Chapter 15: Quaternions: Multiplication in the Space of Mass-Points

Be fruitful and multiply Genesis 1:28

1. Vector Spaces and Division Algebras

The space of mass-points is a 4-dimensional vector space: addition, subtraction, and scalar multiplication are all well-defined operations on mass-points. But what about a more universal form of multiplication? Can we multiply together two points or more generally two mass-points?

In lower dimensional vector spaces, we can indeed multiply any two elements in the space. Every 1-dimensional vector space is isomorphic (equivalent) to the real number line. We can multiply any two points along the number line by the standard rules for multiplication of real numbers. Every 2-dimensional vector space is isomorphic to the plane. If we identify each pair of rectangular coordinates (a,b) in the plane with the complex number $a+bi$, then we can multiply two vectors in the plane using the standard rules for multiplication of complex numbers.

Vectors in 3-dimensions are endowed with two distinct products: dot product and cross product. The dot product of two vectors is a scalar; the cross product of two vectors is a vector. When we add or subtract two vectors, we get another vector. When we say that we want to multiply two vectors, we typically mean that we also want the result to be another vector. The dot product of two vectors is not a vector, so dot product is not the kind of multiplication that we seek. The cross product of two vectors is a vector; nevertheless cross product also has some undesirable properties. Multiplication is typically associative, commutative and distributes through addition. Moreover, in order to solve simple equations, we would like to have an identity for multiplication as well as multiplicative inverses (division). Although cross product distributes through addition, the cross product is neither associative nor commutative. Moreover there is no multiplicative identity for cross product -- that is, there is no vector u such that for all vectors v, we have $u \times v = v \times u = v$ -- since by definition $u \times v \perp v$. Hence there can be no multiplicative inverses for cross product. In fact, in 3-dimensions there is no entirely satisfactory notion of vector multiplication.

A *division algebra* (an algebra where we can perform division) is a set S where we can add, subtract, multiply, and divide any two elements in S. (One exception: we cannot divide by zero, the identity for addition.) Addition and multiplication must satisfy the usual rules: addition is associative and commutative; multiplication is associative and distributes through addition. Notice that we shall not insist that multiplication is commutative, since the multiplication that we have in mind for mass-points is not commutative. There is an identity 0 for addition and an identity 1 for multiplication so that for any element s in S

$$s+0=0+s=s$$

$$s \cdot 1 = 1 \cdot s = s.$$

Finally, for each element s in S, there is an inverse for addition $(-s)$ and an inverse for multiplication (s^{-1}) so that

$$s+(-s)=(-s)+s=0$$

$$s \cdot s^{-1}=s^{-1} \cdot s=1 \qquad (s \neq 0).$$

Thus in a division algebra, we can subtract and divide, as well as add and multiply.

There are only three finite dimensional vector spaces that are also division algebras -- that is, there are only three dimensions in which we can define addition, subtraction, multiplication and division -- dimensions 1,2,4. These vector spaces correspond to the line (real numbers), the plane (complex numbers), and the space of mass-points (quaternions). (There is also a non-associative multiplication in dimension 8, the *Cayley numbers* or *octonions*.) We are really unbelievably lucky that Computer Graphics is grounded in the 4-dimensional vector space of mass-points.

In this chapter we shall define quaternion multiplication in the space of mass-points and investigate applications of quaternion multiplication to Computer Graphics. We will begin, however, with a simpler, better known paradigm: complex multiplication for vectors in the plane.

2. Complex Numbers

Complex numbers are low dimensional easy to understand analogues of the quaternions. Many of the fundamental algebraic and geometric properties of the quaternions appear as well in the complex numbers. One of the themes of this chapter is that many of the algebraic and geometric properties of the quaternions are easier to understand when reduced to the analogous properties of the complex numbers. With this kinship in mind, we begin our discussion of quaternions with a brief review of the complex numbers.

Complex numbers are often represented by vectors in 2-dimensions. The complex number $w=a+bi$ corresponds to the vector $v=(a,b)$ in the xy-plane. Addition, subtraction, and scalar multiplication of complex numbers correspond to coordinate-free operations on vectors in the plane. Rectangular coordinates are introduced in practice to perform these computations, but in principle there is no need for preferred directions -- for coordinate axes -- to define addition, subtraction, or scalar multiplication for vectors in the plane.

Complex multiplication is different. For complex multiplication we must choose a preferred direction: the direction of the identity vector for multiplication. We denote this identity vector by 1 and we associate this vector with the unit vector along the positive x-axis. The unit vector perpendicular to the identity vector we denote by i and we associate this vector with the unit vector along the positive y-axis. Relative to this coordinate system for every vector v in the plane there are constants a,b such that

$$v = a1 + bi\,.$$

Typically we drop the symbol 1, and we write simply

$$v = a + bi\,. \tag{2.1}$$

In the complex number system, i denotes $\sqrt{-1}$, so

$$i^2 = -1\,. \tag{2.2}$$

Since 1 is the identity for multiplication, we have the rules

$$(1)(1) = 1, \qquad (i)(1) = (1)(i) = i, \qquad (i)(i) = -1\,. \tag{2.3}$$

Therefore if we want complex multiplication to distribute through addition, then we must define

$$(a + b\,i)(c + d\,i) = a(c + d\,i) + bi(c + d\,i) = (ac - bd) + (ad + bc)i\,. \tag{2.4}$$

Using Equation (2.4), it is easy to verify that complex multiplication is associative, commutative, and distributes through addition (see Exercise 1).

Every non-zero complex number has a multiplicative inverse. Let $v = a + bi$. The *complex conjugate* of v, denoted by v^*, is defined by

$$v^* = a - bi\,. \tag{2.5}$$

By equation (2.4)

$$v\,v^* = a^2 + b^2 = |v|^2\,. \tag{2.6}$$

Therefore

$$v^{-1} = \frac{v^*}{|v|^2}\,.$$

From Equations (2.4) and (2.5) it also is easy to verify that

$$(w_1 w_2)^* = w_1^* w_2^*\,. \tag{2.7}$$

Hence by Equations (2.6) and (2.7)

$$|w_1 w_2|^2 = (w_1 w_2)(w_1 w_2)^* = (w_1 w_1^*)(w_2 w_2^*) = |w_1|^2\,|w_2|^2\,,$$

so

$$|w_1 w_2| = |w_1|\,|w_2|\,. \tag{2.8}$$

Equation (2.4) encapsulates the algebra of complex multiplication. But what is the geometry underlying complex multiplication?

To understand the geometry underlying complex multiplication, observe that multiplication by a fixed complex number w is a linear transformation on vectors in the plane, since by the distributive property

$$w(z_1+z_2)=wz_1+wz_2\ .$$

Moreover, multiplication by a complex number of unit length is an isometry (preserves length) because by Equation (2.8)

$$|\, w_1 \,|=1 \Rightarrow |\, w_1 w_2 \,|=|\, w_2 \,|\ .$$

But the only linear isometries in the plane are rotations, so multiplication by a complex number of length one must rotate vectors in the plane.

Let's look at some examples. Consider first multiplication with i. If $v=a+bi$, then

$$iv=i(a+bi)=-b+ai\ .$$

Thus if $v=(a,b)$, then $iv=(-b,a)$, so

$$v\cdot iv=(a,b)\cdot(-b,a)=0\ .$$

Hence, iv is perpendicular to v. Therefore multiplication by i is equivalent to rotation by 90°. Also,

$$(-1)v=-(a+bi)=-a-bi=-v\ ,$$

so multiplication by -1 is equivalent to rotation by 180°. Thus there is indeed a link between rotation and complex multiplication. Let's explore this connection further.

Recall from Chapter 4 that the matrix which rotates vectors in the plane by the angle ϕ is given by:

$$rot(\phi)=\begin{pmatrix}\cos(\phi) & \sin(\phi)\\ -\sin(\phi) & \cos(\phi)\end{pmatrix}.$$

Let $z=(x,y)=x+iy$ be an arbitrary vector in the plane. Then after rotation by the angle ϕ

$$z_{new}=z*rot(\phi)=(x\ \ y)*\begin{pmatrix}\cos(\phi) & \sin(\phi)\\ -\sin(\phi) & \cos(\phi)\end{pmatrix},$$

or equivalently

$$x_{new}=x\cos(\phi)-y\sin(\phi)$$
$$y_{new}=x\sin(\phi)+y\cos(\phi)\ .$$

Now let $w(\phi)$ be the unit vector that makes an angle ϕ with the x-axis (see Figure 1). Then

$$w(\phi)=\cos(\phi)+i\sin(\phi)\ ,$$

and

$$w(\phi)z=\big(\cos(\phi)+i\sin(\phi)\big)\big(x+iy\big)=\left(\underbrace{x\cos(\phi)-y\sin(\phi)}_{x_{new}}\right)+\left(\underbrace{x\sin(\phi)+y\cos(\phi)}_{y_{new}}\right)i\ .$$

Thus multiplying the complex number z by $w(\phi)$ is equivalent to rotating the vector z by the angle ϕ. Moreover, since complex multiplication is associative, we can compose two rotations simply by multiplying the associated complex numbers -- that is, for all z

$$w(\phi+\theta)\, z = w(\phi)\, w(\theta)\, z \ ,$$

so

$$w(\phi+\theta) = w(\phi)\, w(\theta) \, .$$

Representing rotations in the plane by complex numbers is more efficient than matrix representations for rotations in the plane. To store one complex number, we need to store only two real numbers, whereas to store one 2×2 matrix, we need to store four real numbers. To compose two rotations by complex multiplication requires only 4 real multiplications, whereas to compose two rotations by matrix multiplication requires 8 real multiplications.

