Geometric algebra: a computational framework for geometrical applications (part II: applications)

Leo Dorst* and Stephen Mann†

Abstract

Geometric algebra is a consistent computational framework in which to define geometric primitives and their relationships. This algebraic approach contains all geometric operators and permits coordinate-free specification of computational constructions. It contains primitives of any dimensionality (rather than just vectors). This second paper on the subject uses the basic products to represent rotations (naturally incorporating quaternions), intersections, and differentiation. It shows how using well-chosen geometric algebra models, we can eliminate special cases in incidence relationships, yet still have the efficiency of the Plücker coordinate intersection computations.

Keywords: geometric algebra, Clifford algebra, rotation reprensentation, quaternions, dualization, meet, join, Plücker coordinates, homogeneous coordinates, geometric differentiation, computational geometry

1 Introduction

This is the second of two papers introducing geometric algebra. In the previous paper, we introduced *blades*, a computational algebraic representation of *oriented subspaces*, which are the basic element of computation in geometric algebra. We also looked at the geometric product, and two products derived from the geometric product, the inner and outer products. A crucial feature of the geometric product is that it is invertible.

From that first paper, you should have gathered that every vector space with an inner product has a geometric algebra, whether you choose to use it or not. This paper shows how to call upon this structure for the definition of common geometrical constructs, ensuring a totally consistent computational framework. The goal is to show

*Informatics Institute, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

†Computer Science Department, University of Waterloo, Waterloo, Ontario, N2L 3G1, CANADA

you that this can be done, and that it is compact, directly computational, and transcends the dimensionality of subspaces. We will not use geometric algebra to develop new algorithms for graphics; but we hope to convince you that some of the lower level algorithmic aspects can be taken care of in an automatic way, without tricks, exceptions or hidden degenerate cases by using geometric algebra as a language.

In section 2 we show how geometric algebra represents rotations naturally by elements we recognize as quaternions. In Section 3 we indicate how linear algebra is enriched within geometric algebra by the capability to compute directly with subspaces in a coordinate-free manner. Section 4 briefly touches on the possibility of differentiating geometric objects. By using an appropriate model of a target geometry, we can extend the applicability of the geometric algebra approach; two such models for Euclidean geometry are introduced in Section 5. We give some remarks on implementation in Section 6, and we conclude the paper with suggestions for futher reading.

2 Rotations

Geometric algebra handles rotations of general subspaces in V^m through an interesting 'sandwiching product' using geometric products. We introduce this construction gradually in this section.

2.1 Rotations in 2D

In the last paper, we saw that the ratio of vectors defines a rotation/dilation operator. Let us do a slightly simpler problem: in Figure 1a, if $\mathbf{a}$ and $\mathbf{b}$ have the same norm, what is the vector $\mathbf{x}$ in the $(\mathbf{a},\mathbf{b})$-plane that is to $\mathbf{c}$ as the vector $\mathbf{b}$ is to $\mathbf{a}$? Geometric algebra phrases this as $\mathbf{x}\,\mathbf{c}^{-1} = \mathbf{b}\,\mathbf{a}^{-1}$, and solves it as (see eq.(13) from [5])

$$\mathbf{x} = (\mathbf{b}\,\mathbf{a}^{-1})\,\mathbf{c} = \frac{1}{|\mathbf{a}|^2}(\mathbf{b}\cdot\mathbf{a} + \mathbf{b}\wedge\mathbf{a})\,\mathbf{c} = \frac{|\mathbf{b}|}{|\mathbf{a}|}(\cos\phi - \mathbf{I}\sin\phi)\mathbf{c} = e^{-\mathbf{I}\phi}\mathbf{c} \qquad (1)$$

Here $\mathbf{I}\phi$ is the angle in the $\mathbf{I}$ plane from $\mathbf{a}$ to $\mathbf{b}$, so $-\mathbf{I}\phi$ is the angle from $\mathbf{b}$ to $\mathbf{a}$. Figure 1a suggests that $\mathbf{x}$ is obtained from $\mathbf{c}$ by a rotation, so we should apparently interpret 'pre-multiplying by $e^{-\mathbf{I}\phi}$' as a *rotation operator* in the $\mathbf{I}$-plane.

> Let us consider the geometrical meaning of the terms of this rotation operation when re-expressed into its components:
>
> $$e^{-\mathbf{I}\phi}\,\mathbf{c} = \mathbf{c}\cos\phi - \mathbf{I}\mathbf{c}\sin\phi = \mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi$$
>
> What is $\mathbf{c}\mathbf{I}$? Temporarily introduce orthonormal coordinates $\{\mathbf{e}_1,\mathbf{e}_2\}$ in the $\mathbf{I}$-plane, with $\mathbf{e}_1$ along $\mathbf{c}$, so that $\mathbf{c} \equiv c\,\mathbf{e}_1$. Then $\mathbf{I} = \mathbf{e}_1\wedge\mathbf{e}_2 = \mathbf{e}_1\,\mathbf{e}_2$. Therefore $\mathbf{c}\mathbf{I} = c\,\mathbf{e}_1\,\mathbf{e}_1\mathbf{e}_2 = c\,\mathbf{e}_2$: it is $\mathbf{c}$ *turned over a right angle*, following the orientation of the 2-blade $\mathbf{I}$ (here anti-clockwise). So $\mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi$ is 'a bit of $\mathbf{c}$ plus a bit of its anti-clockwise perpendicular' – and those amounts are precisely right to make it equal to the rotation by ϕ, see Figure 1b.

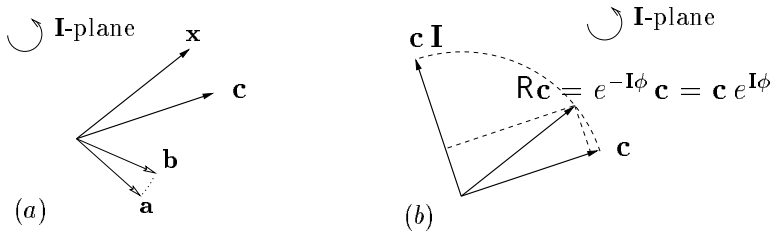

Figure 1: *(a) The rotation operator as a ratio between vectors; (b) Coordinate-free specification of rotation.*

The vector $\mathbf{c}$ in the $\mathbf{I}$-plane anti-commutes with $\mathbf{I}$: $\mathbf{c}\,\mathbf{I} = -\mathbf{I}\,\mathbf{c}$ (showing that $\mathbf{I}$ is not merely a complex number, even though $\mathbf{I}^2 = -1$). Using this to switch $\mathbf{I}$ and $\mathbf{c}$ in eq.(1), we obtain

$$\mathsf{R}_{\mathbf{I}\phi}\mathbf{c} = e^{-\mathbf{I}\phi}\mathbf{c} = \mathbf{c}\,e^{\mathbf{I}\phi}. \tag{2}$$

The planar rotation is therefore alternatively representable as a post-multiplication.

2.2 Angles as geometrical objects

In eq.(1), the combination $\mathbf{I}\phi$ is a full indication of the angle between the two vectors: it denotes not only the magnitude, but also the plane in which the angle is measured, and even the orientation of the angle. If you would ask for the scalar magnitude of the geometrical quantity $\mathbf{I}\phi$ in the plane $-\mathbf{I}$ (the plane 'from $\mathbf{b}$ to $\mathbf{a}$' rather than 'from $\mathbf{a}$ to $\mathbf{b}$'), it is $-\phi$; so the scalar value of the angle automatically gets the right sign. The fact that the angle as expressed by $\mathbf{I}\phi$ is now a geometrical quantity independent of the convention used in its definition removes a major headache from many geometrical computations involving angles. We call this true geometric quantity the *bivector angle* (it is just a 2-blade, of course, not a new kind of element – but we use it as an angle, hence the name).

