Geometric Algebra: A Computational Framework For Geometrical Applications

Part 2

Stephen Mann
University of Waterloo

Leo Dorst
University of Amsterdam

This second part of our tutorial uses geometric algebra to represent rotations, intersections, and differentiation. We show how a small set of products simplifies many geometrical operations.

This is the second of a two-part tutorial on geometric algebra. In part one,[1] we introduced blades, a computational algebraic representation of oriented subspaces, which are the basic elements of computation in geometric algebra. We also looked at the geometric product and two products derived from it, the inner and outer products. A crucial feature of the geometric product is that it is invertible.

From that first article, you should have gathered that every vector space with an inner product has a geometric algebra, whether or not you choose to use it. This article shows how to call on this structure to define common geometrical constructs, ensuring a consistent computational framework. The goal is to show 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 you can automatically take care some of the lower level algorithmic aspects, without tricks, exceptions, or hidden degenerate cases by using geometric algebra as a language.

Rotations

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

Rotations in 2D

In the last article, 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 $(\mathbf{a} \wedge \mathbf{b})$ 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 (see Equation 13 from part one[1]) as

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

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 we obtain $\mathbf{x}$ from $\mathbf{c}$ by a rotation, so we should apparently interpret premultiplying by $e^{-\mathbf{I}\phi}$ as a rotation operator in the $\mathbf{I}$ plane.

The vector $\mathbf{c}$ in the $\mathbf{I}$ plane anticommutes 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 Equation 1, we obtain that the rotation is alternatively representable as a postmultiplication:

$$e^{-\mathbf{I}\phi}\mathbf{c} = \mathbf{c}\cos\phi - \mathbf{I}\mathbf{c}\sin\phi$$
$$= \mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi = \mathbf{c}e^{\mathbf{I}\phi} \qquad (2)$$

What is $\mathbf{c}\mathbf{I}$? First, 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 anticlockwise). So $\mathbf{c}\cos\phi + \mathbf{c}\mathbf{I}\sin\phi$ is "a bit of $\mathbf{c}$ plus a bit of its anticlockwise perpendicular," and those amounts are precisely right to make it equal to the rotation by ϕ (see Figure 1b).

0272-1716/02/$17.00 © 2002 IEEE

Angles as geometrical objects

In Equation 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 angle's orientation. If you ask for the scalar magnitude of $\mathbf{I}\phi$ in the plane $-\mathbf{I}$ (the plane from **b** to **a** rather than from **a** to **b**), it is $-\phi$. Therefore, 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, and not a new kind of element, but we use it as an angle, hence the name).

Rotations in m dimensions

Equation 2 rotates only within the plane **I**. Generally, we would like to have rotations in space. For a vector **x**, the outcome of a rotation $\mathrm{R}_{\mathbf{I}\phi}$ should be

$$\mathrm{R}_{\mathbf{I}\phi}\mathbf{x} = \mathbf{x}_{\perp} + \mathrm{R}_{\mathbf{I}\phi}\mathbf{x}_{||}$$

where $\mathbf{x}_{\perp}$ and $\mathbf{x}_{||}$ are the perpendicular and parallel components of **x** relative to the rotation plane **I**, respectively. We have seen that we can separate a vector into such components by commutation (as in Equations 31 and 32 from part one[1]). As you can verify, the following formula effects this separation and rotation simultaneously:

$$\text{rotation over } \mathbf{I}\phi\text{: } \mathbf{x} \mapsto \mathrm{R}_{\mathbf{I}\phi}\mathbf{x} = e^{-\mathbf{I}\phi/2}\mathbf{x}\, e^{\mathbf{I}\phi/2} \qquad (3)$$

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

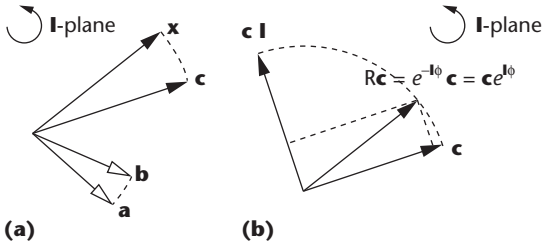
Two successive rotations R_1 and R_2 are equivalent to a single new rotation R of which the rotor R is the geometric product of the rotors R_2 and R_1, since

$$\begin{aligned}\left(\mathrm{R}_2 \circ \mathrm{R}_1\right)\mathbf{x} &= R_2\left(R_1\mathbf{x}\,R_1^{-1}\right)R_2^{-1} \\ &= \left(R_2R_1\right)\mathbf{x}\left(R_2R_1\right)^{-1} = R\,\mathbf{x}\,R^{-1}\end{aligned}$$