Rotation in the plane closely resembles exponentiation because composing two rotations is equivalent to adding the corresponding angles of rotation. Since complex multiplication represents rotation, complex multiplication is also closely related to exponentiation. Let z be an arbitrary complex number, and define

$$e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \cdots \ .$$

Then

$$e^{i\phi} = 1 + i\phi + \frac{(i\phi)^2}{2!} + \frac{(i\phi)^3}{3!} + \frac{(i\phi)^4}{4!} + \frac{(i\phi)^5}{5!} \cdots \ .$$

But $i^2 = -1$, so substituting and collecting terms and then recalling the Taylor expansions of sine and cosine, we arrive at the following fundamental identity:

Euler's Formula

$$e^{i\phi} = \left(1 - \frac{\phi^2}{2!} + \frac{\phi^4}{4!} - \cdots\right) + \left(\phi - \frac{\phi^3}{3!} + \frac{\phi^5}{5!} - \cdots\right) i = \cos(\phi) + i\sin(\phi) \, . \qquad (2.9)$$

Since

$$w(\phi) = \cos(\phi) + i\sin(\phi) = e^{i\phi} \, ,$$

we conclude that

$$w(\phi)\, w(\theta) = e^{i\phi} e^{i\theta} = e^{i(\phi+\theta)} = w(\phi+\theta) \ ,$$

so multiplying two complex numbers of unit length is equivalent to adding the angles that these vectors form with the x-axis.

What happens if we multiply a fixed complex number z by an arbitrary complex number w? If $w = r$, where r is a real number, then $wz = rz$, so the effect is to scale the vector z by the constant r. Thus both rotation and scaling can be represented by complex multiplication. Finally, if $w = x + iy$ is an arbitrary complex number, then we can write w in polar form as

$$w = re^{i\theta} = rz(\theta)$$

where $r = |w|$ and $\theta = arctan(y/x)$. Hence

$$wz = rz(\theta)\, z \,,$$

so the effect of multiplying z by w is to rotate z by the angle θ and to scale the result by the constant r. Thus any conformal (angle preserving) linear transformation of vectors in the plane can be represented by multiplication with a single complex number.

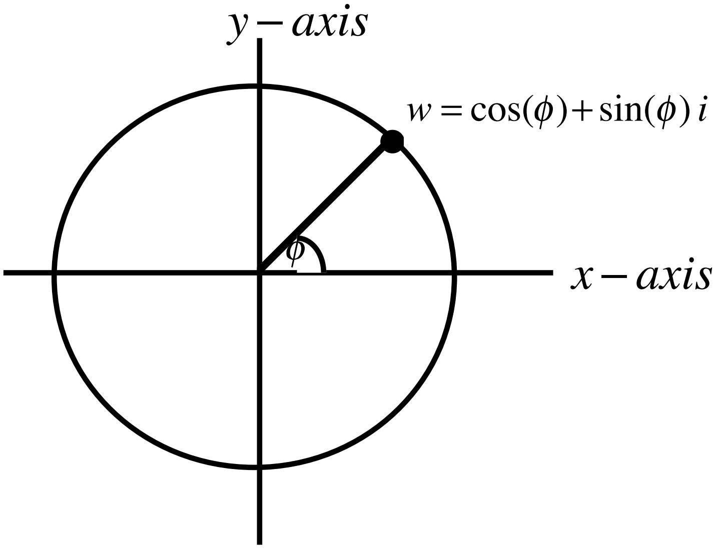


Figure 1: Multiplying a complex number z by a unit vector $w = \cos(\phi) + \sin(\phi)\, i$ is equivalent to rotating the vector z by the angle ϕ.

3. Quaternions

Quaternions are just another name for the entities in the 4-dimensional space of mass-points, just as *complex numbers* are just another name for the entities in the 2-dimensional space of vectors in the plane. In fact, quaternions are an extension of complex numbers to 4-dimensions, so we shall see shortly that we can multiply two quaternions in a manner similar to the way that we can multiply two complex numbers.

3.1 Quaternion Multiplication. To multiply two complex numbers we need to choose a preferred direction: the direction of the identity vector for complex multiplication. Similarly, to multiply two quaternions, we must choose a preferred mass-point in the space of mass-points: the identity for quaternion multiplication. We shall denote this special mass-point by O, and we will identify this mass-point with the point located at the origin, not in the 4-dimensional vector space of mass-points, but rather in the 3-dimensional affine space of affine points. Thus O is a special mass-point with mass $m=1$ (see Figure 2). The choice of the point O is arbitrary, just as the choice of the origin is arbitrary, but to perform quaternion multiplication, we must fix O once and for all. In 4-dimensions the three-dimensional vectors lie in the quaternion subspace orthogonal to O. Thus if we were to introduce rectangular coordinates into the 4-dimensional space of mass-points, then we would have $O=(0,0,0,1)$, and the 3-dimensional vectors v would be written as $v=(v_1,v_2,v_3,0)$ or equivalently $v=v_1 i+v_2 j+v_3 k$, where i,j,k are the unit vectors along the x,y,z coordinate axes.

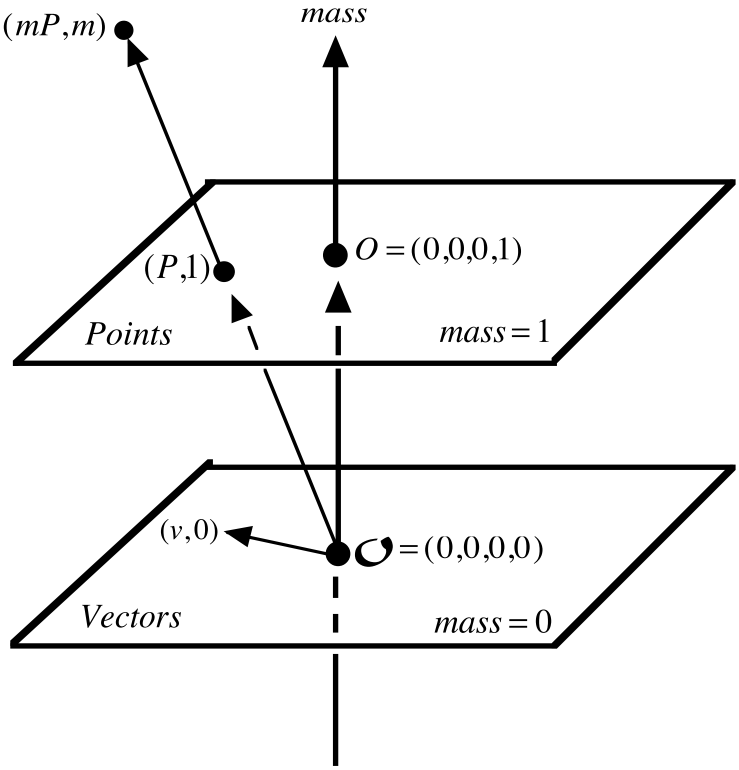


Figure 2: The 4-dimensional vector space of mass-points: three of the dimensions are spatial; the fourth dimension is due to the mass. In this 4-dimensional space, the origin O for the points in 3-dimensions is orthogonal to every vector v in 3-dimensions.

Every mass-point can be written as a scalar mass m times the affine point O plus a vector v. For suppose that (mP,m) is a mass-point with mass m. Then

$$(mP,m)-mO=(mP,m)-(m\,Origin,m)=\big(m(P-Origin),0\big).$$

Setting $v=\big(m(P-Origin),0\big)$, we have

$$(mP,m)=mO+v. \tag{3.1}$$

Interpreted in terms of coordinates, Equation (3.1) means that every quaternion q can be written as $q=(q_1,q_2,q_3,q_4)$ or equivalently

$$q = q_4O + q_1i + q_2j + q_3k\,. \qquad (3.2)$$

In many texts, the letter O is dropped, and quaternions are written as

$$q = q_4 + q_1i + q_2j + q_3k\,. \qquad (3.3)$$

In this notation, a quaternion q seems to be a weird kind of hermaphrodite, a scalar q_4 added to a vector $q_1i + q_2j + q_3k$. In fact, there are no mysterious hermaphrodites; the quaternion q in Equation (3.2) is simply a mass-point with mass q_4.