2.3 Rotations in m dimensions

The above rotates only *within* a plane; in general we would like to have rotations in space. For a vector $\mathbf{x}$, the outcome of a rotation $\mathsf{R}_{\mathbf{I}\phi}$ should be:

$$\mathsf{R}_{\mathbf{I}\phi}\mathbf{x} = \mathbf{x}_\perp + \mathsf{R}_{\mathbf{I}\phi}\mathbf{x}_\parallel,$$

where $\mathbf{x}_\perp$ and $\mathbf{x}_\parallel$ are the perpendicular and parallel components of $\mathbf{x}$ relative to the rotation plane $\mathbf{I}$, respectively. We have seen that the separation of a vector into such components can be done by commutation (as in eq.(31) and eq.(32) from [5]). As you may verify, the following formula effects this separation and rotation simultaneously

$$\text{rotation over } \mathbf{I}\phi\text{:}\quad \mathbf{x} \;\mapsto\; \mathsf{R}_{\mathbf{I}\phi}\mathbf{x} = e^{-\mathbf{I}\phi/2}\,\mathbf{x}\,e^{\mathbf{I}\phi/2} \tag{3}$$

The operator $e^{-\mathbf{I}\phi/2}$, used in this way, is called a *rotor*. In the 2-dimensional rotation we treated before, $\mathbf{x}\,\mathbf{I} = -\mathbf{I}\,\mathbf{x}$, so moving either rotor 'to the other side of $\mathbf{x}$' retrieves eq.(2) if $\mathbf{x}$ is in the $\mathbf{I}$-plane.

Two successive rotations R_1 and R_2 are equivalent to a single new rotation R of which the rotor R is the geometric product of the rotors R_2 and R_1, since

$$(\mathsf{R}_2 \circ \mathsf{R}_1)\mathbf{x} = R_2\,(R_1\,\mathbf{x}\,R_1^{-1})\,R_2^{-1} = (R_2\,R_1)\,\mathbf{x}\,(R_2\,R_1)^{-1} = R\,\mathbf{x}R^{-1},$$

with $R = R_2\,R_1$. Therefore the combination of rotations is a simple consequence of the application of the geometric product on rotors, i.e. elements of the form $e^{-\mathbf{I}\phi/2} = \cos\phi/2 - \mathbf{I}\sin\phi/2$, with $\mathbf{I}^2 = -1$. This is true in any dimension greater than 1 (and even in dimension 1 if you realize that any bivector there is zero, so that rotations do not exist).

> Let's see how it works in 3-space. In 3 dimensions, we are used to specifying rotations by a *rotation axis* $\mathbf{a}$ rather than by a *rotation plane* $\mathbf{I}$. Given a unit vector $\mathbf{a}$ for an axis, we find the plane as the 2-blade complementary to it in the 3-dimensional space with volume element $\mathbf{I}_3$: $\mathbf{I} = \mathbf{a}\rfloor\mathbf{I}_3 = \mathbf{a}\mathbf{I}_3 = \mathbf{I}_3\mathbf{a}$. A rotation over an angle ϕ around an axis with unit vector $\mathbf{a}$ is therefore represented by the rotor $e^{-\mathbf{I}_3\mathbf{a}\phi/2}$.
>
> To compose a rotation R_1 around the $\mathbf{e}_1$ axis of $\pi/2$ with a subsequent rotation R_2 over the $\mathbf{e}_2$ axis over $\pi/2$, we write out their rotors (using the shorthand $\mathbf{e}_{23} = \mathbf{e}_2 \wedge \mathbf{e}_3 = \mathbf{e}_2\,\mathbf{e}_3$ etc.):
>
> $$R_1 = e^{-\mathbf{I}_3\mathbf{e}_1\pi/4} = \frac{1-\mathbf{e}_{23}}{\sqrt{2}} \quad \text{and} \quad R_2 = e^{-\mathbf{I}_3\mathbf{e}_2\pi/4} = \frac{1-\mathbf{e}_{31}}{\sqrt{2}}$$
>
> The total rotor is their product, and we rewrite it back to the exponential form to find the axis:
>
> $$\begin{aligned} R \equiv R_2\,R_1 &= \tfrac{1}{2}(1-\mathbf{e}_{23})\,(1-\mathbf{e}_{31}) = \tfrac{1}{2}(1-\mathbf{e}_{23}-\mathbf{e}_{31}-\mathbf{e}_{12}) \\ &= \tfrac{1}{2} - \tfrac{1}{2}\sqrt{3}\,\mathbf{I}_3\,\frac{\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3}{\sqrt{3}} \equiv e^{-\mathbf{I}_3\mathbf{a}\pi/3} \end{aligned}$$
>
> Therefore the total rotation is over the axis $\mathbf{a} = (\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3)/\sqrt{3}$, over the angle $2\pi/3$.

Geometric algebra permits straightforward generalization to the rotation of higher dimensional subspaces: a rotor can be applied immediately to an arbitrary blade through the formula

$$\text{general rotation:} \quad \mathbf{X} \;\mapsto\; R\,\mathbf{X}\,R^{-1}$$

This enables you to rotate a plane in one operation, for instance, using a rotation by R as in the example above,

$$R(\mathbf{e}_1 \wedge \mathbf{e}_2)R^{-1} = \tfrac{1}{4}(1-\mathbf{e}_{23}-\mathbf{e}_{31}-\mathbf{e}_{12})\,\mathbf{e}_{12}\,(1+\mathbf{e}_{23}+\mathbf{e}_{31}+\mathbf{e}_{12}) = \mathbf{e}_{23}$$

There is no need to decompose the plane into its spanning vectors first.

2.4 Quaternions: based on bivectors

You may have recognized the example above as strongly similar to quaternion computations. Quaternions are indeed part of geometric algebra, in the following straightforward manner.

Choose an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^3$. Construct out of that a bivector basis $\{\mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}\}$. Note that these elements satisfy: $\mathbf{e}_{12}^2 = \mathbf{e}_{23}^2 = \mathbf{e}_{31}^2 = -1$, and $\mathbf{e}_{12}\,\mathbf{e}_{23} = \mathbf{e}_{13}$ (and cyclic) and also $\mathbf{e}_{12}\,\mathbf{e}_{23}\,\mathbf{e}_{31} = 1$. In fact, setting $i \equiv \mathbf{e}_{23}$, $j \equiv -\mathbf{e}_{31}$ and $k \equiv \mathbf{e}_{12}$, we find $i^2 = j^2 = k^2 = i\,j\,k = -1$ and $j\,i = k$ and cyclic. Algebraically these objects form a basis for quaternions obeying the quaternion product, commonly interpreted as some kind of '4-D complex number system'. There is nothing 'complex' about quaternions; but they are not really vectors either – they are just real 2-blades in 3-space, denoting elementary rotation planes, and multiplying through the geometric product. Visualizing quaternions is therefore straightforward: each is just a rotation plane with a rotation angle, and the 'bivector angle' concept of Section 2.2 represents that well.

Of course in geometric algebra, quaternions can be combined directly with vectors and other subspaces. In that algebraic combination, they are not merely a form of 'complex scalars': quaternion products are neither fully commutative nor fully anti-commutative (e.g. $i\,\mathbf{e}_1 = \mathbf{e}_1\,i$, but $i\,\mathbf{e}_2 = -\mathbf{e}_2\,i$). It all depends on the relative attitude of the vectors and quaternions, and these rules are precisely right to make eq.(3) be a rotation of a vector.

3 Linear algebra

In the classical ways of using vector spaces, linear algebra is an important tool. In geometric algebra, this remains true: linear transformations are of interest in their own right, or as first order approximations to more complicated mappings. Indeed, linear algebra is an integral part of geometric algebra, and acquires much extended coordinate-free methods through this inclusion. We show some of the basic principles; much more may be found in [3] or [11].