with $R = R_2R_1$. Therefore, the combination of rotations is a simple consequence of the application of the geometric product on rotors—that is, 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 us see how it works in 3-space. In three dimensions, we are used to specifying rotations by a *rotation axis* **a** rather than by a *rotation plane* **I**. Given a unit vector **a** for an axis, we find the plane as the 2-blade complementary to it in the 3D space with volume element $\mathbf{I}_3$: $\mathbf{I} = \mathbf{a} \lrcorner \mathbf{I}_3 = \mathbf{a}\mathbf{I}_3 = \mathbf{I}_3\mathbf{a}$. A rotation over an angle ϕ around an axis with unit vector **a** is therefore represented by the rotor $e^{-\mathbf{I}_3\mathbf{a}\phi/2}$.

For example, 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



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

over $\pi/2$, we write out their rotors (using the shorthand $\mathbf{e}_{23} = \mathbf{e}_2 \wedge \mathbf{e}_3 = \mathbf{e}_2\,\mathbf{e}_3$ and so on):

$$R_1 = e^{-\mathbf{I}_3\mathbf{e}_1\pi/4} = \frac{1-\mathbf{e}_{23}}{\sqrt{2}}$$

and

$$R_2 = e^{-\mathbf{I}_3\mathbf{e}_2\pi/4} = \frac{1-\mathbf{e}_{31}}{\sqrt{2}}$$

The total rotor is their product, and we rewrite it back to the exponential form to find the axis:

$$\begin{aligned}R \equiv R_2R_1 &= \frac{1}{2}\left(1-\mathbf{e}_{31}\right)\left(1-\mathbf{e}_{23}\right) \\ &= \frac{1}{2}\left(1-\mathbf{e}_{23}-\mathbf{e}_{31}+\mathbf{e}_{12}\right) \\ &= \frac{1}{2} - \frac{1}{2}\sqrt{3}\,\mathbf{I}_3\frac{\mathbf{e}_1+\mathbf{e}_2-\mathbf{e}_3}{\sqrt{3}} \\ &\equiv e^{-\mathbf{I}_3\mathbf{a}\pi/3}\end{aligned}$$

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. We can apply a rotor immediately to an arbitrary blade through the formula

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

This lets you rotate a plane in one operation, for instance, using a rotation by R (as in the example we just saw):

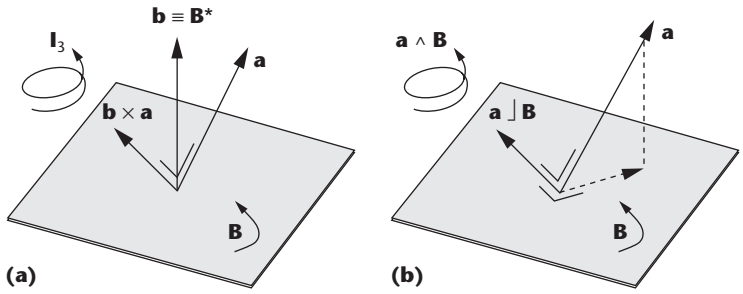
$$\begin{aligned}&R\left(\mathbf{e}_1 \wedge \mathbf{e}_2\right)R^{-1} = \\ &\frac{1}{4}\left(1-\mathbf{e}_{23}-\mathbf{e}_{31}+\mathbf{e}_{12}\right)\mathbf{e}_{12}\left(1+\mathbf{e}_{23}+\mathbf{e}_{31}-\mathbf{e}_{12}\right) = -\mathbf{e}_{31}\end{aligned}$$

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

Quaternions based on bivectors

You might have recognized the last example 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



2 (a) The dual B* of a bivector B and the cross product with a. (b) The same result using the inner product of blades (from part one[1]).

of that a bivector basis $\{\mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}\}$. Note that these elements satisfy $\mathbf{e}_{12}^2=\mathbf{e}_{23}^2=\mathbf{e}_{31}^2=-1$, $\mathbf{e}_{12}\,\mathbf{e}_{23} = \mathbf{e}_{31}$ (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 $i^2 = j^2 = k^2 = ijk = -1$ and $ji = k$ and cyclic. Algebraically, these objects form a basis for quaternions obeying the quaternion product, commonly interpreted as some kind of 4D complex number system. To us, there is nothing complex about quaternions, but they are not vectors either—they are real 2-blades in 3-space, denoting elementary rotation planes and multiplying through the geometric product. Therefore, visualizing quaternions is straightforward; each is a rotation plane with a rotation angle, and the bivector angle concept represents that well.

Of course, in geometric algebra, we can combine quaternions 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 anticommutative (for example, $i\,\mathbf{e}_1 = \mathbf{e}_1\,i$, 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 Equation 3 be the rotation of a vector.