Quaternion multiplication is an extension of complex multiplication. To extend complex multiplication to the quaternions, we need only extend complex multiplication to the 4-dimensional basis O, i, j, k. The point O is the identity for quaternion multiplication, so

$$O^2 = O. \qquad Oi = i = iO \qquad Oj = j = jO \qquad Ok = k = kO \qquad (3.4)$$

Multiplication on the basis vectors i,j,k is defined by the following rules:

$$i^2 = j^2 = k^2 = -O \quad \text{and} \quad ij = k = -ji, \;\; jk = i = -kj, \;\; ki = j = -ik\,. \qquad (3.5)$$

We then extend quaternion multiplication to two arbitrary quaternions, by assuming that multiplication distributes through addition. For example, for two vectors u,v

$$\begin{aligned} uv &= (u_1i + u_2j + u_3k)(v_1i + v_2j + v_3k) \\ &= u_1i(v_1i + v_2j + v_3k) + u_2j(v_1i + v_2j + v_3k) + u_3k(v_1i + v_2j + v_3k) \\ &= -(u_1v_2 + u_2v_2 + u_3v_3)O + (u_1v_2 - u_2v_1)k + (u_3v_1 - u_1v_3)j + (u_2v_3 - u_3v_3)i \end{aligned}$$

or equivalently

$$uv = -(u \cdot v)O + u \times v \qquad (3.6)$$

More generally for two arbitrary quaternions $aO + u$ and $bO + v$

: $(aO + u)(bO + v) = aO(bO + v) + u(bO + v) = abO + av + bu + uv$

so substituting the right hand side of Equation (3.6) for uv, we find that

$$(aO + u)(bO + v) = (ab - u \cdot v)\,O + (av + bu + u \times v) \qquad (3.7)$$

Equation (3.7) is the general formula for the product of two quaternions. Notice how the vector products -- dot product, cross product, and scalar product -- all appear in the formula for quaternion multiplication. Except for the choice of the special point O, quaternion multiplication is coordinate-free, since dot product, cross product and scalar product are all coordinate-free. Using Equation (3.7) it is straightforward, though somewhat tedious, to verify that quaternion multiplication is indeed associative and distributes through addition (see Exercise 5). Notice, however, that quaternion multiplication is neither commutative nor anticommutative, since

$$v \cdot w = w \cdot v$$

but

$$v \times w = -w \times v\,.$$

By Equation (3.6) the product of two vectors u,v in 3-dimensions is a 4-dimensional mass-point with mass $m=-u\cdot v$. In particular

$$u^2=-(u\cdot u)O=-|u|^2\,O,$$

so

$$u^{-1}=\frac{-u}{|u|^2}.$$

Every non-zero quaternion has a multiplicative inverse. Let $q=m_q O+v_q$. The *conjugate* of *q,* denoted by q^**,* is defined by

$$q^*=m_q O-v_q.$$

By equation (3.7)

$$qq^*=\left(m_q^2+v_q\cdot v_q\right)O=\left(m_q^2+|v|^2\right)O.$$

But recall that in quaternion space $v_q\perp O$, so by the Pythagorean theorem

$$m_q^2+|v_q|^2=|q|^2.$$

Hence

$$qq^*=|q|^2\,O. \qquad (3.8)$$

Therefore

$$q^{-1}=\frac{q^*}{|q|^2}.$$

In terms of coordinates, if $q=(q_1,q_2,q_3,q_4)$, then

$$q=q_4O+q_1i+q_2j+q_3k$$

$$q^*=q_4O-q_1i-q_2j-q_3k$$

$$|q|^2=q_1^2+q_2^2+q_3^2+q_4^2.$$

Using the conjugate, we can prove that the length of the product of two quaternions is equal to the product of their lengths. But first we need the following important result.

Lemma 3.1: $(pq)^*=q^*p^*$ (3.9)

Proof: Let $p=m_pO+v_p$ and $q=m_qO+v_q$. Then by Equation (3.7)

$$pq=(m_pO+v_p)(m_qO+v_q)=(m_pm_q-v_p\cdot v_q)O+(m_qv_p+m_pv_q+v_p\times v_q)$$

$$q^*p^*=(m_qO-v_q)(m_pO-v_p)=(m_qm_p-v_q\cdot v_p)O+(-m_qv_p-m_pv_q+v_q\times v_p).$$

Comparing the right hand sides of these two equations and recalling that the cross product is anticommutative, we conclude that

$$(pq)^* = q^* p^* .$$

Corollary 3.1: $|pq| = |p||q|$ (3.10)

Proof: By Lemma 3.1 and Equation (3.8)

$$|pq|^2 O = (pq)(pq)^* = (pq)(q^* p^*) = p(qq^*)p^* = |p|^2\, |q|^2\, O .$$

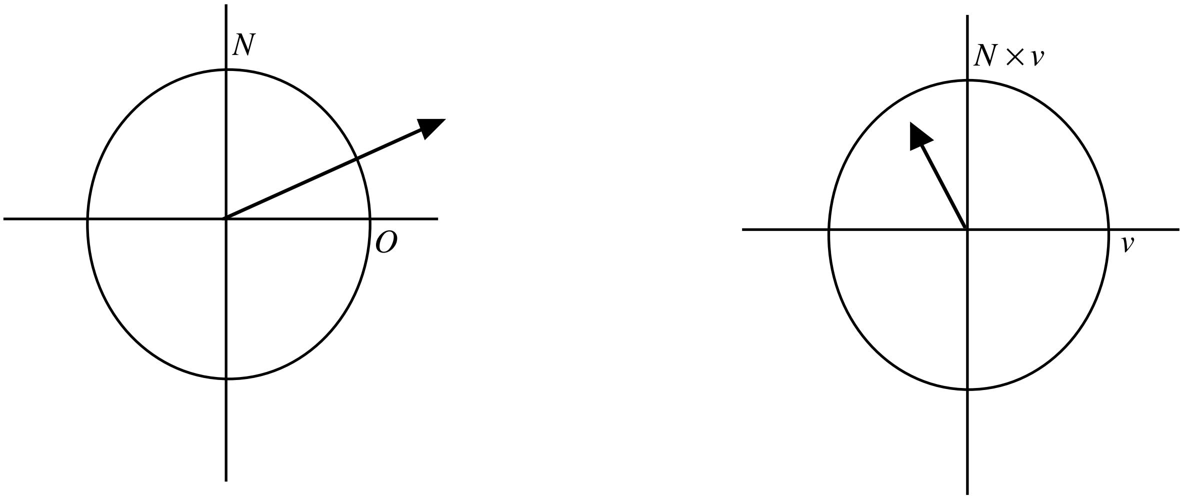
Hence since length is non-negative

$$|pq| = |p||q| .$$

By Corollary 3.1, the length of the product of two quaternions is indeed equal to the product of their lengths. Since quaternion multiplication distributes through addition, multiplication by a unit quaternion is a linear transformation that preserves lengths (a linear isometry). Therefore just as multiplication by a complex number of length one represents a rotation in the plane, we should expect multiplication by a unit quaternion to represent a rotation in 4-dimensions. In Section 3.3 we will use this observation to visualize the geometric effects of quaternion multiplication, but first we need a better way to visualize quaternions in 4-dimensions.

3.2 Mutually Orthogonal Planes in 4-Dimensions. When we represent the 4-dimensional space of mass-points by vectors (arrows) in 4-dimensions, we necessarily have to crop one or two dimensions to draw Figure 2 on a flat page. Thus even though we indicate each mass-point in this figure by four coordinates, at best we can actually see only two or three dimensions. In order to see all four dimensions at the same time, we now introduce another way to visualize quaternions by looking at two mutually orthogonal planes in 4-dimensions.

Let $N = (N, 0)$ be a unit vector in 3-dimensions and let $v = (v, 0)$ be a unit vector in 3-dimensions orthogonal to *N*. Since in the 4-dimensional space of mass-points every vector $v = (v, 0)$ is orthogonal to the origin *O* of the points in affine space (see Figure 2), the two planes in 4-dimensions spanned by O, N and $v, N \times v$ are mutually orthogonal: every mass-point in the plane spanned by O, N is orthogonal to every vector in the plane spanned by $v, N \times v$. Hence every mass-point is the sum of its components in these two mutually orthogonal planes. Therefore we can visualize simultaneously all four dimensions of any mass-point by modeling its projections into these two mutually orthogonal planes (see Figure 3).



(a) the plane of O, N

(b) the plane perpendicular to O, N

Figure 3: A mass-point represented by its projections (arrows) into two mutually orthogonal planes in 4-dimensions.

Although Figure 3 depicts two planes in 4-dimensions, the geometric interpretations of these two planes in terms of mass-points are markedly different: the plane perpendicular to O, N represents *a plane of vectors* in 3-dimensions, but in 3-dimensions the plane of O, N represents *a line of points*.

Consider first the plane perpendicular to O, N. Since the vectors v and $N \times v$ are orthogonal, the plane in 4-dimensions of the vectors

$$w = a\,v + b\,N \times v$$

spanned by $v, N \times v$ represents the plane of 3-dimensional vectors perpendicular to N (see Figure 4).

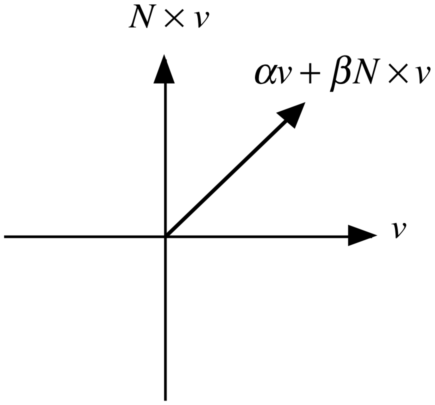


Figure 4: The plane of vectors perpendicular to O, N in 4-dimensions is the plane of vectors in 3-

dimensions perpendicular to N. This plane is spanned by the vectors v and $N \times v$, where v is any nonzero vector in 3-dimensions perpendicular to N.

