Adv. appl. Clifford alg. 18 (2008), 523–538

0188-7009/030523-16, *published online* May 19, 2008
DOI 10.1007/s00006-008-0084-9

Advances in Applied Clifford Algebras

Oriented Conformal Geometric Algebra

Jonathan Cameron and Joan Lasenby

Abstract. In [16] Stolfi developed a complete theory of *Oriented Projective Geometry*. He showed that assigning meaning to the sign of an otherwise homogeneous representation of geometry could provide a multitude of benefits. This paper extends his work by applying the same approach to Conformal Geometric Algebra. Oriented Conformal Geometric Algebra allows intuitive manipulation of such concepts as half-spaces, inclusion within geometric entities and ordered intersections. It also illustrates the non-commutative nature of the meet. The paper concludes with some examples of applications in which Oriented Conformal Geometric Algebra is already providing benefits.

Mathematics Subject Classification (2000). 15A66, 51A05.

Keywords. Conformal geometric algebra, oriented projective geometry, interpolation.

1. Introduction

1.1. Oriented Projected Geometry

Stolfi [16] observed that many of the algorithms in common use in computer graphics and computer vision, made use of the sign of the 4th component of the supposedly homogeneous coordinates of projective geometry. He therefore created a complete theory of Oriented Projective Geometry in which the sign of the previously homogeneous vectors is used to give a concept of direction. Thus $(x, y, z, \omega) \neq (-x, -y, -z, -\omega)$ but instead is defined as a point of opposite orientation. Whilst this abstract concept of orientation of a point does not provide many advantages, the orientation of higher dimensional entities reflects their direction e.g. two concurrent lines of opposite orientation can be considered to point in opposite directions.

This paper replicates and extends the results of Stolfi's work by giving meaning to the sign of a blade in Conformal Geometric Algebra (CGA) [8, 14, 12]. To maintain consistency with the terminology selected by Stolfi, this extension of CGA utilizing directed / signed entities will be termed Oriented CGA or OCGA.

Some elements of OCGA have previously been presented within the literature [15, 6, 9].

1.2. Conformal Geometric Algebra

There is insufficient space here to provide a comprehensive introduction to Conformal Geometric Algebra. We refer the reader to [12, 5].

In this paper we have elected to use the following formulation and notation. Let us augment the 3 spatial basis vectors $e_{1,2,3}$ defined such that $(e_i)^2 = 1$ with an additional pair $e_{4,5}$ where $(e_4)^2 = 1$ and $(e_5)^2 = -1$. We can now define a pair of useful null vectors

$$n = e_4 + e_5 \qquad\qquad \bar{n} = e_4 - e_5 \tag{1.1}$$

n can be shown to correspond to the point at infinity whilst $\bar{n}$ corresponds to the origin[1].

The Hestenes' transform applied to a 3D spatial vector x is given by

$$X = H(x) = \frac{1}{2}\left(x^2 n + 2x - \bar{n}\right) \tag{1.2}$$

so as to give us a null vector representation of points in space. Note that throughout this paper, spatial vectors are indicated via lower case non-bold letters, whilst their Hestenes' transforms are denoted by capital letters.

We also adopt the notation, e_{abcd} to denote higher grade basis elements

$$e_{abcd} = e_a \wedge e_b \wedge e_c \wedge e_d \tag{1.3}$$

and $[]^*$ to indicate multiplication by the positively oriented pseudo-scalar, commonly referred to as the dual operator.

2. Blades in OCGA

As with unoriented CGA, blades are defined as the outer product of a number of null vectors and provide representations of various geometric elements.

2.1. 1-blade – vector

In this paper it will be assumed that all null-vectors, X, are defined such that $X \cdot n = -1$ unless otherwise stated. Note that an equivalent set of results can be built up with the default orientation being negative such that $X \cdot n = 1$.

[1]Strictly speaking $-\frac{1}{2}\bar{n}$ corresponds to the origin.

2.2. 2-blade – oriented point pair

As noted in [12](Sec. 4.1.1.) and elsewhere, it is possible to factorize a 2-blade into a pair of null vectors that are unique up to a scale. This result holds equally well for OCGA but consideration must be given to the various 2-blades which will give the same pair of null vectors after separation as they are effectively identical. Note that there is no need to make use of Stolfi's notation, $\neg$, to represent the operation of reversing the orientation of a blade as in OCGA this negation operator simply corresponds to multiplying the blade in question by -1.

$$A \wedge B = -(B \wedge A) = -B \wedge A = B \wedge -A \tag{2.1}$$

Of the above forms, the first is the most useful, consisting as it does, of a pair of 'positively' oriented null vectors. In CGA these 2-blades represent unordered pairs of points. However, the use of projectors in the standard separation method [12](Sec. 4.1.1.) ensures that one projector P will always return the first point, and the second $P^{\sim}$ will return the second. As a result, we can assign an orientation to these two points, considering one to occur before the other. The usefulness of this result will become apparent when intersections are considered.