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 it acquires much extended coordinate-free methods through this inclusion. We show some of the basic principles in this article, but you can find more in related literature.[2,3]

Outermorphisms

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 $f: V^m \to V^m$ of a vector space has a natural extension to the whole geometric algebra of that vector space, as an *outermorphism*—that is, as a mapping that preserves the outer product structure:

$$f(\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \ldots \wedge \mathbf{a}_k) \equiv f(\mathbf{a}_1) \wedge f(\mathbf{a}_2) \wedge \ldots \wedge f(\mathbf{a}_k) \qquad (4)$$

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 $f(\alpha) = \alpha$. Geometrically, this means that a linear transformation leaves the origin intact.

Linear transformations are outermorphisms, which explains why we can generalize so many operations from vectors to general subspaces in a straightforward manner.

Dual representation

Linear algebra often uses dual representations of hyperplanes to extend the scope of vector-based operations. In geometric algebra, dualization is also important and in fact is a 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:

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

where $\mathbf{I}_m$ is the pseudoscalar of V^m (an m-blade giving the volume element) and $\tilde{}$ is the reversion (obtained by reversing the factors of the blade it is applied to and leading to a grade-dependent 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 3D 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 $\mathbf{B} = b_1\mathbf{e}_2 \wedge \mathbf{e}_3 + b_2\mathbf{e}_3 \wedge \mathbf{e}_1 + b_3\mathbf{e}_1 \wedge \mathbf{e}_2$. We take the dual relative to the space with volume element $\mathbf{I}_3 \equiv \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ (that is, the right-handed volume formed by using a right-handed basis). The subspace of $\mathbf{I}_3$ dual to $\mathbf{B}$ is then

$$\begin{aligned}\mathbf{B} \rfloor \tilde{\mathbf{I}}_3 &= \\ &= \left(b_1\mathbf{e}_2 \wedge \mathbf{e}_3 + b_2\mathbf{e}_3 \wedge \mathbf{e}_1 + b_3\mathbf{e}_1 \wedge \mathbf{e}_2\right) \rfloor \left(\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1\right) \\ &= b_1\mathbf{e}_1 + b_2\mathbf{e}_2 + b_3\mathbf{e}_3 \end{aligned} \qquad (5)$$

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 will 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 $\boldsymbol{B}$ if

$$\mathbf{x} \in \boldsymbol{B} \;\Leftrightarrow\; \mathbf{x} \wedge \mathbf{B} = 0 \qquad (6)$$

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

$$\mathbf{x} \in \boldsymbol{B} \;\Leftrightarrow\; \mathbf{x} \rfloor \mathbf{B}^* = 0 \qquad (7)$$

We switch between the two standpoints by using the distributive relation $(A \wedge B) \rfloor C = A \rfloor (B \rfloor C)$ (which is Equa-

tion 19 from part one[1]) used for a vector **x**, a blade **B**, and a pseudoscalar **I**:

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

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

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

If **x** is known to be in the subspace of **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 obvious.

Cross product

Classical computations with vectors in 3-space often use the cross product $\times$, which produces from two vectors **a** and **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):

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

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

Equation 10 explicitly shows several things that we always need 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 because it is perpendicular to a plane (this is coded in the usage of the inner product $\rfloor$), and
- the construction only works in three dimensions because only then is the dual of a 2-blade a vector (this is coded in the 3-gradedness of $\mathbf{I}_3$).

The vector product $\mathbf{a} \wedge \mathbf{b}$ does not depend on any of these embedding properties yet characterizes the (**a**, **b**) 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.

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 Equation 4 because perpendicularity is a more complicated concept to transform than spanning. Hestenes[3] gives the general transformation formula. For blades, it becomes

$$f(\mathbf{A} \rfloor \mathbf{B}) = \bar{f}^{-1}(\mathbf{A}) \rfloor f(\mathbf{B})$$

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

Because of the complexity of this transformation behavior, we should steer clear of any constructions that involve the inner product, especially when characterizing basic properties of geometric 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 plane's normal vector. (This is a well-known fact, but it is always a shock to novices.) Rather, the normal vector is a cross product of vectors, which transforms as

$$f(\mathbf{a} \times \mathbf{b}) = \bar{f}^{-1}(\mathbf{a}) \times \bar{f}^{-1}(\mathbf{b}) \det(f) \tag{11}$$

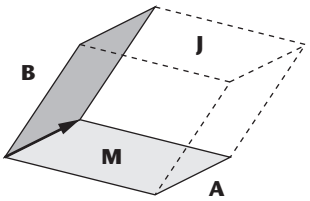
where det(f) is the determinant defined by $\det(f) = f(\mathbf{I}_m)\mathbf{I}_m^{-1}$. (This is a coordinate-free definition of the determinant of the matrix representation of f.) The right-hand side of Equation 11 is usually not equal to $f(\mathbf{a}) \times f(\mathbf{b})$, so a linear transformation is not cross-product preserving. Therefore, it is 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 because 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.