3.1 Outermorphisms: spanning is linear

When vectors are transformed by a linear transformation on the vector space, the blades they span can be viewed to transform as well, simply by the rule: 'the transform of a span of vectors is the span of the transformed vectors'. This means that a linear transformation $\mathsf{f} : V^m \to V^m$ of a vector space has a natural extension to the whole geometric algebra of that vector space, as an *outermorphism*, i.e. a mapping that preserves the outer product structure:

$$\mathsf{f}(\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \cdots \wedge \mathbf{a}_k) \equiv \mathsf{f}(\mathbf{a}_1) \wedge \mathsf{f}(\mathbf{a}_2) \wedge \cdots \wedge \mathsf{f}(\mathbf{a}_k). \tag{4}$$

Note that this is grade-preserving: a k-blade transforms to a k-blade. We supplement this by stating what the extension does to scalars, which is simply $\mathsf{f}(\alpha) = \alpha$. Geometrically, this means that a linear transformation 'leaves the origin intact.'

The fact that linear transformations are outermorphisms explains why we can generalize so many operations from vectors to general subspaces in a straightforward manner.

3.2 Dual representation

Dual representations of hyperplanes are often used in linear algebra to extend the scope of vector-based operations. In geometric algebra, dualization is also important, and in fact a very flexible tool to convert between viewpoints of spanning and perpendicularity, for arbitrary subspaces.

Consider an m-dimensional space V^m and a blade $\mathbf{A}$ in it. The *dual* of the blade $\mathbf{A}$ in V^m is its complement, definable using the contraction inner product:

$$\mathbf{A}^* = \mathbf{A} \rfloor \widetilde{\mathbf{I}}_m,$$

where $\mathbf{I}_m$ is the pseudoscalar of V^m (an m-blade giving the volume element) and $\widetilde{\cdot}$ is the *reversion* (obtained by reversing the factors of the blade it is applied to, and leading to a mere sign change).

The characterization of a subspace by a dual blade rather than by a blade enables manipulation of expressions involving 'spanning' to being about 'perpendicularity' and vice versa.

> A familiar example in a 3-dimensional Euclidean space is the dual of a 2-blade (or bivector). Using an orthonormal basis $\{\mathbf{e}_i\}_{i=1}^3$ and the corresponding bivector basis, we write: $\mathbf{B} = b_1\mathbf{e}_2 \wedge \mathbf{e}_3 + b_2\mathbf{e}_3 \wedge \mathbf{e}_1 + b_3\mathbf{e}_3 \wedge \mathbf{e}_2$. We take the dual relative to the space with volume element $\mathbf{I}_3 \equiv \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ (i.e. the 'right-handed volume' formed by using a right-handed basis). The subspace of $\mathbf{I}_3$ dual to $\mathbf{B}$ is then
>
> $$\begin{aligned} \mathbf{B} \rfloor \widetilde{\mathbf{I}}_3 &= (b_1\mathbf{e}_2 \wedge \mathbf{e}_3 + b_2\mathbf{e}_3 \wedge \mathbf{e}_1 + b_3\mathbf{e}_1 \wedge \mathbf{e}_2) \rfloor (\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1) \\ &= b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3. \end{aligned} \tag{5}$$
>
> This is a vector, and we recognize it (in this Euclidean space) as the *normal vector* to the planar subspace represented by $\mathbf{B}$, see Figure 2b. So we have normal vectors in geometric algebra as the duals of 2-blades, if we would want them – but we'll soon see a better alternative.

We can use either a blade or its dual to represent a subspace, and it is convenient to have some terminology. We will say that *a blade* $\mathbf{B}$ *represents a subspace* $\mathcal{B}$ if

$$\mathbf{x} \in \mathcal{B} \iff \mathbf{x} \wedge \mathbf{B} = 0 \tag{6}$$

and that *a blade* $\mathbf{B}^*$ *dually represents the subspace* $\mathcal{B}$ if

$$\mathbf{x} \in \mathcal{B} \iff \mathbf{x} \rfloor \mathbf{B}^* = 0. \tag{7}$$

Switching between the two standpoints is done by the distributive relation $(A \wedge B) \rfloor C = A \rfloor (B \rfloor C)$ (which is eq.(19) from [5]), used for a vector $\mathbf{x}$, a blade $\mathbf{B}$ and a pseudoscalar $\mathbf{I}$:

$$(\mathbf{x} \wedge \mathbf{B}) \rfloor \widetilde{\mathbf{I}} = \mathbf{x} \rfloor (\mathbf{B} \rfloor \widetilde{\mathbf{I}}), \tag{8}$$

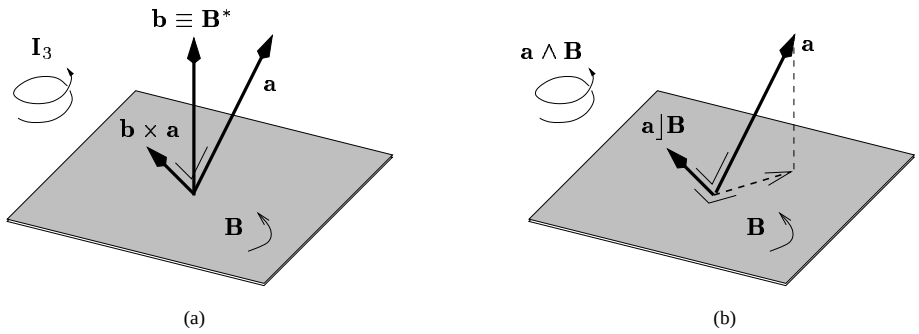

Figure 2: *(a) The dual bB^* of a bivector $\mathbf{B}$ and the cross product with $\mathbf{a}$; (b) The same result using the inner product of blades (from [5]).*

and by a converse (but conditional) relationship which we state without proof

$$(\mathbf{x} \rfloor \mathbf{B}) \rfloor \widetilde{\mathbf{I}} = \mathbf{x} \wedge (\mathbf{B} \rfloor \widetilde{\mathbf{I}}) \quad \text{if } \mathbf{x} \wedge \mathbf{I} = 0. \tag{9}$$

If $\mathbf{x}$ is known to be in the subspace of $\mathbf{I}$, we can write these simply as $(\mathbf{x} \wedge \mathbf{B})^* = \mathbf{x} \rfloor \mathbf{B}^*$ and $(\mathbf{x} \rfloor \mathbf{B})^* = \mathbf{x} \wedge \mathbf{B}^*$, which makes the equivalence of the two representations above obvious.

3.3 The cross product

Classical computations with vectors in 3-space often use the cross product, which produces from two vectors $\mathbf{a}$ and $\mathbf{b}$ a new vector $\mathbf{a} \times \mathbf{b}$ perpendicular to both (by the right-hand rule), proportional to the area they span. We can make this in geometric algebra as the dual of the 2-blade spanned by the vectors, see Figure 2b:

$$\mathbf{a} \times \mathbf{b} \equiv (\mathbf{a} \wedge \mathbf{b}) \rfloor \widetilde{\mathbf{I}}_3 \quad \left(= (\mathbf{a} \wedge \mathbf{b})^*\right). \tag{10}$$

You may verify that computing this explicitly using eq.(1) from [5] and eq.(5) indeed retrieves the usual expression $\mathbf{a} \times \mathbf{b} = (a_2 b_3 - a_3 b_2)\, \mathbf{e}_1 + (a_3 b_1 - a_1 b_3)\, \mathbf{e}_2 + (a_1 b_2 - a_2 b_1)\, \mathbf{e}_3$.

Eq.(10) shows a number of things explicitly that one always needs to remember about the cross product: there is a convention involved on handedness (this is coded in the sign of $\mathbf{I}_3$); there are metric aspects since it is perpendicular to a plane (this is coded in the usage of the inner product ' $\rfloor$ '); and the construction really only works in three dimensions, since only then is the dual of a 2-blade a vector (this is coded in the 3-gradedness of $\mathbf{I}_3$). The vector relationship $\mathbf{a} \wedge \mathbf{x}$ does not depend on any of these embedding properties, yet characterizes the $(\mathbf{a}, \mathbf{x})$-plane just as well. In geometric algebra, we therefore have the possibility of replacing the cross product by a more elementary construction. Linear algebra gives a good reason for doing so.