2.3. 3-blade – oriented lines

Define the line with direction from A to B as

$$\begin{aligned} L &= A \wedge B \wedge n \\ &= B \wedge n \wedge A \\ &= n \wedge A \wedge B \end{aligned}$$

Algebraically it can be trivially shown that these 3 orderings give the same blade. OCGA also provides a geometric justification for this cyclic reordering property. Consider a ray passing through the 3 points in space, a, b, and n, in the order given. As all straight lines can be considered as circles passing through infinity, it is readily apparent that these 3 lines must be both concurrent and pointing in the same direction. Hence, they all have the same orientation.

Due to the anti-commuting nature of the outer product of vectors, $L' = B \wedge A \wedge n = -L$ would give the line in the opposite direction differing only in sign. In CGA these two lines are considered to be different representations of the same entity, but as we shall see this results in a loss of representational power.

Half space of line defined by a point

A half-space is simply the set of points within an n-dimensional geometric entity that lie on one side of an $n-1$ dimensional geometric entity. In the case of the half space of a line L, the separating entity is simply an oriented point with null representation A, lying on the line. The oriented line and point in combination allow the definition of an inclusion test t_1 for a point with null representation P.

$$t_1 = (P \wedge A \wedge n)L \tag{2.2}$$

where the point is defined as inside the half-space if the orientation of the line $P \wedge A \wedge n$ is the same as L and outside if it has the opposite orientation. These conditions correspond to inclusion for $t_1 > 0$ and exclusion for $t_1 < 0$. The case of $t_1 = 0$ indicates that the test point is coincident with the boundary of the half-space. Clearly, the result of this test is dependent on both the orientation of the null-vector A and that of the embedding space, in this case the line L. The dependence on the orientation of the embedding space is always the case for operations in OCGA. This point was also made by Stolfi [16] (Chap. 7) and any algorithm must carefully define the containing space or problems may occur.

Many of the results that follow rely on generalizing this half-space concept to higher dimensions.

Inclusion of a point in a line segment

To illustrate the usefulness of the half-space test given above, consider developing an inclusion test for a point within a line segment. This is easily accomplished via a pair of half-space tests. The line through a then b is given by the blade $L = A \wedge B \wedge n$. To identify whether a point p lies between a and b we simply need to check the orientations of the lines from a to p and p to b. If these orientations are the same as that of a to b then inclusion has been established.

The two tests are

$$\begin{aligned} t_1 &= (A \wedge P \wedge n)L \\ t_2 &= (P \wedge B \wedge n)L \end{aligned} \tag{2.3}$$

If both signs are positive then the orientations of the lines created are all the same and the point is indeed within the segment $a \rightarrow b$. If either of the tests equals 0, then the point is coincident with the relevant segment end point. If either is negative, then the point lies outside the segment. Thus we can rapidly check all possible options. As observed by Stolfi, it is often useful to separate the coincident point cases, as naturally occurs within these tests.

It is possible and entirely valid to generalize these results to define segments passing through infinity and in figure 1 the various possible oriented line segments between a and b are illustrated. Note that, to perform an inclusion test, simply replace, in turn, the two end points in the definition with the test point and take the geometric product with the original line as with the example tests in equation 2.3.

2.4. 3-blade – oriented circle

An analogous set of results exist for the concept of the oriented circle, with n, the point at infinity, replaced by a third point on the circle.

Half-space defined by point pair

In order to separate the set of points on a circle into a pair of regions, defined as outside and inside, we clearly need a pair of end points. In fact, if the point n is considered to be on the same footing as any other then the half-space test for a line