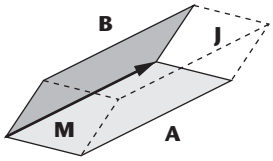
Intersecting subspaces

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 related literature, causing some confusion. In this article, we will use the set notations $\cup$ and $\cap$ to make the formulas more easily readable.

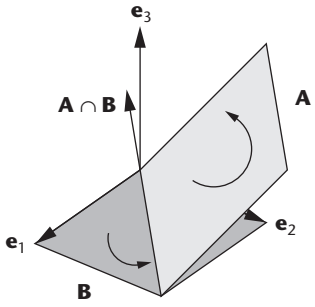
The Join of two subspaces is their smallest superspace—that is, the smallest space containing them both. Representing the spaces by blades **A** and **B**, the Join is denoted $\mathbf{A} \cup \mathbf{B}$. If the subspaces of **A** and **B** are disjoint, their Join is obviously proportional to $\mathbf{A} \wedge \mathbf{B}$. A problem is that if **A** and **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 blades' reshapable nature (see Figure 3). (Stolfi[4] also observed this ambiguity.) Fortunately, in all geometrically relevant entities that we compute, it appears that this scalar ambiguity cancels out.

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 cannot characterize it with a list of algebraic computation rules. Although computation of the Join





3 The ambiguity of scale for Meet **M** and Join **J** of two blades **A** and **B**. Both figures are acceptable solutions to the problem of finding a blade representing the union and intersection of the subspaces of the blades **A** and **B**.



4 An example of the Meet of two planes.

may appear to require some optimization process, the smallest superspace is directly related to the nonzero part of highest grade in the expansion of the geometric product and can therefore be done in constant time.[5]

The Meet of two subspaces **A** and **B** is their largest common subspace. Given the Join $\mathbf{J} \equiv \mathbf{A} \cup \mathbf{B}$ of **A** and **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. In formula, this is

$$\left(\mathbf{A} \cap \mathbf{B}\right) \rfloor \tilde{\mathbf{J}} = \left(\mathbf{B} \rfloor \tilde{\mathbf{J}}\right) \wedge \left(\mathbf{A} \rfloor \tilde{\mathbf{J}}\right)$$

or

$$\left(\mathbf{A} \cap \mathbf{B}\right)^* = \mathbf{B}^* \wedge \mathbf{A}^* \tag{12}$$

with the dual taken with respect to the Join **J**. (The somewhat strange order in Equation 12 means that the Join **J** can be written using the Meet **M** in the factorization $\mathbf{J} = (\mathbf{A}\mathbf{M}^{-1}) \wedge \mathbf{M} \wedge (\mathbf{M}^{-1}\mathbf{B})$, and it corresponds to Stolfi[4] for vectors.) This leads to a formula for the Meet of **A** and **B** relative to the chosen Join (use Equation 8):

$$\mathbf{A} \cap \mathbf{B} = \left(\mathbf{B} \rfloor \tilde{\mathbf{J}}\right) \rfloor \mathbf{A} \tag{13}$$

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 $\mathbf{A} \rfloor \tilde{\mathbf{A}} = 1 = \mathbf{B} \rfloor \tilde{\mathbf{B}}$; it gives the Meet of these blades a numerical factor that we can interpret geometrically.) These are planes in general position in 3D space, so their Join is proportional to $\mathbf{I}_3$. It makes sense to take $\mathbf{J} = \mathbf{I}_3$. For the Meet, this gives

$$\begin{aligned}
\mathbf{A} \cap \mathbf{B} &= \\
&= \frac{1}{2}\left(\left(\mathbf{e}_1 \wedge \mathbf{e}_2\right) \rfloor \left(\mathbf{e}_3 \wedge \mathbf{e}_2 \wedge \mathbf{e}_1\right)\right) \rfloor \left(\left(\mathbf{e}_1 + \mathbf{e}_2\right) \wedge \left(\mathbf{e}_2 + \mathbf{e}_3\right)\right) \\
&= \frac{1}{2}\mathbf{e}_3 \rfloor \left(\left(\mathbf{e}_1 + \mathbf{e}_2\right) \wedge \mathbf{e}_3\right) \\
&= -\frac{1}{2}\left(\mathbf{e}_1 + \mathbf{e}_2\right) = -\frac{1}{\sqrt{2}}\left(\frac{\mathbf{e}_1 + \mathbf{e}_2}{\sqrt{2}}\right)
\end{aligned} \tag{14}$$

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 Stolfi,[4] the sign of $\mathbf{A} \cap \mathbf{B}$ is the right-hand rule applied to the turn required to make **A** coincide with **B**, in the correct orientation.