3.4 No normal vectors or cross products!

The transformation of an inner product under a linear mapping is more involved than that of the outer product in eq.(4), for 'perpendicularity' is a more complicated concept to transform than 'spanning'. The general transformation formula is given in [11], for blades it becomes

$$\mathsf{f}(\mathbf{A} \rfloor \mathbf{B}) = \overline{\mathsf{f}}^{\,-1}(\mathbf{A}) \rfloor \mathsf{f}(\mathbf{B}),$$

where $\overline{\mathsf{f}}$ is the *adjoint*, defined as the extension of an outermorphism of the linear mapping defined for vectors by $\overline{\mathsf{f}}(\mathbf{a}) \cdot \mathbf{b} = \mathbf{a} \cdot \mathsf{f}(\mathbf{b})$ (its matrix representation on vectors would be the *transpose*).

Because of the complexity of this transformation behavior, one should steer clear of any constructions that involve the inner product, especially in the characterization of basic properties of one's objects. The practice of characterizing a plane by its normal vector – which contains the inner product in its duality – should be avoided. Under linear transformations, *the normal vector of a transformed plane is not the transform of the normal vector of the plane!* (this is a well known fact, but always a shock to novices). The normal vector is in fact a cross product of vectors, which transforms as

$$\mathsf{f}(\mathbf{a} \times \mathbf{b}) = \overline{\mathsf{f}}^{\,-1}(\mathbf{a}) \times \overline{\mathsf{f}}^{\,-1}(\mathbf{b}) / \det(\mathsf{f}) \tag{11}$$

where $\det(\mathsf{f})$ is the determinant defined by $\det(\mathsf{f}) = \mathsf{f}(\mathbf{I}_m)\mathbf{I}_m^{-1}$ (this is a coordinate-free definition of the determinant of the matrix representation of f). The right hand side of eq.(11) is usually not equal to $\mathsf{f}(\mathbf{a}) \times \mathsf{f}(\mathbf{b})$, so a linear transformation is not an 'innermorphism'. It is therefore much better to characterize the plane by a 2-blade, now that we can. *The 2-blade of the transformed plane is the transform of the 2-blade of the plane*, since linear transformations are outermorphisms preserving the 2-blade construction. Especially when the planes are tangent planes constructed by differentiation, 2-blades are appropriate: under *any* transformation f, the construction of the tangent plane depends only on the first order linear approximation mapping f of f. Using blades for those tangent spaces should enormously simplify the treatment of objects through differential geometry, especially in the context of affine transformations.

3.5 Intersecting subspaces

Geometric algebra also contains operations to determine the *union* and *intersection* of subspaces. These are the join and meet operations. Several notations exist for these in the literature, causing some confusion. For this paper, we will simply use the set notations $\cup$ and $\cap$ to make the formulas more easily readable.

The join of two subspaces is their smallest superspace, i.e. the smallest space containing them both. Representing the spaces by blades $\mathbf{A}$ and $\mathbf{B}$, the join is denoted $\mathbf{A} \cup \mathbf{B}$. If the subspaces of $\mathbf{A}$ and $\mathbf{B}$ are disjoint, their join is obviously proportional to $\mathbf{A} \wedge \mathbf{B}$. But a problem is that if $\mathbf{A}$ and $\mathbf{B}$ are not disjoint (which is precisely the case we are interested in), then $\mathbf{A} \cup \mathbf{B}$ contains an unknown scaling factor that is fundamentally unresolvable due to the reshapable nature of the blades (discussed in Section 2.2 from [5]), see Figure 3; this ambiguity was also observed in [13]. Fortunately, in all geometrically relevant entities that we compute, it appears that this scalar ambiguity cancels.

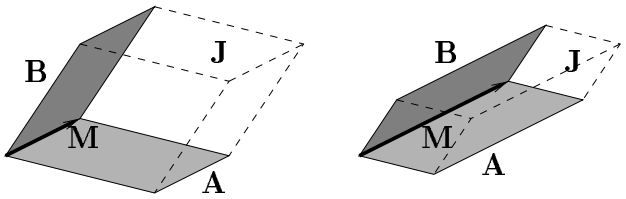

Figure 3: *The ambiguity of scale for* meet $\mathbf{M}$ *and* join $\mathbf{J}$ *of two blades* $\mathbf{A}$ *and* $\mathbf{B}$. *Both figures are acceptable solutions to the problem of finding a blade representing the union and intersection of the subspaces of the blades* $\mathbf{A}$ *and* $\mathbf{B}$.

The join is a more complicated product of subspaces than the outer product and inner product; we can give no simple formula for the grade of the result, and it cannot be characterized by a list of algebraic computation rules. Although computation of the join may appear to require some optimization process, the smallest superspace is directly related to the non-zero part of the highest grade in the expansion of the geometric product, and can therefore be done in constant time [1].

The meet of two subspaces $\mathbf{A}$ and $\mathbf{B}$ is their largest common subspace. Given the join $\mathbf{J} \equiv \mathbf{A} \cup \mathbf{B}$ of $\mathbf{A}$ and $\mathbf{B}$, we can compute their meet $\mathbf{A} \cap \mathbf{B}$ by the property that its dual (with respect to the join) is the outer product of their duals (this is a not-so-obvious consequence of the required 'containment in both'). In formula, this is

$$(\mathbf{A} \cap \mathbf{B})\rfloor\widetilde{\mathbf{J}} = (\mathbf{B}\rfloor\widetilde{\mathbf{J}}) \wedge (\mathbf{A}\rfloor\widetilde{\mathbf{J}}) \text{ or } (\mathbf{A} \cap \mathbf{B})^* = \mathbf{B}^* \wedge \mathbf{A}^* \tag{12}$$

with the dual taken with respect to the join $\mathbf{J}$.[1] This leads to a formula for the meet of $\mathbf{A}$ and $\mathbf{B}$ relative to the chosen join (use eq.(8)):

$$\mathbf{A} \cap \mathbf{B} = (\mathbf{B}\rfloor\widetilde{\mathbf{J}})\rfloor\mathbf{A}. \tag{13}$$

Let us do an example.

> The intersection of two planes represented by the 2-blades $\mathbf{A} = \frac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2) \wedge (\mathbf{e}_2 + \mathbf{e}_3)$ and $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$. (We have normalized them so that $\mathbf{A}\rfloor\widetilde{\mathbf{A}} = 1 = \mathbf{B}\rfloor\widetilde{\mathbf{B}}$; it gives the meet of these blades a numerical factor that can be interpreted geometrically.) These are planes in general position in 3-dimensional space, so their join is proportional to $\mathbf{I}_3$. It makes sense to take $\mathbf{J} = \mathbf{I}_3$. This gives for the meet:
>
> $$\begin{aligned} \mathbf{A} \cap \mathbf{B} &= \tfrac{1}{2}\left((\mathbf{e}_1 \wedge \mathbf{e}_2)\rfloor(\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1)\right)\rfloor\left((\mathbf{e}_1 + \mathbf{e}_2) \wedge (\mathbf{e}_2 + \mathbf{e}_3)\right) \\ &= \tfrac{1}{2}\,\mathbf{e}_3\rfloor\left((\mathbf{e}_1 + \mathbf{e}_2) \wedge \mathbf{e}_3\right) \\ &= -\tfrac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2) = -\tfrac{1}{\sqrt{2}}\left(\frac{\mathbf{e}_1 + \mathbf{e}_2}{\sqrt{2}}\right) \end{aligned} \tag{14}$$

[1] The somewhat strange order in eq.(12) means that the join $\mathbf{J}$ can be written using the meet $\mathbf{M}$ in the factorization $\mathbf{J} = (\mathbf{A}\mathbf{M}^{-1}) \wedge \mathbf{M} \wedge (\mathbf{M}^{-1}\mathbf{B})$, and it corresponds to [13] for vectors.