On the other hand, consider the plane of O, N. In 3-dimensions O represents a point, not a vector. Thus the plane of 4-dimensional vectors

$$P = cO + sN$$

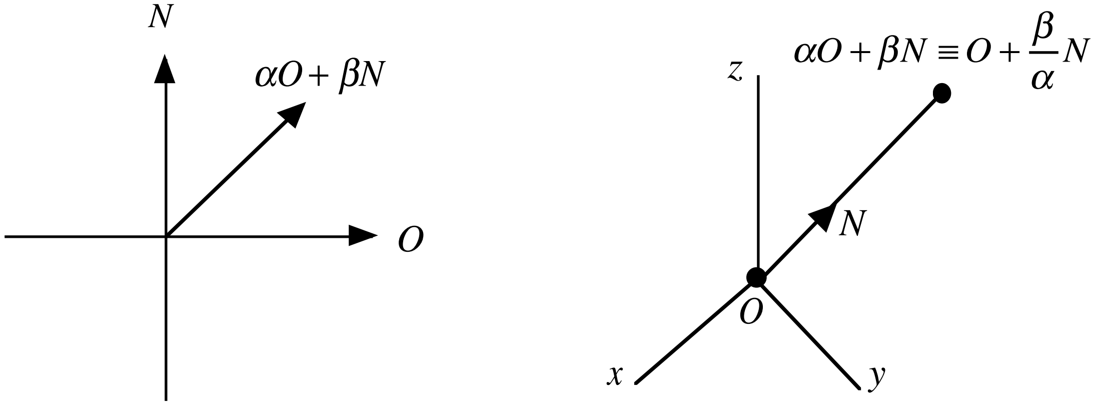
spanned by O, N actually represents a line in 3-dimensions: the line through the affine point O in the direction of the unit vector N. We shall write

$$(mP, m) \equiv (P, 1)$$

to denote that (mP, m) and $(P, 1)$ have the same location. Thus

$$P \equiv O + \frac{s}{c} N$$

-- that is, P is a mass-point with mass $m = c$ on the line $P(t) = O + tN$. (If $c = 0$, then P is a vector sN parallel to the line $P(t)$.) The plane spanned by O, N does have two dimensions, but only one dimension is spatial; the other dimension, the coefficient of O, represents mass not length (see Figure 5).



(a) a plane of vectors in 4-dimensions (b) a line of points in 3-dimensions.

Figure 5: (a) The plane of vectors (mass-points) in 4-dimensions spanned by O, N is equivalent to (b) the line of affine points in 3-dimensions passing through the point O in the direction of the vector N. This line is actually 2-dimensional, but the second dimension is mass-like not spatial, so this dimension is not visible in (b).

In addition to this geometric distinction, there is also an algebraic distinction between these two mutually orthogonal planes in 4-dimensions. We shall see in Section 3.3 that the plane of O, N is isomorphic algebraically to the complex numbers $\boldsymbol{C}$, whereas the plane perpendicular to O, N is isomorphic not to $\boldsymbol{C}$ but to $\boldsymbol{C}v$, where v is any unit vector perpendicular to N. We are going to take advantage of these asymmetries in the algebraic and geometric properties of these two mutually orthogonal planes in 4-dimensions in Section 3.4 when we study conformal transformations on points and vectors in 3-dimensions based on quaternion multiplication.

3.3 The Geometry of Quaternion Multiplication. We want to see what happens when we multiply an arbitrary quaternion p by a unit quaternion

$$q(N,\theta) = \cos(\theta)\,O + \sin(\theta)N \ ,$$

where N is a unit vector and O is the identity for quaternion multiplication. There is one key algebraic insight that will allow us to visualize the geometric effects of this quaternion multiplication: *For every unit vector N, the plane of O, N is isomorphic to the complex plane*.

Indeed by Equation (3.6) for any unit vector N, we have

$$N^2 = -O \ .$$

Since

$$O^2 = O \ , \ ON = NO = N \ , \text{ and } \ N \perp O \ ,$$

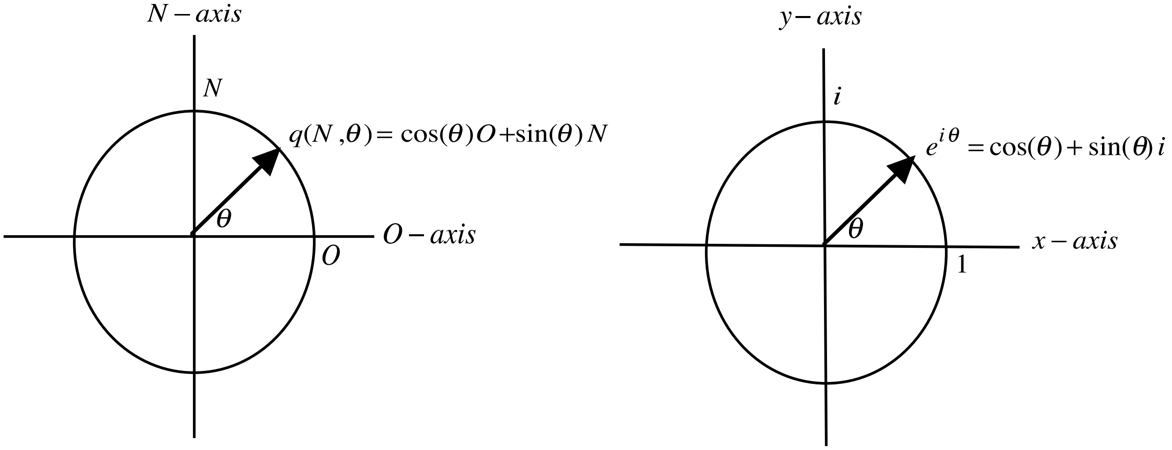
multiplication in the plane determined by the two quaternions O,N behaves exactly like multiplication in the complex plane. Therefore, just as multiplication by the complex number

$$e^{i\theta} = \cos(\theta) + \sin(\theta)\,i$$

rotates vectors in the complex plane by the angle θ (see Section 2), multiplication by the unit quaternion

$$q(N,\theta) = \cos(\theta)\,O + \sin(\theta)N$$

rotates quaternions in the plane of O,N by the angle θ (see Figure 6).


$N-axis$
N
$q(N,\theta) = \cos(\theta)O + \sin(\theta)N$
θ
$O-axis$
O
$y-axis$
i
$e^{i\theta} = \cos(\theta) + \sin(\theta)\,i$
θ
$x-axis$
1

(a) the quaternion plane of O, N *(b) the complex plane*

Figure 6: For any unit vector *N*: (a) the plane determined by *O, N* is isomorphic algebraically and homeomorphic geometrically to (b) the complex plane: $O \leftrightarrow 1$ and $N \leftrightarrow i$, since $O^2 = O$, $N^2 = -O$, $O\,N = N\,O = N$, and $N \perp O$. Hence

$$q(N,\theta) = \cos(\theta)\, O + \sin(\theta)\, N \leftrightarrow e^{i\theta} = \cos(\theta) + \sin(\theta)\, i \,\mathbf{.}$$

Therefore we have the following results.

Lemma 3.2: *Let N be a unit vector. Then*

i. $q(N,\theta)\; q(N,\phi) = q(N,\theta+\phi)$

ii. $q(N,\theta)\; q(N,\phi) = q(N,\phi)\; q(N,\theta)$

Proof: To prove *i*, we simply apply quaternion multiplication -- Equation (3.7) -- and the trigonometric identities for the sine and the cosine of the sum of two angles:

$$\begin{aligned} i.\;\; q(N,\theta)\, q(N,\phi) &= \big(\cos(\theta)O + \sin(\theta)\, N\big)\big(\cos(\phi)O + \sin(\phi)N\big) \\ &= \big(\cos(\theta)\cos(\phi) - \sin(\theta)\sin(\phi)\big)O + \big(\sin(\theta)\cos(\phi) + \cos(\theta)\sin(\phi)\big)N \\ &= \cos(\theta+\phi)O + \sin(\theta+\phi)N \\ &= q(N,\theta+\phi)\,. \end{aligned}$$

ii. Follows immediately from *i*.

Since multiplying arbitrary quaternions by a fixed unit quaternion is a linear isometry in 4-dimensions, unit quaternions $q(N,\theta)$ represent rotations in 4-dimensions. By Lemma 3.2 we understand how $q(N,\theta)$ rotates quaternions in the plane determined by O,N because in this plane multiplication by $q(N,\theta)$ behaves just like multiplication by $e^{i\theta}$ in the complex plane. But to fully visualize how multiplication by $q(N,\theta)$ affects arbitrary quaternions, we still need to see how $q(N,\theta)$ affects quaternions in the complementary plane of quaternions perpendicular to O,N .

In facilitate our presentation, we now introduce the following notation:

$L_q(p) = q\,p$ left multiplication by q

$R_q(p) = p\,q$ right multiplication by q

$S_q(p) = q\,p\,q^* = L_q\big(R_{q^*}(p)\big)$ sandwiching *p* between q and q^* .

The functions $L_q(p)$ and $R_q(p)$ are linear transformations on the vector space of quaternions because quaternion multiplication distributes through quaternion addition. The sandwiching operator $S_q(p)$ is also a linear transformations because $S_q(p)$ is the composite of two linear transformations. We shall see in Section 3.4 that the sandwiching operator $S_q(p)$ plays a central role in the use of quaternions for computing rotations in 3-dimensions.

