points $\mathbf{p}_i$. The result $s = (s_1, s_2, s_3, s_4)$ represents the center point of the sphere (s_1, s_2, s_3) and its radius in terms of $r^2 = s_1^2 + s_2^2 + s_3^2 - 2s_4$

6 BOUNDING SPHERE ALGORITHM

The problem of defining a bounding sphere of a point cloud can be subdivided into three sub-problems:

1. How to enclose a set of points by a minimal sphere.
2. How to minimally expand an existing bounding sphere when adding more points.
3. How to merge existing bounding spheres.

Because points can be treated as spheres of zero radius, case 2 becomes part of case 3 if the latter is solved for bounding sheres of general radii, including the radius zero.

6.1 Point Clouds with One, Two or Three Points

Conformal geometric algebra adds an origin-infinity plane to 3D Eucldean space. The origin-infinity plane is given by its bivector blade $E = e_\infty \wedge \mathbf{e}_0$. The blade E can be used to extract pure Euclidean parts from conformal multivectors M

$$M_{Euclid} = (M \wedge E) \cdot E\,. \tag{24}$$

We represent spheres by 5D vectors S in the $\mathcal{G}_{4,1}$ conformal model of three-dimensional Euclidean space. In general the conformal model uses blades A_V for the socalled inner product null space representation (IPNS) of conformal subspaces V,

$$X \cdot A_V = 0 \Leftrightarrow X \in V, \tag{25}$$

which is dual to the direct outer product null space representation (OPNS)

$$X \wedge A_V^* = 0 \Leftrightarrow X \in V, \qquad A_V^* = A_V I_5^{-1} = -A_V e_{123} E. \tag{26}$$

Section 2 gave examples of blades A_V for circles and lines and Equ. (29) shows how to obtain a sphere (blade) vector ($A_V = S$).

If the cloud consists of only one point

$$P = \mathbf{p} + \frac{1}{2}\mathbf{p}^2 e_\infty + e_0, \tag{27}$$

this point defines its own bounding sphere with conformal center $C = P$ and radius zero $r = 0$.

If the cloud consists of two points,

$$P_k = \mathbf{p}_k + \frac{1}{2}\mathbf{p}_k^2 e_\infty + e_0, \quad 1 \leq k \leq 2, \tag{28}$$

the minimal bounding sphere has P_1 and P_2 as its poles. The conformal sphere vector (see Fig. 5) is then given by

$$S = \frac{1}{2}(P_1 + P_2). \tag{29}$$

The factor one half is convenient for keeping the inner product of S with e_∞ to

$$S \cdot e_\infty = -1. \tag{30}$$

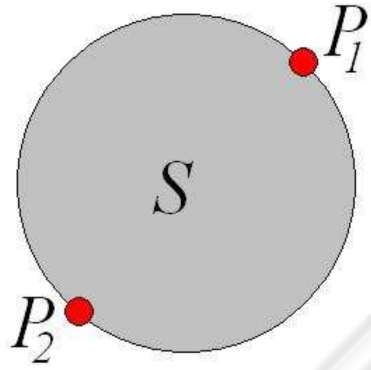

Figure 5: Sphere vector constructed from two pole points.

This is not necessary, because the conformal objects are homogeneous, but helpful regarding numerical implementations. We can always *norm* a sphere by

$$S \rightarrow \frac{S}{-S \cdot e_\infty} \tag{31}$$

and thus achieve (30). The radius of a normed sphere is given by

$$r^2 = S^2. \tag{32}$$

The conformal center of a normed sphere is then given by

$$C = S + \frac{1}{2} r^2 e_\infty. \tag{33}$$

The Euclidean center vector of a sphere is given according to (24) by

$$\mathbf{c} = (C \wedge E) \cdot E. \tag{34}$$

In the case of three conformal points

$$P_k = \mathbf{p}_k + \frac{1}{2}\mathbf{p}_k^2 e_\infty + e_0, \quad 1 \leq k \leq 3, \tag{35}$$

we can first define an initial sphere with two points (e.g. P_1, P_2) as in Equ. (29). Then we can regard the third point as a second sphere with zero radius and center P_3 and apply the method for the bounding of two spheres described in subsection 6.3. Or we can directly expand the sphere to minimally include the third point in the following way.