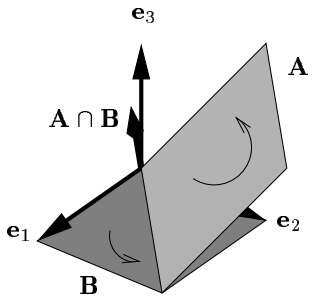

Figure 4: *An example of the* meet

> We have expressed the result in normalized form; the numerical factor for the resulting blade is in fact the sine of the angle between the arguments. Figure 4 illustrates the answer; as in [13] the sign of $\mathbf{A} \cap \mathbf{B}$ is the right-hand rule applied to the turn required to make $\mathbf{A}$ coincide with $\mathbf{B}$, in the correct orientation.

Classically, one computes the intersection of two planes in 3-space by first converting them to normal vectors, and then taking the cross product. We can see that this gives the same answer in this non-degenerate case in 3-space, using our previous equations eq.(9), eq.(8), and noting that $\widetilde{\mathbf{I}}_3 = -\mathbf{I}_3$:

$$\begin{aligned}(\mathbf{A}\rfloor\widetilde{\mathbf{I}}_3) \times (\mathbf{B}\rfloor\widetilde{\mathbf{I}}_3) &= \Big((\mathbf{A}\rfloor\widetilde{\mathbf{I}}_3) \wedge (\mathbf{B}\rfloor\widetilde{\mathbf{I}}_3)\Big)\rfloor\widetilde{\mathbf{I}}_3 = \Big((\mathbf{B}\rfloor\widetilde{\mathbf{I}}_3) \wedge (\mathbf{A}\rfloor\widetilde{\mathbf{I}}_3)\Big)\rfloor\mathbf{I}_3 \\ &= (\mathbf{B}\rfloor\widetilde{\mathbf{I}}_3)\rfloor\Big((\mathbf{A}\rfloor\widetilde{\mathbf{I}}_3)\rfloor\mathbf{I}_3\Big) = (\mathbf{B}\rfloor\widetilde{\mathbf{I}}_3)\rfloor\mathbf{A} = \mathbf{A} \cap \mathbf{B}.\end{aligned}$$

So the classical result is a special case of eq.(13), but eq.(13) is much more general: it applies to the intersection of subspaces of *any* grade, within a space of *any* dimension. Again we see the power of geometric algebra in compact expressions valid for any grade or dimension.

The norm of the meet gives an impression of the 'strength' of the intersection. Between normalized subspaces in Euclidean space, the magnitude of the meet is the sine of the angle between them. From numerical analysis, this is a well-known measure for the 'distance' between subspaces in terms of their orthogonality: it is 1 if the spaces are orthogonal, and decays gracefully to 0 as the spaces get more parallel, before changing sign. This numerical significance is very useful in applications.

4 Differentiation

Geometric algebra has an extended operation of differentiation, which contains the classical vector calculus, and much more. It is possible to differentiate with respect

to a scalar or a vector, as before, but now also with respect to k-blades. This enables efficient encoding of differential geometry, in a coordinate-free manner, and gives an alternative look at differential shape descriptors like the 'second fundamental form' (it becomes an immediate indication of how the tangent plane changes when we slide along the surface). This would lead us too far, but we will show two examples of differentiation.

4.1 Scalar differentiation of a rotor

Suppose we have a rotor $R = e^{-\mathbf{I}\phi/2}$ (where $\mathbf{I}\phi$ is a function of time t), and use it to produce a rotated version $\mathbf{X} = R\,\mathbf{X}_0\,R^{-1}$ of some constant blade $\mathbf{X}_0$. Scalar differentiation with respect to t gives (using the chain rule and commutation rules)

$$\begin{aligned}
\tfrac{d}{dt}\mathbf{X} &= \tfrac{d}{dt}(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2}) \\
&= -\tfrac{1}{2}\tfrac{d}{dt}(\mathbf{I}\phi)(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2}) + \tfrac{1}{2}(e^{-\mathbf{I}\phi/2}\mathbf{X}_0 e^{\mathbf{I}\phi/2})\tfrac{d}{dt}(\mathbf{I}\phi) \\
&= \tfrac{1}{2}(\mathbf{X}\,\tfrac{d}{dt}(\mathbf{I}\phi) - \tfrac{d}{dt}(\mathbf{I}\phi)\,\mathbf{X}) \\
&= \mathbf{X} \times \tfrac{d}{dt}(\mathbf{I}\phi) \qquad (15)
\end{aligned}$$

using the commutator product $\times$ defined in geometric algebra as the shorthand $A \times B \equiv \frac{1}{2}(AB - BA)$; this product often crops up in computations with Lie groups such as the rotations.

The simple expression that results assumes a more familiar form when $\mathbf{X}$ is a vector $\mathbf{x}$ in 3-space, the attitude of the rotation plane is fixed so that $\frac{d}{dt}\mathbf{I} = 0$, and we introduce a scalar angular velocity $\omega \equiv \frac{d}{dt}\phi$. It is then common practice to introduce the vector dual to the plane as the angular velocity vector $\boldsymbol{\omega}$, so $\boldsymbol{\omega} \equiv \omega\mathbf{I}\rfloor\widetilde{\mathbf{I}}_3 = \omega\mathbf{I}/\mathbf{I}_3$. Therefore $\omega\mathbf{I} = \boldsymbol{\omega}\mathbf{I}_3$, which equals $\boldsymbol{\omega}\rfloor\mathbf{I}_3$. Using the fact that $\mathbf{x} \times \mathbf{B} = \frac{1}{2}(\mathbf{x}\,\mathbf{B} - \mathbf{B}\,\mathbf{x}) = \mathbf{x}\rfloor\mathbf{B}$ for a vector $\mathbf{x}$ and a 2-blade $\mathbf{B}$, we obtain

$$\tfrac{d}{dt}\mathbf{x} = \mathbf{x} \times \tfrac{d}{dt}(\mathbf{I}\phi) = \mathbf{x} \times (\boldsymbol{\omega}\rfloor\mathbf{I}_3) = \mathbf{x}\rfloor(\boldsymbol{\omega}\rfloor\mathbf{I}_3) = (\mathbf{x} \wedge \boldsymbol{\omega})\rfloor\mathbf{I}_3 = \boldsymbol{\omega} \times \mathbf{x}$$

where $\times$ is the vector cross product. As before when we treated other operations, we find that an equally simple geometric algebra expression is much more general; here eq.(15) describes the differential rotation of k-dimensional subspaces in n-dimensional space, rather than merely of vectors in 3-D.

4.2 Vector differentiation of spherical projection

Suppose that we project a vector $\mathbf{x}$ on the unit sphere by the function $\mathbf{x} \mapsto \mathsf{P}(\mathbf{x}) = \mathbf{x}/|\mathbf{x}|$. We compute its derivative in the $\mathbf{a}$ direction, denoted as $(\mathbf{a}\cdot\partial_{\mathbf{x}})\mathsf{P}(\mathbf{x})$ or $\mathsf{P}_{\mathbf{a}}(\mathbf{x})$, as a standard differential quotient and using Taylor series expansion. Note how geometric algebra permits compact expression of the result, with geometrical significance:

$$\begin{aligned}
(\mathbf{a}\cdot\partial_{\mathbf{x}})\frac{\mathbf{x}}{|\mathbf{x}|} &\equiv \lim_{\lambda\to 0}\frac{1}{\lambda}\left(\frac{\mathbf{x}+\lambda\mathbf{a}}{|\mathbf{x}+\lambda\mathbf{a}|} - \frac{\mathbf{x}}{|\mathbf{x}|}\right) = \lim_{\lambda\to 0}\frac{1}{\lambda}\left(\frac{\mathbf{x}+\lambda\mathbf{a}}{|\mathbf{x}|\sqrt{1+2\lambda\mathbf{a}\cdot\mathbf{x}^{-1}}} - \frac{\mathbf{x}}{|\mathbf{x}|}\right) \\
&= \lim_{\lambda\to 0}\frac{(\mathbf{x}+\lambda\mathbf{a})(1-\lambda\mathbf{a}\cdot\mathbf{x}^{-1}) - \mathbf{x}}{\lambda|\mathbf{x}|} = \frac{\mathbf{a} - \mathbf{x}(\mathbf{a}\cdot\mathbf{x}^{-1})}{|\mathbf{x}|}
\end{aligned}$$