Classically, we usually compute the intersection of two planes in 3-space by first converting them to normal vectors and then taking the cross product. This gives the same answer in this nondegenerate case in 3-space, using our previous Equation 9, Equation 8, and noting that $\tilde{\mathbf{I}}_3 = -\mathbf{I}_3$:

$$\begin{aligned}
\left(\mathbf{A} \rfloor \tilde{\mathbf{I}}_3\right) \times \left(\mathbf{B} \rfloor \tilde{\mathbf{I}}_3\right) &= \left(\left(\mathbf{A} \rfloor \tilde{\mathbf{I}}_3\right) \wedge \left(\mathbf{B} \rfloor \tilde{\mathbf{I}}_3\right)\right) \rfloor \tilde{\mathbf{I}}_3 \\
&= \left(\left(\mathbf{B} \rfloor \tilde{\mathbf{I}}_3\right) \wedge \left(\mathbf{A} \rfloor \tilde{\mathbf{I}}_3\right)\right) \rfloor \mathbf{I}_3 \\
&= \left(\mathbf{B} \rfloor \tilde{\mathbf{I}}_3\right) \rfloor \left(\left(\mathbf{A} \rfloor \tilde{\mathbf{I}}_3\right) \rfloor \mathbf{I}_3\right) \\
&= \left(\mathbf{B} \rfloor \tilde{\mathbf{I}}_3\right) \rfloor \mathbf{A} = \mathbf{A} \cap \mathbf{B}
\end{aligned}$$

So the classical result is a special case of Equation 13, but Equation 13 is much more general; it applies to the intersection of subspaces of any grade, within a space of any dimension.

The norm of the Meet gives an impression of the intersection strength. 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 signs. This numerical significance is useful in applications.

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 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.

A rotor's scalar differentiation

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 $\mathbf{X} = R\mathbf{X}_0R^{-1}$ of some constant blade $\mathbf{X}_0$. Using the chain rule and commutation rules, scalar differentiation with respect to t gives

$$\begin{aligned}\frac{d}{dt}\mathbf{X} &= \frac{d}{dt}\left(e^{-\mathbf{I}\phi/2}\mathbf{X}_o e^{\mathbf{I}\phi/2}\right)\\ &= -\frac{1}{2}\frac{d}{dt}\left(\mathbf{I}\phi\right)\left(e^{-\mathbf{I}\phi/2}\mathbf{X}_o e^{\mathbf{I}\phi/2}\right)\\ &\quad + \frac{1}{2}\left(e^{-\mathbf{I}\phi/2}\mathbf{X}_o e^{\mathbf{I}\phi/2}\right)\frac{d}{dt}\left(\mathbf{I}\phi\right)\\ &= \frac{1}{2}\left(\mathbf{X}\frac{d}{dt}\left(\mathbf{I}\phi\right) - \frac{d}{dt}\left(\mathbf{I}\phi\right)\mathbf{X}\right)\\ &= \mathbf{X}\otimes\frac{d}{dt}\left(\mathbf{I}\phi\right) \end{aligned} \tag{15}$$

using the commutator product $\otimes$ defined in geometric algebra as the shorthand $A\otimes B \equiv 1/2(AB-BA)$. This product often crops up in computations with continuous groups such as the rotations.

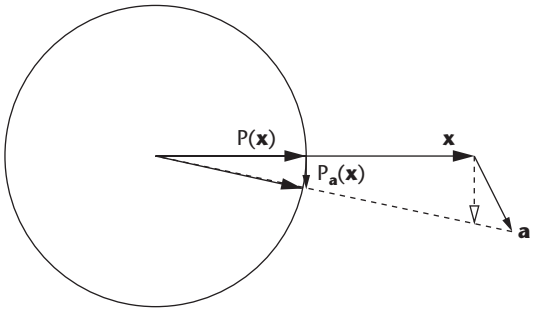
The simple expression that results assumes a more familiar form when $\mathbf{X}$ is a vector $\mathbf{x}$ in 3-space and when the attitude of the rotation plane is fixed so that $d\mathbf{I}/dt = 0$. We introduce a scalar angular velocity $\omega \equiv d\phi/dt$, and the vector dual to the plane as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

$$\begin{aligned}\frac{d}{dt}\mathbf{x} &= \mathbf{x}\otimes\frac{d}{dt}\left(\mathbf{I}\phi\right) = \mathbf{x}\otimes\left(\mathbf{w}\rfloor\mathbf{I}_3\right)\\ &= \mathbf{x}\rfloor\left(\mathbf{w}\rfloor\mathbf{I}_3\right) = \left(\mathbf{x}\wedge\mathbf{w}\right)\rfloor\mathbf{I}_3 = \mathbf{w}\times\mathbf{x}\end{aligned}$$