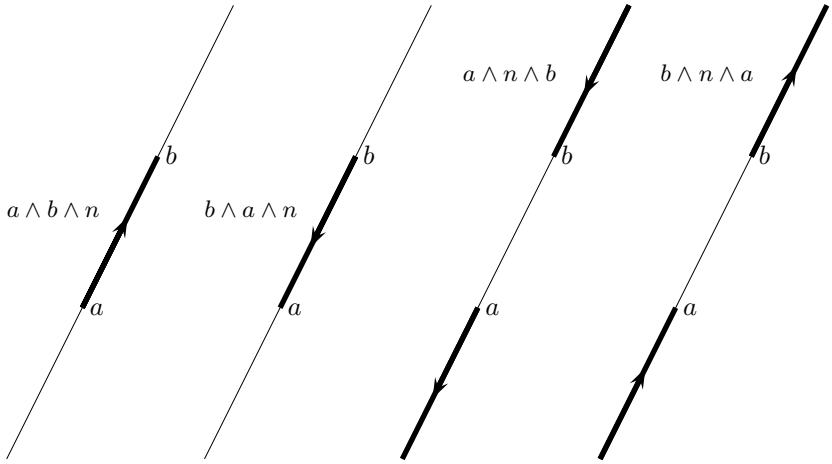


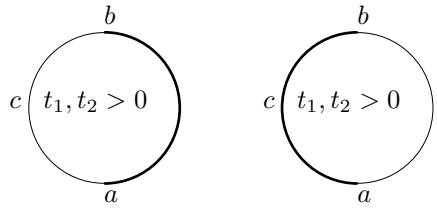
FIGURE 1. The 4 possible oriented line segments between a and b

can be more properly considered to be a segment inclusion test for the segment between the point in question and the point at infinity as approached from one side along the line.

Now for the half-space test, on the point p, for a segment bounded by a and b on a circle $\Gamma = A \wedge B \wedge C$, a pair of test functions must be evaluated

$$\begin{aligned} t_1 &= (P \wedge B \wedge C)\Gamma \\ t_2 &= (A \wedge P \wedge C)\Gamma \end{aligned} \tag{2.4}$$

Yet again, these are simply half-space tests and as such are dependent not only on the orientation of the point pair, but also on the orientation of the embedding circle Γ as illustrated in figure 2. Again included points are defined as those for which the two circles $P \wedge B \wedge C$ and $A \wedge P \wedge C$ have the same orientation as Γ and hence $t_1, t_2 > 0$.



(a) $\Gamma = A \wedge B \wedge C$ (b) $\Gamma = B \wedge A \wedge C$

FIGURE 2. Circular arc definition for inclusion tests – endpoints a and b

2.5. 4-blade – planes

The results for lines given above generalize directly to include all hyperplanes. As in non-oriented CGA a plane can be defined from any three points on its surface.

Half-space of plane defined by oriented 3-blade

Having now introduced the 4-blade, we have a suitable embedding entity in which to consider the concept of half-spaces as defined by lines or circles lying in any plane.

Any line L lying on a given plane Φ, inherently divides the plane into two separate regions. These can be considered to be the left and the right of the line although that particular terminology is slightly misleading given the dependence on which side the plane is being viewed from. One way of discovering whether a point, null representation P, on the plane, Φ, lies on one side of the line, L, or the other is to consider the sign of

$$t_1 = (P \wedge L)\Phi \tag{2.5}$$

Yet again the inclusion test is dependent both on the orientation of the entity defining the edge of the half-space and on the orientation of the embedding space. If $t_1 < 0$, then the plane formed by $P \wedge L$ is concurrent with Φ and has the same orientation and thus the point is contained within the half-space.

Note, that the above definition is still entirely valid if the half space is defined by a circle instead of a line.

Triangular facet inclusion

Inclusion within a triangular facet can be considered as a set of 3 half-space tests as illustrated in figure 3.

This method is considerably more intuitive than the reciprocal vector based method given in [12].

Accelerating the facet inclusion test

In applications such as ray-tracing it is often desirable to perform a lot of ray–facet intersection tests for a given facet. These tests can be considerably accelerated through some precomputation. Consider the form of the 3 half-space inclusion tests. Each can be expressed in the form:

$$(P \wedge L) \cdot \pm\Phi < 0 \tag{2.6}$$

if we express L and Φ as

$$\begin{aligned} L = {} & L_1 e_{123} + L_2 e_{124} + L_3 e_{125} + L_4 e_{134} + L_5 e_{135} \\ & + L_6 e_{145} + L_7 e_{234} + L_8 e_{235} + L_9 e_{245} + L_{10} e_{345} \end{aligned} \tag{2.7}$$

$$\Phi = \Phi_1 e_{1234} + \Phi_2 e_{1235} + \Phi_3 e_{1245} + \Phi_4 e_{1345} + \Phi_5 e_{2345} \tag{2.8}$$

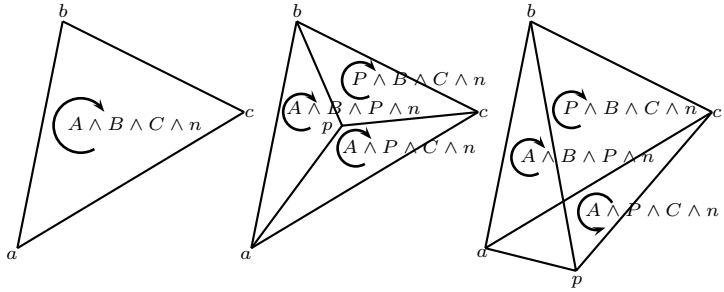


FIGURE 3. Facet containment test: note how for an internal point all the test planes formed have same orientation as that defined by the original facet, whereas with an external point at least one will have opposite orientation (e.g. $A \wedge P \wedge C \wedge n$)