6.2 Minimally Including a New Point

We describe this alternative (compared to subsection 6.3) way in order to show that geometric algebra offers a variety of algorithmic constructions, some of which may be preferable for specific tasks and for numerical optimization.

We show two variants in the form of CLUCalc Scripts. The first more from a geometric algebra principle point of view, the second based on further code performance optimization with Maple.

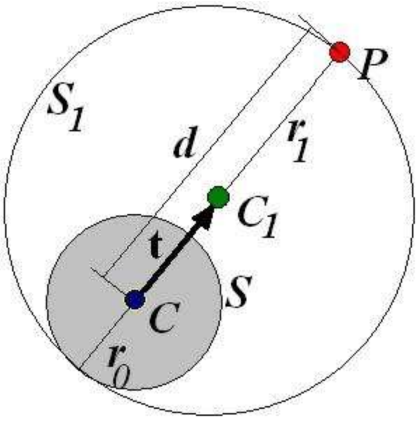

Figure 6: Minimally adding a point P to a sphere S by adjusting center and radius.

```
DefVarsN3(); // se of conformal model
:IPNS;  // Use of inner product null space repr.
C = VecN3(c1,c2,c3); // Conformal sphere center
r=r0;   // Sphere radius
S =C-0.5*r*r*einf;
        //Definition of initial sphere S
P=VecN3(p1,p2,p3);  // New point outside S
d = sqrt(-2*P.C);
   //Distance between sphere center C and P
if (d>r){
  CP = (p1-c1)*e1 + (p2-c2)*e2 + (p3-c3)*e3;
     // Vector CP
  T  = 1 + 0.5* 0.5*(1-r/d)*CP *einf;
     // Translator to new center
  C1 = ~T*C*T;
     // Center of new sphere
  r1 = (d+r)/2;
     // Radius of new sphere
  S1 = C1 - 0.5*r1*r1*einf;
    // Def. of new sphere
}
```

We see in the CLUCalc Script that the sphere is only expanded if the new point P is outside the sphere S ($d > r$). C_1 is the center of the expanded sphere S_1, obtained by shifting C in the direction of P by $\mathbf{t} = \frac{1}{2}(d-r)\frac{CP}{|CP|}$, because $d = |CP|$. $r_1 = (d+r)/2$ is the radius of the minimally expanded sphere S_1. Compare Fig. 6.

For numerical optimization we can replace the definition of d and the if loop by Maple optimized code

```
d=sqrt((p1-c1)*(p1-c1)+(p2-c2)*(p2-c2)+(p3-c3)*(p3-c3));

if (d>r){
  c1x = (-r*p1+r*c1+p1*d+c1*d)/d/2;
  c1y = ( r*c2+p2*d+c2*d-r*p2)/d/2;
  c1z = ( p3*d-r*p3+r*c3+c3*d)/d/2;
  r1  = (d+r)/2;
  S1  = VecN3(c1x,c1y,c1z) - 0.5*r1*r1*einf;}
```

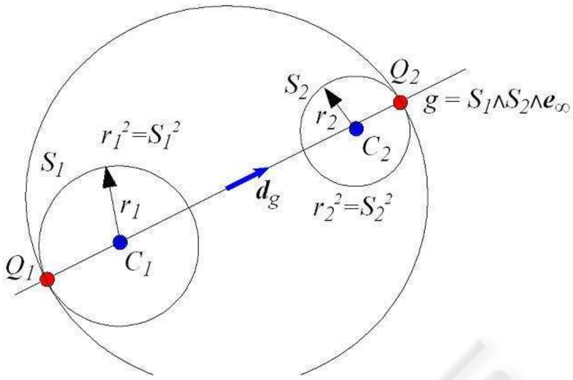

Figure 7: Minimal bounding sphere of two spheres.

6.3 Bounding Two Spheres

We can merge (bound) two spheres by defining a straight line g which connects the centers of the two (normed) spheres S_1 and S_2

$$g^* = (S_1 \wedge S_2 \wedge e_\infty)^*. \tag{36}$$

The Eulidean unit vector in the direction of g is then

$$\mathbf{d} = \frac{g^* \cdot e_{123}}{|g^* \cdot e_{123}|}. \tag{37}$$

We now calculate the poles of the minimal bounding sphere. The first pole is the point of intersection of g with S_1 away from S_2