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

Differentiation of spherical projection

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



5 The derivative of the spherical projection.

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

We recognize the result as the rejection of $\mathbf{a}$ by $\mathbf{x}$, scaled appropriately (see the "Projecting subspaces" section from part one[1]). Figure 5 confirms the outcome. You can 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.

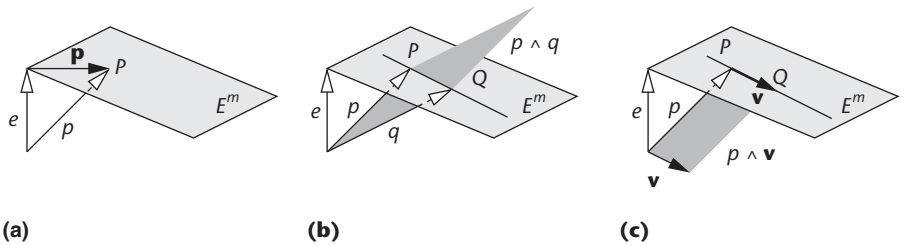
For more advanced usage of differentiation relative to blades, see the Doran et al.[2] tutorial, which introduces these differentiations using examples from physics, and the Lasenby et al.[6] application paper.

Models of geometry

Geometric algebra can help us express several standard models of geometry. 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 as well as a model of Euclidean geometry that naturally handles spheres.

Homogeneous model

So far we have been treating only homogeneous subspaces of the vector space V^n—that is, 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 ideas behind homogeneous coordinates to geometric algebra. It turns out that such elements of geometry can also be represented by blades, in a representational space with an extra dimension. The geometric algebra of this space gives us precisely what we need. In this view, more complicated geometrical objects do not require new operations or tech-



6 Representing offset subspaces of E^m in $(m+1)$ dimensional space. (a) A point P denoted by a vector p with Euclidean part p. (b) A line element is represented by the bivector formed as the outer product of two points. (c) Reshaping the bivector shows the correspondence with Plücker coordinates; here, $\mathbf{v} \equiv \mathbf{q} - \mathbf{p}$.

niques, merely the standard computations in a higher dimensional space.

The homogeneous model is often described as augmenting a 3D 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 nonlinear operations such as translations implementable as linear mappings.

We 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 $e \rfloor \mathbf{x} = 0$ for all $\mathbf{x} \in E^m$. We also need to define $e \rfloor e$ to make our algebra complete. This involves some dilemmas that are only fully resolvable in the double homogeneous model, which we explain later on. For now, we can take $e \rfloor e = 1$. We interpret e as 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 translating the point at the origin over the Euclidean vector $\mathbf{p}$. We do this by adding $\mathbf{p}$ to e. This construction gives the representation of the point P at location $\mathbf{p}$ as the vector p in $(m+1)$-dimensional space:

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

This is 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 $m = 2$).

We can multiply these vectors in $(m+1)$-dimensional space 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:

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

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 P and Q. The former is the *direction vector* of the directed line; the latter is an area that we 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 $\mathbf{d}$:

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

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 $d = e + \mathbf{d}$), separating the positional part 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 we can construct in all these ways.

Temporarily reverting to the cross product to make a connection with a classical representation, we can rewrite Equation 16 as

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

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

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

The six coefficients $[-v_1, -v_2, -v_3, m_1, m_2, m_3]$ of the line in this representation are 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 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 $\mathbf{x} \rfloor \mathbf{n} = \delta$, this can be written as $(e + \mathbf{x}) \rfloor (\mathbf{n} - \delta e) = 0$, so $x \rfloor (\mathbf{n} - \delta e) = 0$. 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 $\{-\mathbf{e}_2\mathbf{e}_3 e, -\mathbf{e}_3\mathbf{e}_1 e, -\mathbf{e}_1\mathbf{e}_2 e, \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\}$—it is clearly different from the vector basis for points.

Of course, we can 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 P, Q, and 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 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, so its grade 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 (that is, 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.

Caseless subspace interactions

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.