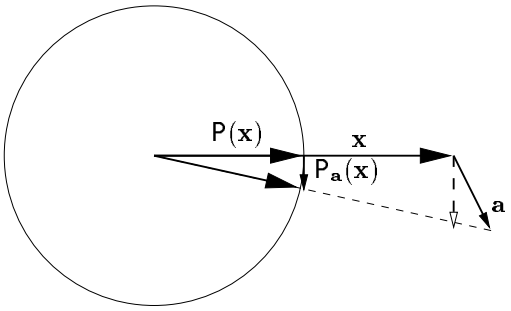

Figure 5: *The derivative of the spherical projection.*

$$= \frac{(\mathbf{a} \wedge \mathbf{x})\,\mathbf{x}^{-1}}{|\mathbf{x}|}$$

We recognize the result as the rejection of $\mathbf{a}$ by $\mathbf{x}$ (Section 4.1 from [5]), scaled appropriately. The sketch of Figure 5 confirms the outcome. You may verify in a similar manner that $(\mathbf{a} \cdot \partial_{\mathbf{x}})\mathbf{x}^{-1} = -\mathbf{x}^{-1}\,\mathbf{a}\,\mathbf{x}^{-1}$, and interpret geometrically.

For more advanced usage of differentiation relative to blades, the interested reader is referred to the tutorial of [3], which introduces these differentiations using examples from physics, and the application paper [12].

5 Models of geometry

Several standard models of geometry can be expressed in geometric algebra. The advantage of doing so is an increase in expressive power, and a structural integration of seemingly *ad hoc* constructions into the geometry. Here we look at geometric algebra representations of homogeneous coordinates and Plücker coordinates, and a model of Euclidean geometry that naturally handles spheres.

5.1 The homogeneous model

So far we have been treating only homogeneous subspaces of the vector space V^m, i.e. subspaces containing the origin. We have spanned them, projected them, and rotated them, but we have not moved them out of the origin to make more interesting geometrical structures such as lines floating in space. We construct those now, by extending the thoughts behind 'homogeneous coordinates' to geometric algebra. It turns out that these elements of geometry can also be represented by blades, in a representational space with an extra dimension. The geometric algebra of this space then turns out to give us precisely what we need. In this view, more complicated geometrical objects do not require new operations or techniques in geometric algebra, merely the standard computations in a higher dimensional space.

The *homogeneous model* in geometric algebra extends the usual 'homogeneous coordinates' model from linear algebra, which works merely on vectors. That model is

often described as augmenting a 3-dimensional vector $\mathbf{v}$ with coordinates $[v_1, v_2, v_3]$ to a 4-vector $[v_1, v_2, v_3, 1]$. That extension makes non-linear operations such as translations implementable as linear mappings.

Following the approach of this paper, we have to give the $(m+1)$-dimensional 'homogeneous space' into which we embed our m-dimensional Euclidean space a full geometric algebra. Let the unit vector for the extra dimension be denoted by e. This vector must be perpendicular to all regular vectors in the Euclidean space E^m, so $e \cdot \mathbf{x} = 0$ for all $\mathbf{x} \in E^m$. We also need to define $e \cdot e$ to make our algebra complete. This involves some dilemmas that are only fully resolvable in the double homogeneous model of Section 5.3. For now we can take $e \cdot e = 1$. We let e denote 'the Euclidean point at the origin'.

Let us now represent the subspaces of interest into this model, simply by using the structure of its geometric algebra.

- *Points:* A point at a location $\mathbf{p}$ is made by translation of the point at the origin over the Euclidean vector $\mathbf{p}$. This is done by adding $\mathbf{p}$ to e. This construction therefore gives the representation of the point $\mathcal{P}$ at location $\mathbf{p}$ as the vector p in $(m+1)$-dimensional space:

  $$p = e + \mathbf{p}$$

  This is just a regular vector in the $(m+1)$-space, now interpretable as a Euclidean point. It is of course no more than the usual 'homogeneous coordinates' method in disguise: p has coordinates $[p_1, p_2, p_3, 1]$ on the orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, e\}$. We will denote points of the m-dimensional Euclidean space in script, the vectors and blades in the corresponding vector space in **bold**, and vectors and blades in the $(m+1)$-dimensional homogeneous space in *italic*. You can visualize this construction as in Figure 6a (necessarily drawn for $m = 2$).

  These vectors in $(m+1)$-dimensional space can be multiplied using the products in geometric algebra. Let us consider in particular the outer product, and form blades.

- *Lines:* To represent a line, we compute the 2-blade spanned by the representative vectors of two points:

  $$p \wedge q = (e + \mathbf{p}) \wedge (e + \mathbf{q}) = e \wedge (\mathbf{q} - \mathbf{p}) + \mathbf{p} \wedge \mathbf{q} \tag{16}$$

  We recognize the vector $\mathbf{q} - \mathbf{p}$, and the area spanned by $\mathbf{p}$ and $\mathbf{q}$. Both are elements that we need to describe an element of the directed line through the points $\mathcal{P}$ and $\mathcal{Q}$. The former is the *direction vector* of the directed line, the latter is an area that we will call the *moment* of the line through p and q. It specifies the distance to the origin, for we can rewrite it to a rectangle spanned by the direction $(\mathbf{q} - \mathbf{p})$ and the *perpendicular support vector* $\mathbf{d}$:

  $$\mathbf{p} \wedge \mathbf{q} = \mathbf{p} \wedge (\mathbf{q} - \mathbf{p}) = \mathbf{d}\,(\mathbf{q} - \mathbf{p}) \tag{17}$$

  where $\mathbf{d} \equiv (\mathbf{p} \wedge (\mathbf{q} - \mathbf{p}))(\mathbf{q} - \mathbf{p})^{-1} = (\mathbf{p} \wedge \mathbf{q})(\mathbf{q} - \mathbf{p})^{-1}$ is the rejection of $\mathbf{p}$ by $\mathbf{q} - \mathbf{p}$. We can therefore rewrite the same 2-blade $p \wedge q$ in various ways, such as $p \wedge q = p \wedge (\mathbf{q} - \mathbf{p}) = d\,(\mathbf{q} - \mathbf{p})$ (with $de + \mathbf{d}$), separating the positional part

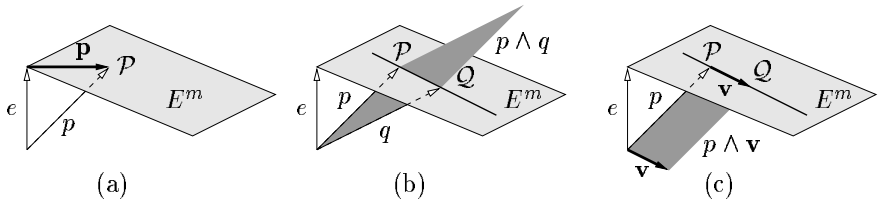

Figure 6: *Representing offset subspaces of E^m in $(m+1)$-dimensional space. In (c),* $\mathbf{v} \equiv \mathbf{q} - \mathbf{p}$.

p or d and the purely Euclidean directional part $\mathbf{v} \equiv \mathbf{q} - \mathbf{p}$, see Figure 6c. The element $p \wedge q$ contains all these potential interpretations in one data structure, which may be constructed in all of these ways.