Then we can form a new non-null vector D such that the $(P \wedge L) \cdot \Phi = P \cdot D$ where D is given by

$$\begin{aligned} D = &(L_7\Phi_1 - L_8\Phi_2 - L_9\Phi_3 - L_{10}\Phi_4)\, e_1 \\ &+ (L_5\Phi_2 - L_4\Phi_1 + L_6\Phi_3 - L_{10}\Phi_5)\, e_2 \\ &+ (L_2\Phi_1 - L_3\Phi_2 + L_6\Phi_4 + L_9\Phi_5)\, e_3 \\ &- (L_1\Phi_1 + L_3\Phi_3 + L_5\Phi_4 + L_8\Phi_5)\, e_4 \\ &- (L_1\Phi_2 + L_2\Phi_3 + L_4\Phi_4 + L_7\Phi_5)\, e_5 \end{aligned} \tag{2.9}$$

Thus a triangular facet intersection test can be reduced down to 3 $5D$ dot products and 3 comparisons with 0. It is worth noting here, that at no stage does this derivation assume a particular form for either of L or Φ and so is equally applicable to performing half-space tests on the surface of a sphere with the boundaries defined by either great or small circles.

The general approach used above to accelerate the facet inclusion test can be applied to many of the results of OCGA and can lead to very efficient implementations, whilst retaining the intuitive derivation of the algorithms. More examples can be found in [4].

Line–line intersection

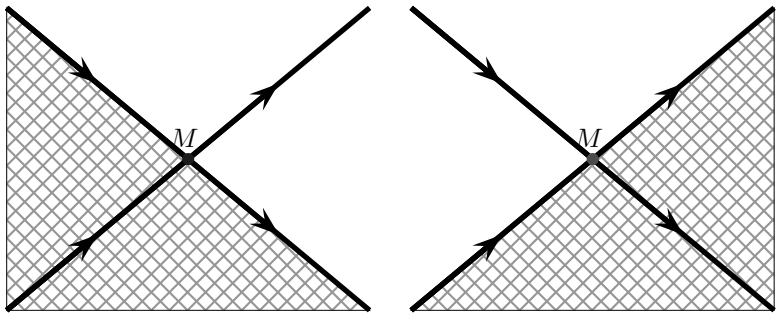
Figure 4 illustrates the operation of intersecting a pair of lines, lying in a common plane using the meet operator given in [12]. The meet of 2 entities defined by blades D_m and E_o of grades m and o, lying within an embedding entity of grade

p is given by

$$D \vee E = \left[\langle DE \rangle_{2p-m-o} \right]^* \tag{2.10}$$

From the line–line intersection example it is clear that the meet operation is not always a commuting operator in OCGA. (e.g. $A \vee B \neq B \vee A$ in general). In fact, it turns out that the commutativity of a given meet operation can be trivially found by considering what Stolfi terms the co-rank of the geometric entities involved. In this more general treatment this co-rank is perhaps more accurately termed the co-grade as it is simply the grade of the embedding space minus that of the entity in question. So if we have the meet of two blades D and E, embedded in blade F then the sign of the meet is given by

$$\begin{aligned} D \vee_F E &= (-1)^{\text{co-grade}_F(D)\text{co-grade}_F(E)} (E \vee_F D) \\ &= (-1)^{(\text{grade}(F)-\text{grade}(D))(\text{grade}(F)-\text{grade}(E))} (E \vee_F D) \end{aligned} \tag{2.11}$$



(a) $L_1 \vee L_2 = M \wedge n$ (b) $L_2 \vee L_1 = n \wedge M$

FIGURE 4. Illustration of a non-commuting case of the meet – two lines within an embedding space in one of two possible orientations

2.6. 4-blade – spheres

As in CGA a sphere can be defined as the outer product of 4 null vectors specifying a non-degenerate set of points on its surface. The concept of direction gets marginally more confusing for a set of 4 points. To avoid confusion it is often easier to simply form the sphere then negate as necessary to ensure n is inside or outside as desired.

2.7. 5-blade – the pseudo-scalar (positive universe)

As with any other entity within OCGA the pseudo-scalar I can have two orientations and these correspond to Stolfi's concept of a oriented universe. Clearly many of the above algorithms are dependent on the sign of the enclosing space and this is equally true when that space is the pseudo-scalar. To simplify algorithm derivation, here I will be defined as e_{12345} and right multiplying by it will be termed the dual operation. Thus we also have a concept of an inverse dual as $I^2 = -1$. To keep in line with the formulation of Stolfi [16], it is useful to connect this pseudo-scalar to his concept of a positive universe.