We are now ready to study the geometric effects of multiplication by the unit quaternions

$$q(N,\theta) = \cos(\theta)\, O + \sin(\theta) N$$
$$q^*(N,\theta) = \cos(\theta) O - \sin(\theta) N = q(N,-\theta) \qquad (3.11)$$

on quaternions in the plane of *O, N* -- which we have already seen is isomorphic to the complex plane -- and on quaternions in the complementary plane of vectors perpendicular to *O, N*.

Proposition 3.2: Rotation in the Plane of *O, N*

Let

- *N = a unit vector*
- *p = a quaternion in the plane of O, N*

Then

i. $L_{q(N,\theta)}(p) = q(N,\theta)\, p$ *rotates p by the angle* θ *in the plane of O, N*

ii. $R_{q(N,\theta)}(p) = p\, q(N,\theta)$ *rotates p by the angle* θ *in the plane of O, N*

iii. $R_{q^*(N,\theta)} p = p\, q^*(N,\theta)$ *rotates p by the angle* $-\theta$ *in the plane of O, N*

iv. $S_{q(N,\theta)}(p) = q(N,\ \theta)\ p\ q^*(N,\ \theta)$ *is the identity on p.*

Proof: *i, ii*. Follow immediately from Lemma 3.2..

iii. Follows immediately from *ii* and Equation (3.11)

iv. Follows immediately from *i* and *iii*.

Proposition 3.3: Rotation in the Plane Perpendicular to *O, N*

Let

- *N = a unit vector*
- *v = a vector in the plane* $\perp$ *O, N*

Then

i. $L_{q(N,\theta)}(v) = q(N,\theta)\, v$ *rotates v by the angle* θ *in the plane* $\perp$ *O, N*

ii. $R_{q(N,\theta)}(v) = v\, q(N,\theta)$ *rotates v by the angle* $-\theta$ *in the plane* $\perp$ *O, N*

iii. $R_{q^*(N,\theta)}(v) = v\, q^*(N,\theta)$ *rotates v by the angle* θ *in the plane* $\perp$ *O, N*

iv. $S_{q(N,\theta)}(v) = q(N,\theta)\ v\ q^*(N,\theta)$ *rotates v by the angle* 2θ *in the plane* $\perp$ *O, N.*

Proof: Since $v \perp N$, it follows that $v \cdot N = 0$. Hence, by Equation (3.6), $N v = N \times v$ and $v N = v \times N$. Therefore:

i. $q(N,\theta)\, v = \big(\cos(\theta) O + \sin(\theta) N\big) v = \cos(\theta) v + \sin(\theta) N \times v$

ii. $v\, q(N,\theta) = v\big(\cos(\theta) O + \sin(\theta) N\big) = \cos(\theta) v + \sin(\theta) v \times N = \cos(\theta) v - \sin(\theta) N \times v$

iii. $v\, q^{*}(N,\theta) = v\big(\cos(\theta) O - \sin(\theta) N\big) = \cos(\theta) v - \sin(\theta) v \times N = \cos(\theta) v + \sin(\theta) N \times v$

Equations *i* and *iii* represent rotation by the angle θ and Equation *ii* represents rotation by the angle $-\theta$ in the plane perpendicular to *O, N* (see Figure 7). Now it follows easily from *i* and *iii* that multiplication by $q(N,\theta)$ on the left and multiplication by $q^{*}(N,\theta)$ on the right both rotate vectors *v* in the plane perpendicular to *O, N* by the same angle θ. Therefore

iv. $q(N,\theta)\; v\; q^{*}(N,\theta)$ rotates *v* by the angle 2θ in the plane $\perp$ *O, N*.

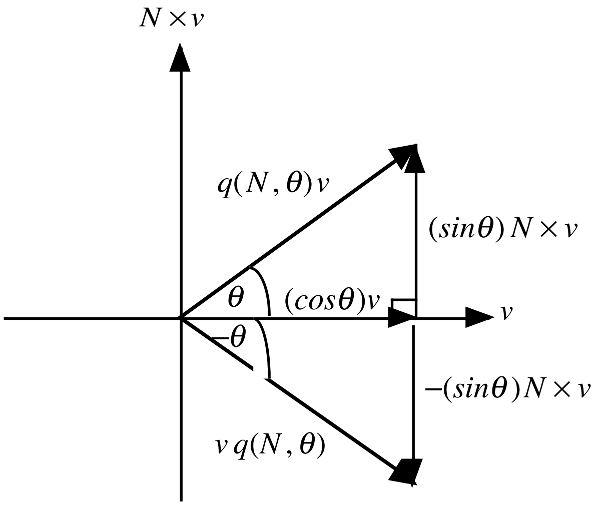
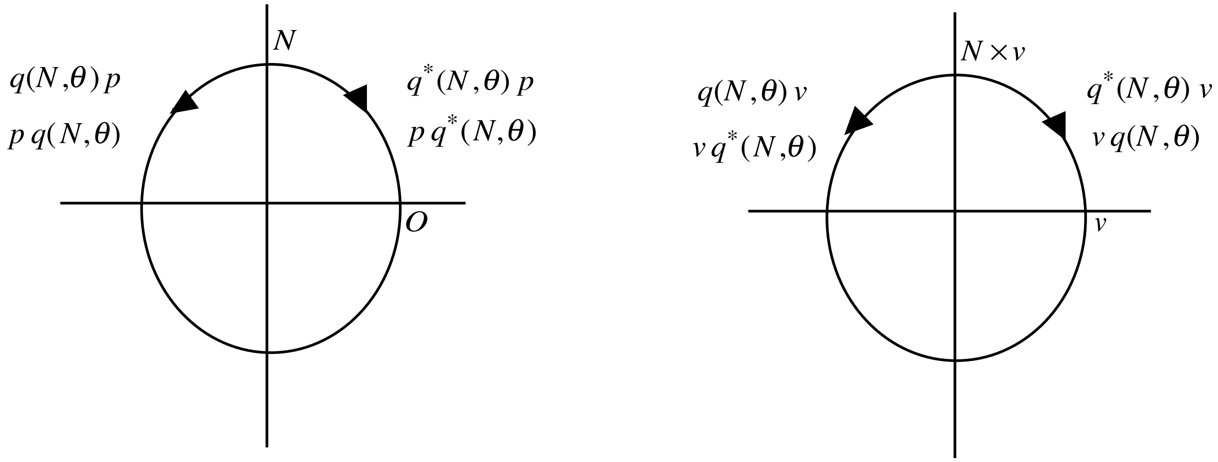


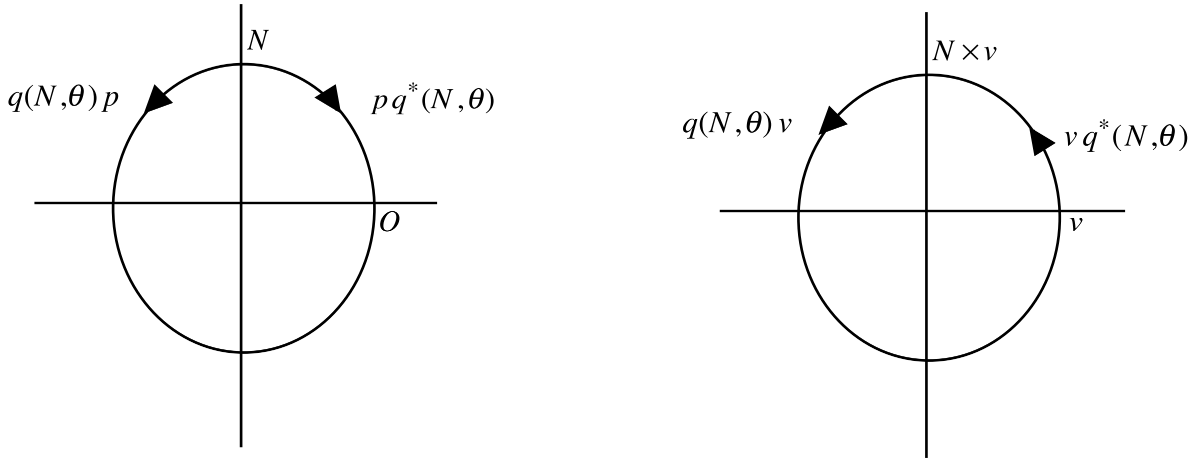
Figure 7: Rotation by the angle ±θ in the plane $\perp$ *O, N*.

Propositions 3.2 are 3.3 are summarized graphically in Figure 8 and 9. These two figures encapsulate geometrically the main properties of quaternion multiplication presented in this section.



(a) the plane of O, N *(b) the plane perpendicular to O, N*

Figure 8: The effects of multiplying a quaternion on the left and the right with $q(N,\theta)$ (or $q^*(N,\theta)$) in two mutually orthogonal planes. Multiplying with $q(N,\theta)$ on the left rotates quaternions in both planes counterclockwise by the angle θ; multiplying with $q(N,\theta)$ on the right rotates quaternions in the plane of *O, N* counterclockwise by the angle θ and rotates quaternions in the plane perpendicular to *O,N* clockwise by the angle θ Complementary results hold when $q(N,\theta)$ is replaced by $q^*(N,\theta)$.