The Meet and Join in the homogeneous model function just as you would expect, providing the intersection of lines, planes, and so forth. Writing these out in their (Plücker) coordinates retrieves the familiar compact formulas for the coefficients of the result, but they are 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 continue our computation 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 precompiled tables of intersection formulas. (We write the contraction of vectors as the classical inner product to show the correspondence clearly.)

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

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

  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.

- *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 a right-multiply by $e\mathbf{I}_3$, so the dual of the line $-\mathbf{v}_2 e + \mathbf{m}_2\mathbf{I}_3$ is $-\mathbf{v}_2\mathbf{I}_3 + \mathbf{m}_2 e$):

$$\begin{aligned} l_1 \cap l_2 &= l_2^* \rfloor l_1 = \left(-\mathbf{v}_2\mathbf{I}_3 + \mathbf{m}_2 e\right) \rfloor \left(-\mathbf{v}_1 e + \mathbf{m}_1\mathbf{I}_3\right) \\ &= \mathbf{m}_2 \cdot \mathbf{v}_1 + \mathbf{v}_2 \cdot \mathbf{m}_1 \end{aligned}$$

  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 computation's numerical significance. These are an intrinsic part of the object's computation, not just secondary aspects we need to think of separately (with the danger of being ad hoc) or that we need to compute separately (costing time).

Double homogeneous model

Hestenes[7] has recently shown that embedding of Euclidean space into a representational space of two extra dimensions and its geometric algebra is powerful and simplifying. 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: $p \cdot q = -\frac{1}{2} d_E^2(P,Q)$. That means that the representational space has a rather special metric because it follows that $p \cdot p = 0$ for any vector p. Only when we assign a special point as the origin can we define vectors denoting the relative position of a point—such a vector $\mathbf{p}$ does of course have a nonzero norm. But we can specify all computations without ever introducing such an origin.
- The outer product constructs spheres—a k-blade rep-

Correction

Due to a printing error, the indices in Equation 11 of part one[1] are unreadable. The equation should read

$$\left(\mathbf{e}_i \wedge \mathbf{e}_j\right)^2 = \left(\mathbf{e}_i \wedge \mathbf{e}_j\right)\left(\mathbf{e}_i \wedge \mathbf{e}_j\right) = \left(\mathbf{e}_i\mathbf{e}_j\right)\left(\mathbf{e}_i\mathbf{e}_j\right)$$
$$= \mathbf{e}_i\mathbf{e}_j\mathbf{e}_i\mathbf{e}_j = -\mathbf{e}_i\mathbf{e}_i\mathbf{e}_j\mathbf{e}_j = -1$$

resents a Euclidean $(k-1)$-sphere. As a consequence, $p \wedge q$ is the ordered point pair (P, Q) and $p \wedge q \wedge r$ is the circle through P, Q, and 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, which has coefficients that immediately provide the sphere's center and radius. Flat subspaces are represented as spheres through infinity. 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 rotations and all conformal mappings, including translation and spherical inversion. This is why the double homogeneous model is often 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 we can specify all operations of Euclidean geometry without ever referring to an origin.

Implementation

The geometric algebra of an m-dimensional vector space contains nicely linear objects (the blades) that can be represented on a basis that should contain 2^m elements (because we need $\binom{m}{k}$ for each k blade). The various products are all linear, and we can implement them using matrix products of $2^m \times 2^m$ matrices. That might 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, or the $(m+2)$-dimensional double homogeneous model. It seems a lost cause.

However, recognizing that the important elements are blades and their products suggests a more efficient implementation. When two blades multiply by an 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 automatically generating optimized code for the inner and outer multiplication of blades of specific grades k and l. Their geometric product generates a more general element of a mixed, but still limited, set of grades $|l-k|, |l-k|+2, \ldots k+l$. 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 that capitalizes on these structural properties of geometric algebra.[8] Our first results look promising and are 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 that classically belong to different realms than vector/matrix algebra (such as quaternions and Plücker coordinates).

Conclusion

This two-part introduction to 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 because it inherently encodes all relationships and symmetries of the geometrical primitives in those operators. Although 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 lets us 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 a dimension greater than 1 (for example, 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 (that is especially true for linear operations) and as easy to compute.
- *Caseless computation*. Degenerate cases are computed automatically, results remain interpretable, and the computation lets us test the solution's numerics.
- *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*. Geometric algebra subsumes and extends Plücker coordinates and the concise expressions they give for the interactions of lines, planes, and so on.

This article only covers some of what we feel are the most important or useful ideas of geometric algebra as it relates to computer graphics. We have left out many topics, including a description of more geometries (the homogeneous model implements and generalizes the Grassmann spaces of Goldman[9] and the double homogeneous model implements and generalizes projective spaces), and we could say a lot more about differentiation

and coordinate-free differential geometry. You should be able to glean the connections from the "Further Reading" sidebar, although an accessible explanation for computer graphics of such issues is still necessary. ■

Acknowledgments

The Netherlands Organization for Scientific Research and the Natural Sciences and Engineering Research Council of Canada supported this work in part.