2.8. Universe half-spaces

In this case the half-space of the universe or pseudo-scalar can be defined by a plane or sphere.

Hence, we can simply use

$$t_1 = (P \wedge A_4)I \tag{2.12}$$

to define the inclusion in the half-space defined by the 4-blade A_4. If $t_1 < 0$ then the P is inside the half-space and this test is sufficient for both planes and spheres.

Using this half-space concept we can now consider testing for tetrahedron inclusion.

Tetrahedron inclusion

The previously described ideas for facet and line segment inclusion tests extend to the full dimensionality of the space and thus provide a concise and easily understood set of 4 inclusion tests for a point, P, and a tetrahedron defined by the 4 null vectors A, B, C and D.

$$\begin{aligned} T &= A \wedge B \wedge C \wedge D \wedge n \\ t_1 &= (P \wedge B \wedge C \wedge D \wedge n)T \\ t_2 &= (A \wedge P \wedge C \wedge D \wedge n)T \\ t_3 &= (A \wedge B \wedge P \wedge D \wedge n)T \\ t_4 &= (A \wedge B \wedge C \wedge P \wedge n)T \\ &\text{iff } t_1, t_2, t_3, t_4 < 0 \text{ does point p lie within T} \end{aligned} \tag{2.13}$$

Clearly, this approach generalizes to any convex object defined in terms of facets. It is also possible to use spherical surfaces to define the faces, although in this case additional constraints may be necessary to avoid having other areas of the space defined as included within the object. Using boolean object definitions any object with spherical or flat facets can be defined and an inclusion test performed in a simple and logical fashion. Clearly the tests above also lend themselves to the acceleration method previously described for facet inclusion.

Line–sphere intersection

As with the case of line–circle intersection, described above, the bivector returned by a line–sphere intersection gives the ordered pair of intersection points of the Line with the Sphere. Here however, the meet operation commutes as the co-grade of the line within the embedding in the Universe is 2 and hence even. The bivector can be split into two null vectors and the ordering of these gives the intersection ordering.

Line–plane intersection

This proceeds in a similar fashion to that for the line-line intersection considered above, with the two intersection points (one at infinity) being ordered as one would expect. However it is not possible to utilize the standard 2-blade projector-based separation method, as one of the points is n. One approach to get round this problem is suggested in [12](Sec. 5.3) and with minor corrections it will serve us here. The meet formula $[\langle L\Phi\rangle_3]^*$ gives a bivector B. Note the presence of the $[]^*$ operator. Thus the orientation of the result is again dependent on the embedding space. The embedding space used here is the positive universe defined in Section 2.7.

$$a = \frac{1}{2}(B \wedge \bar{n}) \cdot N \quad \Longleftrightarrow \; B = n \wedge A$$
$$a = \frac{1}{2}(\bar{n} \wedge B) \cdot N \quad \Longleftrightarrow \; B = A \wedge n$$

where $N = e_4 \wedge e_5$.

Note that a is the spatial vector representing the intersection point, rather than the null vector representation, A. As a result the approach in [12] will only work if the bivector resulting from the meet formula is of form $A \wedge n$ and not $n \wedge A$, where it will flip the sign of all components.

As with line–line intersections, this ordering result allows us the option of extracting only the first point of contact (or entry point) of the directed ray with the plane rather than having to pick from the two points returned by the meet operation in standard CGA. This could lead to simplification of some algorithms, involving different operations being applied for front face / back face intersections. Care is needed with ensuring the desired orientation of the facets is achieved, but this is no more than required by many conventional methods.

Line–facet intersection

Using the above result for line–plane intersection we can extract the point of intersection of a ray (directed or otherwise) with a plane. For many applications we now want to check whether this intersection point lies inside the triangular facet from which the plane was defined. This can be easily checked using the facet inclusion test given above.

2.9. Relationship to perceptron classification

This concept of oriented halfspaces has been used extensively within the multilayer perceptron literature, where it is referred to a sidedness, as it provides a means of defining a hyperplane decision boundary between two different classes. The classification tests based on the learnt decision boundaries may be written in OCGA as

$$\begin{aligned} t_1 &= P \cdot \Phi^* \\ &= (P \wedge \Phi) I_n \end{aligned} \tag{2.14}$$

where Φ^* is the dual in the containing space I_n, of the hyperplane (or sphere) representing the classification boundary. A recent paper by Perwass et al. [15] extended this treatment to include hypersphere decision boundaries, improving the performance for little computational cost. This treatment is effectively applying the half space concepts of OCGA to generalize a result of oriented projective geometry.

3. Object to Object Interpolation

A recent result from A. Lasenby [13] gave a very simple method for interpolating between spheres and planes. The result was used in [6] within a method for generating circle splines. OCGA provides us with additional tools which prove useful in analysis of this result when used to interpolate between pairs of spheres.