(a) the plane of O, N *(b) the plane perpendicular to O, N*

Figure 9: The effects of sandwiching a quaternion between $q(N,\theta)$ and $q^*(N,\theta)$ in two mutually orthogonal planes. In (a) the plane of *O, N*, the rotation induced by multiplying with $q(N,\theta)$ on the left is canceled by the rotation induced by multiplying with its conjugate $q^*(N,\theta)$ on the right, whereas in (b) the plane of vectors perpendicular to the plane of *O, N,* the rotation induced by

multiplying with $q(N,\theta)$ on the left is reinforced by the rotation induced by multiplying with its conjugate $q^*(N,\theta)$ on the right.

With this introduction to quaternion algebra and geometry in 4-dimensions, we are finally ready to turn our attention to the main applications of quaternion multiplication in Computer Graphics: computing conformal transformations on points and vectors in 3-dimensions.

3.4 Quaternion Representations for Conformal Transformations. Quaternion multiplication can be used to model rotations of vectors in 3-dimensions much like complex multiplication can be used to model rotations of vectors in the plane. In fact, quaternion multiplication can be used to model any conformal transformation -- rotation, mirror image, and uniform scaling -- of vectors in 3-dimensions.

The key to modeling conformal transformations with quaternions is the sandwiching map

$$S_q(v) = q\, v\, q^* .$$

Notice that

$$(S_p \circ S_q)(v) = S_{pq}(v) \qquad (3.12)$$

because by Lemma 3.1

$$(S_p \circ S_q)(v) = S_p\big(S_q(v)\big) = S_p\big(q\, v\, q^*\big) = p\, q\, v\, q^* p^* = (p\, q)\, v\, (pq)^* = S_{pq}(v) .$$

Proposition 3.4: *Let* $q(N,\phi) = \cos(\phi/2)O + \sin(\phi/2)N$. *Then a vector v can be rotated around the axis N through the angle* ϕ *by sandwiching v between* $q(N,\phi)$ *and* $q(N,\phi)^*$.

Proof: Let $v = v_{||} + v_\perp$, where

- $v_{||} =$ component of *v* parallel to *N*
- $v_\perp =$ component of *v* perpendicular to *N*.

Then

- $S_{q(N,\theta/2)}(v_{||}) = v_{||}$ (Proposition 3.2)
- $S_{q(N\theta/2)}(v_\perp)$ *rotates* $v_\perp$ *by the angle* θ *in the plane* $\perp$ *to O, N.* (Proposition 3.3)

Therefore $S_{q(N,\theta/2)}(v)$ has the same effect on the components of *v* as rotating *v* in 3-dimensions by the angle θ around the axis vector *N* (see Chapter 12, Section 12.3.2).

Notice that the quaternion $q(N,\phi)$ is a unit quaternion because by Equation (3.8)

$$|q(N,\phi)|^2 O = q(N,\phi)q(N,\phi)^* = \left(\cos^2(\phi/2)+\sin^2(\phi/2)\right)O = O .$$

Thus every rotation is represented by a unit quaternion. Conversely, every unit quaternion represents a rotation because if $q = m_q O + v_q$ is a unit quaternion, then $m_q^2 + |v_q|^2 = 1$. Hence there is an angle θ and a unit vector N such that $m_q = \cos(\theta)$ and $v_q = \sin(\theta)N$, so $q = \cos(\theta)O + \sin(\theta)N = q(N,\theta)$.

To model mirror image with sandwiching, observe that to find the mirror image of a vector *v* in the plane perpendicular to the unit normal vector *N*, we can rotate *v* by 180^0 around *N* and then negate the result. Now

$$q(N,\pi/2) = \cos(\pi/2)O + \sin(\pi/2)N = N .$$

Thus to rotate *v* by 180^0 around *N,* we can sandwich *v* between *N* and N^*. Since for vectors $N^* = -N$, it follows that mirror image of *v* in the plane perpendicular to *N* is given by

$$-S_N(v) = NvN .$$

Thus we are led to the following result.

Proposition 3.5: *A vector v can be mirrored in the plane perpendicular to the unit normal N by sandwiching v between two copies of N.*

Proof: Let $v = v_{||} + v_{\perp}$, where

- $v_{||} =$ component of *v* parallel to *N*
- $v_{\perp} =$ component of *v* perpendicular to *N.*

Since $N = \cos(\pi/2)O + \sin(\pi/2)N = q(N,\pi/2)$:

- $-S_N(v_{||}) = -S_{q(N,\pi/2)}(v_{||}) = -v_{||}$ (Proposition 3.2)
- $-S_N(v_{\perp}) = -S_{q(N,\pi/2)}(v_{\perp}) = v_{\perp}$ (Proposition 3.3)

Hence sandwiching *v* between two copies of *N* has the same effect on the components of *v* as mirroring *v* in 3-dimensions in the plane perpendicular to *N* (see Chapter 12, Section 12.3.3).

Proposition 3.6: *A vector v can be scaled by the factor c by sandwiching v between two copies of* $\sqrt{c}\,O$.

Proof: $S_{\sqrt{c}\,O}(v) = \sqrt{c}\,O\,v\,\sqrt{c}\,O = c\,v$.

Theorem 3.1: *Every conformal transformation of vectors in 3-dimensions can be modeled by sandwiching with a single quaternion.*
Proof: This result follows from Propositions 3.4-3.6, since by Equation (3.12) composition of sandwiching is equivalent to multiplication of quaternions.

Conformal transformations of vectors in 3-dimensions can be computed by sandwiching the vectors with the appropriate quaternions. But what about conformal transformations on points in 3-dimensions? To rotate points about arbitrary lines, or to mirror points in arbitrary planes, or to scale points uniformly about arbitrary points, we cannot simply sandwich the points we wish to transform with some appropriate quaternion (see Exercises 11 and 15). Recall, however, that rotation, mirror image, and uniform scaling, all have a fixed point Q of the transformation. Since $P = Q + (P - Q)$, we can compute conformal transformations on affine points P by using quaternions to transform the vectors $P - Q$ and then adding the resulting vectors to Q. In this way, quaternions can be applied to compute conformal transformations on points as well as on vectors.

3.5 Quaternions vs. Matrices. We now have two ways to represent conformal transformations on vectors in 3-dimensions: 3×3 matrices and quaternions. Here we shall compare and contrast some of the advantages and disadvantages of quaternion and matrix representations for conformal transformations.

A 3×3 matrix contains 9 scalar entries, whereas a quaternion can be represented by 4 rectangular coordinates. Thus if memory is at a premium, quaternions are more than twice as efficient as matrices. On the other hand, by Equation (3.7) to compute the product of two quaternions requires 16 real scalar multiplications: 6 for the cross product, 3 for the dot product, 6 for the two scalar products, and one for the product of the two scalar masses. Similarly, since a vector has zero mass, the product of a quaternion and a vector requires 12 scalar multiplications, since the product of the masses as well as one of the scalar products is zero. Sandwiching a vector with a quaternion computes a product of a quaternion with a vector followed by the product of a quaternion with a quaternion, so the total cost for one conformal transformation using quaternions is $12 + 16 = 28$ scalar multiplications, whereas the cost for multiplying a vector by a 3×3 matrix is only 9 scalar multiplications. Thus matrices are more than 3 times as fast as quaternions for computing arbitrary conformal transformations. But there is also an additional computational consideration. To compose two transformations by quaternion multiplication requires 16 scalar multiplications, whereas to compose two transformations by matrix multiplication requires 27 scalar multiplications. Therefore quaternion multiplication is more than 1.5 times faster than matrix multiplication for composing conformal transformations. Thus some computations favor quaternions, whereas others favor matrices. We summarize these tradeoffs for both memory and speed in Table 1.

	memory	*transformation*	*composition*
quaternions	4 scalars	28 multiplies	16 multiplies
3×3 matrices	9 scalars	9 multiplies	27 multiplies

Table 1: Tradeoffs between speed and memory for quaternions and 3×3 matrices.

3.6 Avoiding Distortion. Aside from considerations of memory and speed, one of the main advantages of quaternions over matrices is avoiding distortions that arise from numerical inaccuracies that are inevitably introduced by floating point computations.

For any 3×3 matrix A representing an affine transformation,

$$\begin{pmatrix} i_{new} \\ j_{new} \\ k_{new} \end{pmatrix} = \begin{pmatrix} i \\ j \\ k \end{pmatrix} * A .$$

Since the matrix with rows i,j,k is the identity matrix, we conclude that

$$A = \begin{pmatrix} i_{new} \\ j_{new} \\ k_{new} \end{pmatrix}. \tag{3.13}$$

Hence the rows of A represent the images of the unit vectors i,j,k along the coordinate axes.

Thus if a 3×3 matrix A represents a conformal transformation, then the rows of A must be mutually orthogonal vectors, since conformal transformations preserve angles and the original vectors i,j,k are mutually orthogonal. To compose conformal transformations, we must multiply matrices. But the entries of these matrices -- especially rotation matrices which contain values of sines and cosines -- are typically represented by floating point numbers. Therefore after composing several conformal transformations, the rows of the product matrix will no longer be mutually orthogonal vectors due to numerical inaccuracies introduced by floating point computations. Thus these matrices no longer represent conformal transformations, so applying these matrices will distort angles in the image.

Quaternions avoid this problem. To compose conformal transformations, we multiply the corresponding quaternions. Since every quaternion represents a conformal transformation, computing transformations by sandwiching with quaternions will never distort angles.