Temporarily reverting to the cross product to make a connection with a classical representation, we can rewrite eq.(16) as

$$p \wedge q = e \wedge (\mathbf{q} - \mathbf{p}) + \mathbf{p} \wedge \mathbf{q} = (\mathbf{p} - \mathbf{q})\, e + (\mathbf{p} \times \mathbf{q})\, \mathbf{I}_3 \tag{18}$$

We recognize the six Plücker coefficients $[\mathbf{p} - \mathbf{q}; \mathbf{p} \times \mathbf{q}]$ characterizing the line by its direction vector $\mathbf{v} = \mathbf{q} - \mathbf{p}$ and its 'moment' vector $\mathbf{m} = \mathbf{p} \times \mathbf{q}$ as

$$\ell = -\mathbf{v}e + \mathbf{m}\mathbf{I}_3$$

The six coefficients $[-v_1, -v_2, -v_3, m_1, m_2, m_3]$ of the line in this representation are in fact the coefficients of a 2-blade on the bivector basis $\{\mathbf{e}_1 e, \mathbf{e}_2 e, \mathbf{e}_3 e, \mathbf{e}_2\mathbf{e}_3, \mathbf{e}_3\mathbf{e}_1, \mathbf{e}_1\mathbf{e}_2\}$. This integrates the Plücker representation fully into the homogeneous model (for which it was indeed historically designed). We will see that the compact and efficient Plücker intersection formulas are now straightforward consequences of the meet operation in geometric algebra.

- *Hyperplanes:* If we have an $(m-1)$-dimensional hyperplane characterized as $\mathbf{x} \cdot \mathbf{n} = \delta$, this can be written as $(e + \mathbf{x}) \cdot (\mathbf{n} - \delta e) = 0$, so as $x \cdot (\mathbf{n} - \delta e) = 0$. Therefore $\mathbf{n} - \delta e$ is the dual of the blade representing the hyperplane. The dual A^* of a blade A in the homogeneous model is obtained relative to the pseudoscalar $e \wedge \mathbf{I}_3 = e\,\mathbf{I}_3$ of the full space as $A^* = A \rfloor (e\mathbf{I}_3)^{-1} = A \rfloor (e\mathbf{I}_3) = A\,(e\mathbf{I}_3)$, so we get
$$(\mathbf{n} - \delta e)\,(e\,\mathbf{I}_3) = \mathbf{n}e\mathbf{I}_3 - \delta\mathbf{I}_3$$
This has the usual four Plücker coefficients $[n_1, n_2, n_3, -\delta]$, but on the trivector basis $\{-\mathbf{e}_2\mathbf{e}_3 e, -\mathbf{e}_3\mathbf{e}_1 e, -\mathbf{e}_1\mathbf{e}_2 e, \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\}$, clearly different from the vector basis for points.

  We can of course also construct the blade representing the hyperplane *directly* (rather than dually), given m points on it, namely as the outer product of the vectors representing those points.

- *And beyond:* These ways of making offset planar subspaces extend easily. An element of the oriented plane through the points $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ is represented by the 3-blade $p \wedge q \wedge r$, and so on for higher dimensional 'offset' subspaces – if the space has enough dimensions to accommodate them. The blades we construct in this way can always be rewritten in the form $A = d\mathbf{A}$, where $\mathbf{A}$ is a purely Euclidean blade, and d is a vector of the form $e + \mathbf{d}$, with $\mathbf{d}$ a Euclidean vector. We should interpret $\mathbf{A}$ as the direction element, and its grade therefore denotes the dimensionality of the flat subspace represented by A. The vector d represents the closest point to the origin (so that $\mathbf{d}$ is the perpendicular support vector).
- *Scalar distances:* A small surprise is that even a 0-blade (i.e. a scalar) is useful: it is the representation of a scalar distance in the Euclidean space (with a sign, but without a direction), as we will see in the next section. Such distances are of course regular elements of geometry, so it is satisfying to find them on a par with position vectors, direction vectors and other elements of higher dimensionality as just another case of a representing blade in the homogeneous model of a flat Euclidean space.

Having such a unified representation for the various geometrical elements implies that computations using them are unified as well: they have just become operations on blades in $(m+1)$-space, blissfully ignorant of what different geometrical situations these computations might represent. This opens the way to caseless computation in geometrical algorithms.

5.2 Caseless subspace interactions in the homogeneous model

The meet and join in the homogeneous model function just as you would expect, providing the intersection of lines, planes, etc. Writing these out in their (Plücker) coordinates retrieves the familiar compact formulas for the coefficients of the result, but now accompanied by automatic evaluation of the basis on which these coefficients should be interpreted. That provides immediate identification of the kind of intersection, in a manner so well integrated that it suggests we might compute on without intermediate interpretation. This leads to caseless geometrical algorithms in which the dimensionality of intermediate results does not affect the data flow.

Even though the actual computation is a caseless meet applied to blades, let us see what is going on in detail in some typical situations. Following the internal computational combination of the Plücker-like coefficients shows how the algebra of the basis elements takes care of the proper intersection computation, at a small additional expense compared to the usual implementation using 'pre-compiled' tables of intersection formulas.

1. *Line and plane:* The meet of a line ℓ and a plane $\pi^* = \mathbf{n} - \delta e$ in general position is computed as:

$$\begin{aligned} \ell \cap \pi &= \pi^* \rfloor \ell = (\mathbf{n} - \delta e) \rfloor (-\mathbf{v} e + \mathbf{m}\mathbf{I}_3) \\ &= -(\mathbf{n} \cdot \mathbf{v}) e + \mathbf{n} \rfloor (\mathbf{m}\mathbf{I}_3) - \delta \mathbf{v} + 0 = -(\mathbf{n} \cdot \mathbf{v}) e + (\mathbf{n} \wedge \mathbf{m}) \mathbf{I}_3 - \delta \mathbf{v} \\ &= -(\mathbf{n} \cdot \mathbf{v}) e + (\mathbf{m} \times \mathbf{n} - \delta \mathbf{v}) \end{aligned}$$

This is the correct result, representing a point at the location $(\delta\mathbf{v}+\mathbf{n}\times\mathbf{m})/(\mathbf{n}\cdot\mathbf{v})$ in its homogeneous (Plücker) coordinates. Note how the orthogonality relationships between the basis elements automatically kill the potential term involving δ and $\mathbf{m}$. But the fact that this term is zero *is* computed, and that is a slight inefficiency relative to the direct implementation of the same result from a table with Plücker formulas. It is the computational price we pay for the membership of the full geometric algebra.

2. *Two lines:* The meet of two lines in general position is a measure of their signed distance (remember that in this model a dual is made through right-multiply by $e\mathbf{I}_3$, so the dual of the line $-\mathbf{v}_2 e + \mathbf{m}_2\mathbf{I}_3$ is $-\mathbf{v}_2\mathbf{I}_3 + \mathbf{m}_2 e$):

$$\ell_1 \cap \ell_2 = \ell_2^* \rfloor \ell_1 = (-\mathbf{v}_2\mathbf{I}_3 + \mathbf{m}_2 e)\rfloor(-\mathbf{v}_1 e + \mathbf{m}_1\mathbf{I}_3) = \mathbf{m}_2\cdot\mathbf{v}_1 + \mathbf{v}_2\cdot\mathbf{m}_1,$$

retrieving the well-known compact Plücker way of determining how lines pass each other in space; three tests on the signs of such quantities representing the edges of a triangle determine efficiently whether a ray hits the triangle. Again the basis orthogonality relationships have made terms containing $\mathbf{m}_1\cdot\mathbf{m}_2$ or $\mathbf{v}_1\cdot\mathbf{v}_2$ equal to zero.

The directional outcomes are accompanied by numerical factors (such as $\mathbf{n}\cdot\mathbf{v}$ in the first example) relating to the numerical significance of the computation. These are an intrinsic part of the computation of the object, not just secondary aspects that need to be thought of separately (with the danger of being *ad hoc*) or that need be computed separately (costing time).

5.3 The double homogeneous model

An embedding of Euclidean space into a representational space of *two* extra dimensions and its geometric algebra has been recently shown to be very powerful and simplifying [10]. This *double homogeneous model* of Euclidean space embeds the Euclidean distance properties into the fabric of the algebra used to compute with it.

- The inner product is defined in terms of the Euclidean distance d_E between points: $p\cdot q = -\frac{1}{2}d_E^2(\mathcal{P},\mathcal{Q})$. That means that the representational space has a rather special metric, since it follows that $p\cdot p = 0$ for any vector p. Only when one assigns a special point as the origin can one define vectors denoting the relative position of a point: such a vector $\mathbf{p}$ does of course have a non-zero norm. But all computations can be specified without ever introducing such an origin.