$$\mathbf{q}_1 = \mathbf{c}_1 - r_1\mathbf{d}, \tag{38}$$

with $\mathbf{c}_1$ and r_1 calculated according to (34) and (32). We similarly obtain the second pole of the bounding sphere as the point of intersection of g with S_2 away from S_1

$$\mathbf{q}_2 = \mathbf{c}_2 + r_2\mathbf{d}, \tag{39}$$

The corresponding conformal poles (27) give according to (29) the mininal bounding sphere (see Fig. 7)

$$S_{12} = \frac{1}{2}(Q_1 + Q_2). \tag{40}$$

In this section we developed the algebraic expressions for the minimal bounding sphere [eqs. (36) to (40)] for easy numerical implementation. It would be possible to completely carry out the calculation of S_{12} on the conformal level. For that purpose we would first calculate the conformal point pairs (OPNS) of intersection of line g with sphere S_1

$$pair_1 = g \cdot S_1 \tag{41}$$

and of line g with sphere S_2

$$pair_2 = g \cdot S_2. \tag{42}$$

With

$$Q_1 = \frac{pair_1 + |pair_1|}{pair_1 \cdot e_\infty} \tag{43}$$

and

$$Q_2 = \frac{pair_2 - |pair_2|}{pair_2 \cdot e_\infty} \tag{44}$$

we can then directly pick the conformal points Q_1 and Q_2 out of the point pairs $pair_1$ and $pair_2$.

If (like in subsection 6.2) the second sphere S_2 happens to be only a point (sphere with zero radius), we can omit the calculation of Q_2 in Equs. (39), (42) and (44). We simply replace $Q_2 = S_2$ in Equ. (40).

6.4 Comparison with Welzl's Bounding Sphere Algorithm

The iteration of the method suggested in subsection 6.3 (test of inclusion, and if necessary the calculation of the new bounding sphere) results in the final bounding sphere of n points in linear time $O(n)$. The proposed method is easy to understand and with given routines for inner and outer products easy to implement. As demonstrated in subsection 6.2 the algorithm can be further optimized with Maple. In average Welzl's algorithm (Welzl, 1991) also runs in asymptotically linear time, but the recursion in Welzl's algorithm makes it harder to examine and guarantee the performance time.

7 CONCLUSIONS

We presented a bunch of basics and algorithms expressed in the mathematical framework of conformal geometric algebra. We are convinced that its easy handling of geometric objects like spheres, circles or planes, its easy handling of distances and angles between them as well as its way of fitting and bounding of geometric objects will provide a promising foundation for the analysis of point clouds.

REFERENCES

Adamson, A. (2007). *Computing Curves and Surfaces from Points*. PhD thesis, Darmstadt University of Technology.

Doran, C. and Lasenby, A. (2003). *Geometric Algebra for Physicists*. Cambridge University Press.

Dorst, L. and Mann, S. (2002). Geometric algebra: a computational framework for geometrical applications (part i: algebra). *Computer Graphics and Application*, 22(3):24–31.

Fontijne, D. and Dorst, L. (2003). Modeling 3D euclidean geometry. *IEEE Computer Graphics and Applications*, 23(2):68–78.

Hildenbrand, D. (2005). Geometric computing in computer graphics using conformal geometric algebra. *Computers & Graphics*, 29(5):802–810.

Hildenbrand, D., Fontijne, D., Perwass, C., and Dorst, L. (2004). Tutorial geometric algebra and its application to computer graphics. In *Eurographics conference Grenoble*.

Hildenbrand, D., Fontijne, D., Wang, Y., Alexa, M., and Dorst, L. (2006). Competitive runtime performance for inverse kinematics algorithms using conformal geometric algebra. In *Eurographics conference Vienna*.

L.Dorst, Fontijne, D., Mann, S., and Kaufman, M. (2007). *Geometric Algebra for Computer Science, An Object-Oriented Approach to Geometry*. Morgan Kaufman.

Mann, S. and Dorst, L. (2002). Geometric algebra: a computational framework for geometrical applications (part ii: applications). *Computer Graphics and Application*, 22(4):58–67.

Rosenhahn, B. (2003). *Pose Estimation Revisited*. PhD thesis, Inst. f. Informatik u. Prakt. Mathematik der Christian-Albrechts-Universität zu Kiel.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

Welzl, E. (1991). Smallest enclosing disks (balls and ellipsoids. In *Lecture Notes in Computer Science*, pages 555:359–370.