Suppose, however, that we want to compose only rotations. Composing rotations by matrix multiplication will generate distortions in both lengths and angles due to numerical inaccuracies introduced by floating point computations. We need to renormalize the resulting matrix so that the rows are mutually orthogonal unit vectors, but it is not clear how to perform this normalization.

This normalization, however, is easy to perform with quaternions. Rotations are represented by unit quaternions. When we compose rotations by multiplying unit quaternions, the result may no longer be a unit quaternion due to floating point computations. But we can normalize the resulting quaternion q simply by dividing by its length $|q|$. The resulting quaternion $q/|q|$ is certainly a unit quaternion, so this normalization avoids distorting either lengths or angles in the image.

3.7 Key Frame Animation. Quaternions have one additional advantage over matrices: quaternions can be used to interpolate in between frames for key frame animation. In key frame animation, an artist draws only a few key frames in the scene, and an animator must interpolate intermediate frames to make the animation look natural.

Consider an object tumbling through a scene. An artist will typically draw only a few frames representing rotations of the object at certain key times. An animator must then find intermediate rotations, so that the tumbling appears smooth. Suppose that from the artist's drawing we know the rotations R_0 and R_1 at times $t=0$ and $t=1$. How do we find the appropriate intermediate rotations at intermediate times?

If the rotations R_0, R_1 are represented by 3×3 matrices, then we seek intermediate rotation matrices $R(t)$ for $0 < t < 1$. We might try linear interpolation and set

$$R(t) = (1-t)R_0 + tR_1 .$$

The problem with this approach is that the matrices $R(t)$ are no longer rotation matrices because the rows of these matrices are no longer mutually orthogonal unit vectors. Thus the matrices $R(t)$ will introduce undesirable distortions into the animation.

Alternatively, we could use unit quaternions q_0, q_1 to represent these two rotations. The quaternion generated by linear interpolation

$$q(t) = (1-t)q_0 + tq_1$$

is no longer a unit quaternion, but we can adjust the length of $q(t)$ by dividing by $|q(t)|$. Now $q(t)/|q(t)|$ is a unit quaternion representing an intermediate rotation. Unfortunately, the tumbling motions generated by this approach will not appear smooth because linear interpolation does not generate uniformly spaced quaternions -- that is, if ϕ is the angle between q_0 and q_1, then $t\phi$ is not generally the angle between q_0 and $q(t)/|q(t)|$.

To correct for this problem, we apply spherical linear interpolation (SLERP, see Chapter 11, Section 11.6) -- that is, we set

$$q(t) = slerp(q_0, q_1, t) = \frac{\sin((1-t)\phi)}{\sin(\phi)} q_0 + \frac{\sin(t\phi)}{\sin(\phi)} q_1 , \qquad (3.14)$$

where ϕ is the angle between q_0 and q_1. Recall that SLERP guarantees that if q_0 and q_1 are unit quaternions, then $q(t)$ is also a unit quaternion, and if ϕ is the angle between q_0 and q_1, then $t\phi$ is the angle between q_0 and $q(t)$. Thus spherical linear interpolation applied to quaternions generates the appropriate intermediate rotations for key frame animation.

3.8 Conversion Formulas. Since we have two ways to represent conformal transformations on vectors in 3-dimensions -- 3×3 matrices and quaternions -- and since each of these representations is superior for certain applications, it is natural to develop formulas to convert between these two representations. If a quaternion represents scaling along with rotation, the scale factor can be retrieved from the magnitude of the quaternion. Similarly if a 3×3 matrix is a composite of rotation and uniform scaling, the scale factor can be retrieved from the determinant of the matrix. Therefore we can easily retrieve these scale factors, and we can always normalize quaternions to have unit length and conformal matrices to have unit determinant. Unit quaternions and 3×3 conformal matrices with unit determinant represent rotations. Since, as we have just seen in Sections 3.3 and 3.4, rotations embody the most important applications of quaternions to Computer Graphics, we will concentrate here on converting between different representations for rotations.

Consider a rotation around the unit axis vector N by the angle ϕ. The rotation matrix (Chapter 13, Section 13.3.2) and the unit quaternion (Section 3.2, Proposition 2) that represent this rotation are given by

$$Rot(N,\phi) = (\cos\phi)\,I + (1-\cos\phi)(N^T * N) + (\sin\phi)(N\times\_) \qquad (3.15)$$

$$q(N,\phi/2) = \cos(\phi/2)O + \sin(\phi/2)N\,. \qquad (3.16)$$

To convert between these two representations for rotation, we shall make extensive use of the following half angle and double angle formulas for sine and cosine (see Exercise 3):

$$\cos^2(\phi/2) = \frac{1+\cos\phi}{2}$$

$$\sin^2(\phi/2) = \frac{1-\cos\phi}{2}$$

$$sin(\phi) = 2\,sin(\phi/2)\cos(\phi/2)$$

Suppose that we are given the rotation matrix

$$Rot(N,\phi) = (\cos\phi)\,I + (1-\cos\phi)(N^T * N) + (\sin\phi)(N\times\_) = (R_{i,j})$$

and we want to find the corresponding unit quaternion

$$q(N,\phi/2) = \cos(\phi/2)O + \sin(\phi/2)N\,.$$

Then we need to find $\cos(\phi/2)$ and $\sin(\phi/2)N$. Expanding Equation (3.15) in matrix form yields:

$$Rot(N,\phi) = \begin{pmatrix} \cos\phi + (1-\cos\phi)N_1^2 & (1-\cos\phi)N_1N_2 + \sin\phi\, N_3 & (1-\cos\phi)N_1N_3 - \sin\phi N_2 \\ (1-\cos\phi)N_1N_2 - \sin\phi\, N_3 & \cos\phi + (1-\cos\phi)\, N_2^2 & (1-\cos\phi)\, N_2N_3 + \sin\phi\, N_1 \\ (1-\cos\phi)N_1N_3 + \sin\phi\, N_2 & (1-\cos\phi)N_2N_3 - \sin\phi\, N_1 & \cos\phi + (1-\cos\phi)N_3^2 \end{pmatrix} . \tag{3.17}$$

Since N is a unit vector

$$N_1^2 + N_2^2 + N_3^2 = N \cdot N = 1 .$$

Therefore summing the diagonal entries of $Rot(N,\phi)$ gives

$$R_{1,1} + R_{2,2} + R_{3,3} = 3\cos\phi + 1 - \cos\phi = 1 + 2\cos\phi .$$

Solving for $\cos\phi$, we find that

$$\cos\phi = \frac{R_{1,1} + R_{2,2} + R_{3,3} - 1}{2} . \tag{3.18}$$

But by the half angle formula

$$\cos(\phi/2) = \sqrt{\frac{1+\cos\phi}{2}} .$$

Therefore by Equation (3.18)

$$\cos(\phi/2) = \frac{\sqrt{R_{1,1} + R_{2,2} + R_{3,3} + 1}}{2} . \tag{3.19}$$

To find $\sin(\phi/2)N$, observe from Equation (3.17) that

$$2sin(\phi)N = \left(R_{2,3} - R_{3,2}, R_{3,1} - R_{1,3}, R_{1,2} - R_{2,1}\right) . \tag{3.20}$$

But

$$sin(\phi) = 2\,sin(\phi/2)\cos(\phi/2) .$$

Substituting this result into Equation (3.20) and solving for $\sin(\phi/2)N$ yields

$$sin(\phi/2)N = \frac{\left(R_{2,3} - R_{3,2}, R_{3,1} - R_{1,3}, R_{1,2} - R_{2,1}\right)}{4\,cos(\phi/2)} . \tag{3.21}$$

Thus by Equations (3.19) and (3.21)

$$q(N,\phi/2) = \frac{\sqrt{R_{1,1} + R_{2,2} + R_{3,3} + 1}}{2} O + \frac{(R_{2,3} - R_{3,2})i + (R_{3,1} - R_{1,3})j + (R_{1,2} - R_{2,1})k}{2\sqrt{R_{1,1} + R_{2,2} + R_{3,3} + 1}} . \tag{3.22}$$

Conversely, suppose that we are given the quaternion

$$q(N,\phi/2) = \cos(\phi/2)O + \sin(\phi/2)N = (q_1, q_2, q_3, q_4) \tag{3.23}$$

and we want to find the matrix

$$Rot(N,\phi) = (\cos\phi)\, I + (1-\cos\phi)(N^T * N) + (\sin\phi)(N \times \_) .$$

Then we need to compute the three matrices $(\cos\phi)I, (1-\cos\phi)(N^T * N), (\sin\phi)(N \times \_)$. By the half angle formula

$$\cos(\phi) = 2\cos^2(\phi/2) - 1 .$$

Thus by Equation (3.23), since $q(N,\phi/2)$ is a unit quaternion,

$$\cos(\phi) = 2q_4^2 - 1 = q_4^2 - q_1^2 - q_2^2 - q_3^2 ,$$

so

$$\cos(\phi)I = \begin{pmatrix} q_4^2 - q_1^2 - q_2^2 - q_3^2 & 0 & 0 \\ 0 & q_4^2 - q_1^2 - q_2^2 - q_3^2 & 0 \\ 0 & 0 & q_4^2 - q_1^2 - q_2^2 - q_3^2 \end{pmatrix}. \qquad (3.24)$$