References

1. L. Dorst and S. Mann, "Geometric Algebra: A Computational Framework for Geometrical Applications (Part 1)," *IEEE Computer Graphics and Applications*, vol. 22, no. 3, May/June 2002, pp. 24-31.
2. C. Doran and A. Lasenby, *Physical Applications of Geometric Algebra*, 2001, http://www.mrao.cam.ac.uk/~clifford/ptIIIcourse/.
3. D. Hestenes, "The Design of Linear Algebra and Geometry," *Acta Applicandae Mathematicae*, vol. 23, 1991, pp. 65-93.
4. J. Stolfi, *Oriented Projective Geometry*, Academic Press, San Diego, 1991.
5. T.A. Bouma, L. Dorst, and H. Pijls, "Geometric Algebra for Subspace Operations," to be published in *Acta Applicandae Mathematicae*, preprint available http://xxx.lanl.gov/abs/math.LA/0104159.
6. J. Lasenby et al., "New Geometric Methods for Computer Vision," *Int'l J. Computer Vision*, vol. 36, no. 3, 1998, pp. 191-213.
7. D. Hestenes, "Old Wine in New Bottles," *Geometric Algebra: A Geometric Approach to Computer Vision, Quantum and Neural Computing, Robotics, and Engineering*, E. Bayro-Corrochano and G. Sobczyk, eds., Birkhäuser, Boston, 2001, pp. 498-520.
8. D. Fontijne, *GAIGEN: A Geometric Algebra Implementation Generator*, http://carol.wins.uva.nl/~fontijne/gaigen/.
9. R. Goldman, "The Ambient Spaces of Computer Graphics and Geometric Modeling," *IEEE Computer Graphics and Applications*, vol. 20, no. 2, Mar./Apr. 2000, pp. 76-84.

Stephen Mann *is an associate professor in the School of Computer Science at the University of Waterloo. He has a BA in computer science and pure mathematics from the University of California, Berkeley, and a PhD in computer science and engineering from the University of Washington. His research interests are in splines and the mathematical foundations of computer graphics.*

Leo Dorst *is an assistant professor at the Informatics Institute at the University of Amsterdam. His research interests include geometric algebra and its applications to computer science. He has an MSc and PhD in the applied physics of computer vision from Delft University of Technology, The Netherlands.*

Further Reading

There is a growing body of literature on geometric algebra. Unfortunately, much of the more readable writing is not very accessible and in specialized books rather than general journals. Researchers have written little with computer science in mind because the initial applications have been to physics. We recommend the following as natural follow-ups to this article:

- Gable, a Matlab package for geometric algebra, accompanied by a tutorial.[1]
- An application software generator Gaigen.[2]
- The introductory chapters of *New Foundations of Classical Mechanics*.[3]
- An introductory course intended for physicists.[4]
- An application to a basic but involved geometry problem in computer vision, with a brief introduction into geometric algebra.[5]
- Papers showing how linear algebra becomes enriched by viewing it as a part of geometric algebra.[6,7]

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

References

1. L. Dorst, S. Mann, and T.A. Bouma, *GABLE: A Geometric Algebra Learning Environment*, http://www.science.uva.nl/~leo/GABLE/.
2. D. Fontijne, *GAIGEN: A Geometric Algebra Implementation Generator*, http://carol.wins.uva.nl/~fontijne/gaigen/.
3. D. Hestenes, *New Foundations for Classical Mechanics*, 2nd ed., D. Reidel, Dordrecht, 2000.
4. C. Doran and A. Lasenby, *Physical Applications of Geometric Algebra*, 2001, http://www.mrao.cam.ac.uk/~clifford/ptIIIcourse/.
5. J. Lasenby et al., "New Geometric Methods for Computer Vision," *Int'l J. Computer Vision*, vol. 36, no. 3, 1998, pp. 191-213.
6. C. Doran, A. Lasenby, and S. Gull, "Linear Algebra," *Clifford (Geometric) Algebras with Applications in Physics, Mathematics, and Engineering*, W.E. Baylis, ed., Birkhäuser, Boston, 1996.
7. D. Hestenes, "The Design of Linear Algebra and Geometry," *Acta Applicandae Mathematicae*, vol. 23, 1991, pp. 65-93.

Readers may contact Stephen Mann at the School of Computer Science, Univ. of Waterloo, 200 University Ave. W, Waterloo, Ontario, Canada, email smann@uwaterloo.ca.