- The outer product constructs spheres: a k-blade represents a Euclidean $(k-1)$-sphere. As a consequence, $p\wedge q$ is the ordered point pair $(\mathcal{P},\mathcal{Q})$, $p\wedge q\wedge r$ the circle through $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$. This provides 'Plücker coordinates for spheres'. The dual of the $(m+1)$-blade representing an m-sphere in E^m is a vector in the double homogeneous representation space whose coefficients immediately provides center and radius of the sphere. Flat subspaces are represented as spheres through infinity, and this is possible because one of the two extra representational dimensions is a vector representing the point at infinity.

- The sandwiching by the geometric product gives not only rotations, but all conformal mappings, including translation and spherical inversion. (This is why the double homogeneous model is sometimes called the *conformal model*.)

- The meet of two blades is interpretable as intersecting k-spheres in m-space, and its embedding again reduces the separate cases that would need to be distinguished for such intersections.

This model looks appropriate for many computer graphics applications, and we are currently developing it further for practical usage. It is a truly coordinate-free model, in which all operations of Euclidean geometry can be specified without ever referring to an origin.

6 Some remarks on implementation

The geometric algebra of an m-dimensional vector space contains nicely linear objects (the blades) that can be represented on a basis which should contain 2^m elements (since we need $\binom{m}{k}$ for each k-blade). The various products are all linear, and can be implemented using matrix products of $2^m \times 2^m$ matrices. That may be straightforward, but it is obviously inefficient in both space and time. This is even more urgent when we use the homogeneous model in which an m-dimensional Euclidean space requires an $(m+1)$-dimensional geometric algebra in the homogeneous model, or the $(m+2)$-dimensional double homogeneous model. It seems a lost cause.

However, the recognition that the important elements are blades and their products suggests a more efficient implementation. When two blades multiply by inner or outer product, a blade of unique grade results. This suggests designing a data representation for the elements of geometric algebra that permits easy retrieval of their grades, and also to automatically generate optimized code for the inner and outer multiplication of blades of specific grades k and ℓ. The geometric product generates a more general element of a mixed, but still limited, set of grades $|\ell-k|, |\ell-k|+2, \cdots k+\ell$. Division requires inversion, but this is closely related to the much simpler operation of reversion, simply switching the signs of certain grades. The structural membership of an element to the larger geometric algebra, with its benefit of unified relationships between the various operations, is then merely paid for by a grade-dependent jump to a piece of code; when processing data in a batch mode with many similar operations, this would not slow down things significantly.

Work is underway on an efficient implementation which capitalizes on these structural properties of geometric algebra [6]. First results look very promising indeed, getting close to the usual efficiency of the geometrical computations, but in a simpler code without exceptions or *ad hoc* data structures, and naturally integrating computational techniques which classically belong to different realms than vector/matrix algebra (such as quaternions and Plücker coordinates).

7 Conclusion

This two-part introduction of geometric algebra intends to alert you to the existence of a small set of products that appears to generate all geometric constructions in one consistent framework. Using this framework can simplify the set of data structures representing objects since it inherently encodes all relationships and symmetries of the geometrical primitives in those operators. While there are many interesting facets to geometric algebra, we would like to highlight the following:

- *Division by subspaces*; having a geometric product with an inverse allows us to divide by subspaces, increasing our ability to manipulate algebraic equations involving vectors.
- *Subspaces are basic elements of computation*; thus, no special representations are needed for subspaces of dimension greater than 1 (e.g., tangent planes), and we can manipulate them like we manipulate vectors.
- *Generalization*; expressions for operations on subspaces are often as simple as those for vectors (especially true for linear operations), and as easy to compute.
- *Caseless computation*; degenerate cases are computed automatically, results remain interpretable, and the computation allows us to test the numerics of the solution.
- *Quaternions*; in geometric algebra, quaternions are subsumed and become a natural part of the algebra, with no need to convert between representations to perform rotations.
- *Plücker coordinates*; Plücker coordinates and the concise expressions they give for the interactions of lines, planes, etc. are subsumed and extended.

This paper only covers some of what we felt to be the most important or useful ideas of geometric algebra as it relates to computer graphics. Many topics have been left out, including a description of more geometries (the homogeneous model implements and generalizes the Grassmann spaces of [8], the double homogeneous model implements and generalizes projective spaces); and a lot more can be said about differentiation and coordinate-free differential geometry. The reader may be able to glean the connections from the suggested further reading (see below), but an accessible explanation for computer graphics of such issues still needs to be given.

Further reading

There is a growing body of literature on geometric algebra. Unfortunately much of the more readable writing is not very accessible, being found in specialized books rather than general journals. Little has been written with computer science in mind, since the initial applications have been to physics. No practical implementations in the form of libraries with algorithms yet exist (though there are packages for Maple [2] and Matlab [7] that can be used as a study-aid or for algorithm design). We would recommend the following as natural follow-ups on this paper:

- GABLE: a Matlab package for geometric algebra, accompanied by a tutorial [7].
- The introductory chapters of 'New Foundations of Classical Mechanics' [9].
- An introductory course intended for physicists [3].
- An application to a basic but involved geometry problem in computer vision, with a brief introduction into geometric algebra [12].
- Papers showing how linear algebra becomes enriched by viewing it as a part of geometric algebra [4, 11].

Read them in approximately this order. We are working on texts more specifically suited for a computer graphics audience; these may first appear as SIGGRAPH courses.

Acknowledgments

This work was supported in part by the Netherlands Organization for Scientific Research and by the Natural Sciences and Engineering Research Council of Canada.

References

[1] T.A. Bouma, L. Dorst, H. Pijls, *Geometric Algebra for Subspace Operations*, submitted to Applicandae Mathematicae, preprint available at `http://xxx.lanl.gov/abs/math.LA/0104159`.

[2] A. Lasenby, M. Ashdown et al., *GA package for Maple V*, 1999, at `http://www.mrao.cam.ac.uk/~clifford/software/GA/`

[3] C. Doran and A. Lasenby, *Physical Applications of Geometric Algebra*, 2001, at `http://www.mrao.cam.ac.uk/~clifford/ptIIIcourse/`

[4] C. Doran, A. Lasenby, S. Gull, *Chapter 6: Linear Algebra*, in: Clifford (Geometric) Algebras with applications in physics, mathematics and engineering, W.E. Baylis (ed.), Birkhäuser, Boston, 1996.

[5] L.Dorst, S.mann, *Geometric algebra: a computational framework for geometrical applications (part 1: algebra)*, IEEE Computer Graphics and Applications, to appear, 2002.

[6] D. Fontijne, *GAIGEN*, a Geometric Algebra Implementation Generator, `carol.wins.uva.nl/~fontijne/gaigen/`

[7] L.Dorst, S.Mann, T.A.Bouma, GABLE: a Geometric AlgeBra Learning Environment, `www.science.uva.nl/~leo/GABLE/`

[8] R. Goldman, *The Ambient Spaces of Computer Graphics and Geometric Modeling*, IEEE Computer Graphics and Applications, vol.20, pp. 76–84, 2000.

[9] D. Hestenes, *New foundations for classical mechanics*, 2nd edition, D. Reidel, Dordrecht, 2000.

[10] D. Hestenes, *Old wine in new bottles*, in: Geometric Algebra: A Geometric Approach to Computer Vision, Quantum and Neural Computing, Robotics and Engineering, Bayro-Corrochano, Sobczyk, eds, Birkhäuser, 2001, Chapter 24, pp. 498-520.

[11] D. Hestenes, *The design of linear algebra and geometry*, Acta Applicandae Mathematicae 23: 65-93, 1991.

[12] J. Lasenby, W. J. Fitzgerald, C. J. L. Doran and A. N. Lasenby. *New Geometric Methods for Computer Vision*, Int. J. Comp. Vision 36(3), p. 191-213 (1998).

[13] J. Stolfi, *Oriented projective geometry*, Academic Press, 1991.