For the purposes of this paper the following slightly different form of the interpolation rotor R_λ generating a sphere Σ^λ between the spheres Σ_2 to Σ_1 will be used:

$$\begin{aligned} R_\lambda &= \sqrt{{\Sigma_1}^2 {\Sigma_2}^2} - \lambda \Sigma_2 \Sigma_1 \qquad\qquad 0 \leq \lambda \leq 1 \\ \Sigma^\lambda &= R_\lambda \Sigma_1 \tilde{R}_\lambda \\ &= {\Sigma_1}^2 {\Sigma_2}^2 \left(1 - \lambda^2\right) \Sigma_1 - 2\lambda {\Sigma_1}^2 \left(\sqrt{{\Sigma_2}^2 {\Sigma_1}^2} - \lambda \left\langle \Sigma_1 \Sigma_2 \right\rangle_0 \right) \Sigma_2 \end{aligned} \tag{3.1}$$

Note that the normalization condition on the rotor has been relaxed. The derivation of equation 3.1 relies on the identity

$$\Sigma_2 \Sigma_1 \Sigma_2 = -\left(\Sigma_2\right)^2 \Sigma_1 + 2 \left\langle \Sigma_1 \Sigma_2 \right\rangle_0 \Sigma_2. \tag{3.2}$$

3.1. Validity of end points

As for any interpolation method, the first thing to verify is that it gives the correct entities at the end points of the interpolation.

$$\begin{aligned} \lambda = 0 \qquad & \Sigma^\lambda = \left({\Sigma_1}^2 {\Sigma_2}^2\right) \Sigma_1 \\ \lambda = 1 \qquad & \Sigma^\lambda = -2 {\Sigma_1}^2 \left(\sqrt{{\Sigma_1}^2 {\Sigma_2}^2} - \left\langle \Sigma_1 \Sigma_2 \right\rangle_0 \right) \Sigma_2. \end{aligned} \tag{3.3}$$

Now, as ${\Sigma_1}^2$ is always negative, the orientation of the interpolate is the same as Σ_2 iff

$$\sqrt{{\Sigma_1}^2{\Sigma_2}^2} - \langle \Sigma_1 \Sigma_2 \rangle_0 > 0. \tag{3.4}$$

Making use of the dual form of the sphere [12]

$$\Sigma_i^* = H(c_i) - \frac{1}{2}\rho_i^2 n \tag{3.5}$$

with the useful properties

$$(\Sigma_i^*)^2 = \rho_i^2 \qquad \langle \Sigma_1 \Sigma_2 \rangle_0 = \frac{1}{2}\left(c_1^2 + c_2^2 - \rho_1^2 - \rho_2^2\right) - c_1 \cdot c_2$$

and assuming that both source spheres are of the same orientation (3.4) can be expressed as

$$(\rho_1 + \rho_2)^2 - (c_1 - c_2)^2 > 0 \tag{3.6}$$

that is to say when the two source spheres overlap. If the two source spheres are of opposite orientation then clearly the interpolate end points will have the same orientation only when they do not overlap.

However, the above conditions for orientation at the end points are not sufficient to ensure that the interpolate remains of the same orientation throughout the interpolation.

Interpolation between two intersecting spheres

Consider the intersection of the two sources spheres. Assuming such an intersection exists, it will consist of points where

$$P \wedge \Sigma_1 = 0 \qquad \text{and} \qquad P \wedge \Sigma_2 = 0. \tag{3.7}$$

From equation 3.1 it is clear that the interpolate can always be expressed as a weighted sum of the two source spheres. Thus

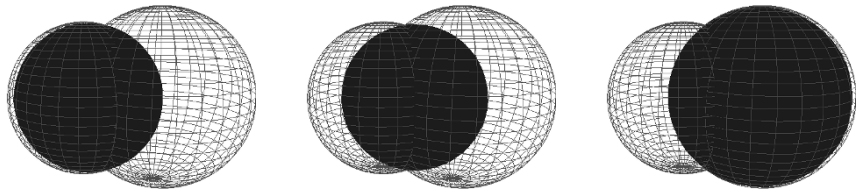
$$P \wedge \Sigma^\lambda = P \wedge (\alpha\Sigma_1 + \beta\Sigma_2) = 0. \tag{3.8}$$

Hence the interpolate always remains in contact with the intersection of the two source spheres.

Finally, in order to show no inversions occur for this special case, consider the position of the centre of the interpolate, easily found by reflecting the point at infinity in the sphere

$$\begin{aligned} c_\lambda &= H^{-1}\left(\Sigma^\lambda n \Sigma^\lambda\right) \\ &= c_2 + (c_1 - c_2)\frac{{\rho_2}^2\left(1-\lambda^2\right)}{{\rho_2}^2 + 2\lambda\rho_1\rho_2 - \lambda^2\left(c_1 - c_2\right)^2 + \lambda^2{\rho_1}^2} \\ &= c_2 + (c_1 - c_2)F(\lambda). \end{aligned} \tag{3.9}$$

Clearly the centre of the interpolate moves along the vector from the centre of one source sphere to the centre of the other. All that remains is to show that the



(a) $\lambda = 0.05$ (b) $\lambda = 0.2$ (c) $\lambda = 0.95$

FIGURE 5. An example of sphere to sphere interpolation for intersecting spheres. The wire frame spheres are Σ_1 and Σ_2 and the solid sphere is Σ^λ.

centre of the interpolate remains between the two centres of the source spheres. If so, then the interpolate remains a finite radius sphere.

Consider the values of λ for which the interpolate centre is coincident with the source sphere centres. These correspond to the conditions $F(\lambda) = 0$ and $F(\lambda) = 1$.

Taking the first condition

$$\begin{aligned} F(\lambda) &= 0 \\ (1-\lambda^2) &= 0 \\ \lambda &= \pm 1 \end{aligned} \tag{3.10}$$

for the second condition

$$\begin{aligned} F(\lambda) &= 1 \\ \lambda\left(\lambda\left(\rho_1^2+\rho_2^2-(c_1-c_2)^2\right)+2\rho_1\rho_2\right) &= 0 \\ \lambda = 0 \text{ or } \lambda\left(\rho_1^2+\rho_2^2-(c_1-c_2)^2\right)+2\rho_1\rho_2 &= 0. \end{aligned} \tag{3.11}$$

Iff $\lambda\left(\rho_1^2+\rho_2^2-(c_1-c_2)^2\right)+2\rho_1\rho_2 \neq 0$ in the range of λ, the centre never passes those of the source spheres. As this is a linear equation, establishing that there is no zero crossing in this range may be done by considering end points.
If $\lambda = 0$ this condition corresponds to

$$2\rho_1\rho_2 > 0 \tag{3.12}$$

which holds for all ρ_1, ρ_2.
If $\lambda = 1$ the condition corresponds to

$$(\rho_1+\rho_2)^2-(c_1-c_2)^2 > 0 \tag{3.13}$$

which is true for all intersecting source spheres.

Figure 5 shows an example interpolation between a pair of intersecting spheres. Clearly, the case of non-intersecting source spheres has not been considered here. A fuller treatment of the interpolation formula can be found in [4].

4. Applications

4.1. Ray-tracing

In [7] a number of different representations of geometry including Oriented Projective Geometry and CGA were compared for implementing a simple Ray-Tracer. In order to investigate possible advantages of the addition of orientation information to CGA, a simple ray-tracer was also implemented. The areas where most advantage was seen involved intersection ordering for front / back-face detection, and the ability to represent directed line segments. Subsequent discussion with D. Fontijne indicated that the implementation of [7] had made internal use of some of the elements of OCGA but the theory had not been formally presented. Figure 6 shows an example image from the OCGA ray-tracer.

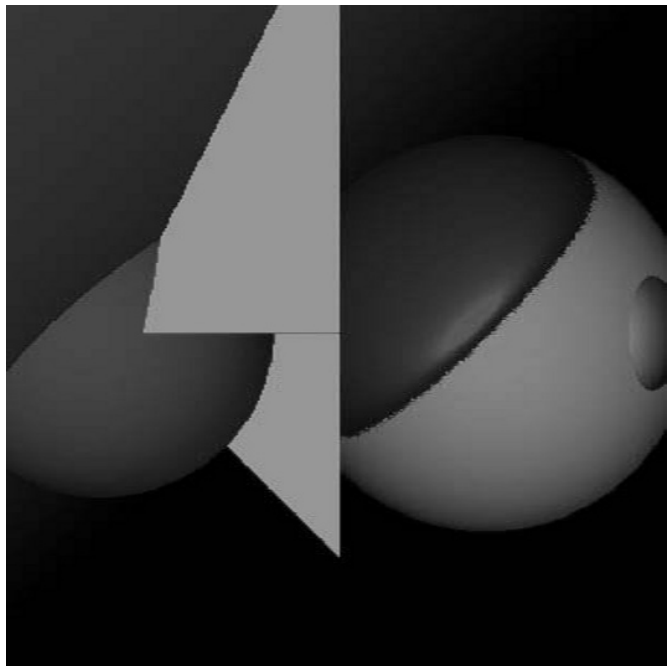


FIGURE 6. An example of the output from a simple OCGA ray-tracer of a scene containing infinite planes, spheres and triangular facets.

4.2. Catadioptric camera geometry

A number of recent papers (for example [1]) related to modeling of Catadioptric Cameras have made use of Stolfi's Oriented Projective Geometry to allow for the separation of rays intersecting the image plane back-face and front-face. In combination with recent CGA based representations of such cameras [2, 17], it

should be possible to simplify the development of algorithms and generalize the models used.

4.3. Physical simulations

The testing of containment is a crucial operation in many physical simulations (see for example [10]) and the clear framework provided by OCGA for containment tests for any object whose surface can be formed from spherical patches (including flat facets), should allow for rapid implementation of algorithms for collision detection and response. OCGA may also facilitate the development of a hybrid k-DOP[2] / sphere tree[3] based bounding volume hierarchy for rapid collision detection. The resulting combination of flat and curved size bounding volumes should reduce the number of elements needed to bound the object to a given accuracy.

5. Conclusions

Clearly, taking account of the sign of the multivectors in CGA provides us with a number of improvements in both understanding (such as the projection operators for splitting vectors), and in new methods (such as the tetrahedron inclusion tests). Hopefully this increased representative power will provide advantages in algorithm development within many areas of computer graphics and computer vision.

References

[1] J. Barreto and K. Daniilidis, *Unifying Image Plane Liftings for Central Catadioptric and Dioptric Cameras*. Proceedings of Omnivis, 2004.

[2] E. Bayro-Corrochano and C. López-Franco, *Omnidirectional Vision: Unified Model Using Conformal Geometry*. Proceedings of ECCV, 2004.

[3] G. Bradshaw and C. O'Sullivan, *Adaptive Medial-Axis Approximation for Sphere-Tree Construction*. ACM Transactions on Graphics, vol 23, (2004), 1–26.

[4] J. Cameron and J. Lasenby, *Oriented Conformal Geometric Algebra*. Cambridge University Engineering Department Technical Report (CUED/ F-INGENG/TR.541), 2005.

[5] C. Doran and A. Lasenby, *Geometric Algebra for Physicists*. CUP, 2003.

[6] C. Doran, *Circle and Sphere Blending with Conformal Geometric Algebra*. In submission to Journal of Geometric Design, 2004.

[7] D. Fontijne and L. Dorst, *Modeling 3D Euclidean Geometry. IEEE Computer Graphics and Applications*, vol 23, (2003), 68–78.

[8] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus*. D. Reidel, 1984.

[9] D. Hestenes and J. Holt, *The Crystallographic Space Groups in Geometric Algebra*. Journal of Mathematical Physics, vol 48 (2007), 023514.

[2]k-DOP (Discrete Oriented Polytope) trees are a bounding volume hierarchy based upon convex polytopes all of whose k facets have outward normals from a given discrete set of k vectors [11].
[3]Sphere trees are a bounding volume hierarchy based upon spheres [3].

[10] D. Kaufman, T. Edmunds and D. Pai, Fast Frictional Dynamics for Rigid Bodies. In Proceedings of SIGGRAPH, 2005.

[11] J. Klosowski, M. Held, J. Mitchel, H. Sowizral and K. Zikan, *Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs*. IEEE Transactions on Visualization and Computer Graphics, vol 4 (1998), 21–36.

[12] A. Lasenby, J. Lasenby and R. Wareham, *A Covariant Approach to Geometry Using Geometric Algebra*. Cambridge University Engineering Department Technical Report (CUED/F-INFENG/TR-483), 2004.

[13] A. Lasenby, *Keynote Address* SIGGRAPH. 2003.

[14] H. Li, D. Hestenes and A. Rockwood, *Generalized Homogeneous Coordinates for Computational Geometry*. In Geometric Computing with Clifford Algebra, Ed. G. Sommer, Springer-Verlag, 2001, 25–58.

[15] C. Perwass, V. Banarer and G. Sommer, *Spherical Decision Surfaces Using Conformal Modelling*. In Pattern Recognition: 25th DAGM Symposium, Ed. B. Michaelis and G. Krell, Springer 2003, 9–16.

[16] J. Stolfi, *Oriented Projective Geometry – A Framework for Geometric Computations*. Academic Press, 1991.

[17] R. Wareham, J. Cameron and J. Lasenby, *Applications of Geometric Algebra in Computer Vision and Graphics*. In Computer Algebra and Geometric Algebra with Applications, Eds. H. Li, P. Olver and G. Sommer, Springer-Verlag, 2005.

Jonathan Cameron and Joan Lasenby
Signal Processing and Communications Laboratory
Department of Engineering,
University of Cambridge
Cambridge, United Kingdom
e-mail: jic23@cam.ac.uk
jl@eng.cam.ac.uk

Received: November 15, 2005.
Accepted: July 8, 2006.