Moreover, since by the half angle formula

$$\sin^2(\phi/2) = \frac{1-\cos(\phi)}{2},$$

it follows that

$$(1-\cos\phi)(N^T * N) = 2\sin(\phi/2)N^T * \sin(\phi/2)N .$$

Hence by Equation (3.23)

$$(1-\cos\phi)(N^T * N) = 2(q_1,q_2,q_3)^T * (q_1,q_2,q_3) ,$$

so

$$(1-\cos\phi)(N^T * N) = \begin{pmatrix} 2q_1^2 & 2q_1q_2 & 2q_1q_3 \\ 2q_1q_2 & 2q_2^2 & 2q_2q_3 \\ 2q_1q_3 & 2q_2q_3 & 2q_3^2 \end{pmatrix}. \qquad (3.25)$$

Finally since

$$sin(\phi) = 2sin(\phi/2)\cos(\phi/2) ,$$

we have

$$(\sin\phi)N = 2sin(\phi/2)\cos(\phi/2)N .$$

Hence again by Equation (3.23)

$$(\sin\phi)N = 2q_4(q_1,q_2,q_3) ,$$

so

$$(\sin\phi)N \times \_ = \begin{pmatrix} 0 & 2q_4q_3 & -2q_4q_2 \\ -2q_4q_3 & 0 & 2q_4q_1 \\ 2q_4q_2 & -2q_4q_1 & 0 \end{pmatrix}. \qquad (3.26)$$

Adding together Equations (3.24), (3.25), (3.26) yields

$$Rot(N,\phi) = \begin{pmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_3q_4 & 2q_1q_3 - 2q_2q_4 \\ 2q_1q_2 - 2q_3q_4 & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_1q_4 \\ 2q_1q_3 + 2q_2q_4 & 2q_2q_3 - 2q_1q_4 & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \qquad (3.27)$$

4. Summary

Complex numbers can be used to represent vectors in the plane. A vector $z = (x,y)$ in the xy-plane can be rotated through the angle ϕ by multiplying the complex number $z = x + i\,y$ with the complex number $w(\phi) = e^{i\phi} = \cos(\phi) + i\sin(\phi)$, which corresponds to the unit vector that makes an angle ϕ with the x-axis. Similarly, the vector $z = (x,y)$ can be scaled by the factor s by multiplying the complex number $z = x + y\,i$ by the real number s. Thus every conformal transformation of vectors in the plane can be represented by multiplication with a single complex number $w = w_1 + i\,w_2$, where $s = |w|$ is the scale factor and $\phi = arctan\left(w_2 / w_1\right)$ is the angle of rotation.

Quaternions are complex numbers on steroids. Quaternion multiplication is an extension of complex multiplication from the 2-dimensional space of vectors in the plane to the 4-dimensional space of mass-points. The product of two quaternions incorporates dot product, cross product and scalar product and is given by the formula

$$(aO + u)(bO + v) = (ab - u \cdot v)O + (b\,u + a\,v + u \times v)\,,$$

where O represents the origin of the points in 3-dimensional affine space.

Quaternion multiplication can be used to model conformal transformations of vectors in 3-dimensions by sandwiching a vector v between a quaternion q and its conjugate q^*:

$$S_q(v) = q\;v\;q^*\,.$$

In particular, we have the following formulas for rotation, mirror image, and uniform scaling.

Rotation -- around the unit direction vector N by the angle ϕ

$$q(N,\phi/2) = \cos(\phi/2)O + \sin(\phi/2)N$$

$$S_{q(N,\phi)}(v) = q(N,\phi)\;v\;q^*(N,\phi)$$

Mirror Image -- in the plane perpendicular to the unit normal N

$$-S_N(v) = N\,v\,N$$

Uniform Scaling -- by the scale factor c

$$S_{\sqrt{c}\,O}(v) = c\,v$$

Quaternions provide more compact representations for conformal transformations than 3×3 matrices, since quaternions are represented by 4 rectangular coordinates whereas 3×3 matrices contain 9 scalar entries. Composing conformal transformations by multiplying quaternions is faster than composing conformal transformations by matrix multiplication because to compute the product of two quaternions requires only 16 real scalar multiplications whereas to compose two transformations by matrix multiplication requires 27 real scalar multiplications. Nevertheless, quaternions are not a panacea. The total cost for computing one conformal transformation on a vector using quaternions is 28 scalar multiplications, whereas the cost for multiplying a vector by a 3×3 matrix is only 9 scalar multiplications. Thus matrices are more than 3 times faster than quaternions for computing conformal transformations.

Quaternions are used in Computer Graphics to avoid distortions that can arise during conformal transformations due to numerical inaccuracies introduced by floating point computations. Quaternion transformations never distort angles because every quaternion represents a conformal transformation. Moreover, unlike 3×3 matrices, quaternions are easily normalized, so unit quaternions can be employed to avoid distorting distances as well as angles during rotation.

Quaternions are also used in Computer Graphics to perform key frame animation. In key frame animation, an artist draws only a few key frames in the scene, and an animator must then interpolate intermediate frames to make the animation look natural. Linear interpolation of rotation matrices does not generate rotation matrices. Linear interpolation of unit quaternions does not generate unit quaternions and even when normalized the resulting unit quaternions are not uniformly spaced along an arc in 4-dimensions, so the resulting motion appears jerky rather than smooth. To interpolate smoothly between two rotations represented by the unit quaternions q_0, q_1, we apply spherical linear interpolation:

$$slerp(q_0, q_1, t) = \frac{\sin((1-t)\phi)}{\sin(\phi)} q_0 + \frac{\sin(t\phi)}{\sin(\phi)} q_1,$$

where ϕ is the angle between q_0 and q_1. Spherical linear interpolation avoids undesirable artifacts that arise from interpolation methods that either fail to generate rotations or build rotations that do not transition smoothly between the artist's original rotations.

Unit quaternions $q(N,\phi/2) = (q_1, q_2, q_3, q_4)$ and 3×3 matrices $Rot(N,\phi) = (R_{i,j})$ can both be used to represent rotations around the unit axis vector N by the angle ϕ. The following formulas can be applied to convert between these two representations:

$$q(N,\phi) = \frac{\sqrt{R_{1,1} + R_{2,2} + R_{3,3} + 1}}{2} O + \frac{(R_{2,3} - R_{3,2})i + (R_{3,1} - R_{1,3})j + (R_{1,2} - R_{2,1})k}{2\sqrt{R_{1,1} + R_{2,2} + R_{3,3} + 1}}$$

$$Rot(N,\phi) = \begin{pmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_3q_4 & 2q_1q_3 - 2q_2q_4 \\ 2q_1q_2 - 2q_3q_4 & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_1q_4 \\ 2q_1q_3 + 2q_2q_4 & 2q_2q_3 - 2q_1q_4 & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}.$$

Exercises:

1. Using Equation (2.4), verify that complex multiplication is associative, commutative, and distributes through addition.

2. Let (r,θ) denote the polar coordinates of a point in the plane. Show that in polar coordinates complex multiplication is equivalent to

$$(r_1,\theta_1)\cdot(r_2,\theta_2)=(r_1 r_2,\theta_1+\theta_2) .$$

3. Here we shall derive the half angle formulas used in the text.
   a. Using Euler's formula -- Equation (2.9) -- show that
      $\cos(2\phi)+i\sin(2\phi)=\left(\cos(\phi)+i\sin(\phi)\right)^2$.
   b. Conclude from part *a* that:
      i. $\cos(2\phi)=\cos^2(\phi)-\sin^2(\phi)=2\cos^2(\phi)-1$
      ii. $\sin(2\phi)=2\sin(\phi)\cos(\phi)$.
   c. Conclude from part *b* that:
      i. $\cos(\phi/2)=\sqrt{\dfrac{1+\cos(\phi)}{2}}$
      ii. $\sin(\phi/2)=\sqrt{\dfrac{1-\cos(\phi)}{2}}$.

4. Let w_1,w_2,w be complex numbers and let $p(z)=a_n z^n+\cdots+a_0$ be a polynomial in the complex variable z with real coefficients $a_0,\ldots,a_n$.
   a. Show that:
      i. $(w_1+w_2)^*=w_1^*+w_2^*$
      ii. $(w_1 w_2)^*=w_1^* w_2^*$
      iii. $(w^n)^*=(w^*)^n$
   b. Conclude from part *a* that if $p(w)=0$, then $p(w^*)=0$.

5. Using Equation (3.7) and the properties of dot product and cross product, verify that quaternion multiplication is associative and distributes through addition.

6. Let a,b,x,y be real numbers, and let $z_1=a+bi$, $z_2=x+yi$ be complex numbers.
   a. Verify Brahmagupta's identity:

$$(a^2+b^2)(x^2+y^2)=(ax-by)^2+(ay+bx)^2 .$$

b. Using Brahmagupta’s identity, prove that

$$|z_1 z_2|=|z_1||z_2| .$$

7. Let a,b,c,d,w,x,y,z be real numbers, and let $q_1=a+bi+cj+dk$, $q_2=w+xi+yj+zk$ be quaternions.

a. Verify the following generalization of Brahmagupta’s identity:

$$(a^2+b^2+c^2+d^2)(w^2+x^2+y^2+z^2)=(aw-bx-cy-dz)^2+(ax+bw+cz-dy)^2 \\ +(ay-bz+cw+dx)^2+(az+by-cx+dw)^2 .$$

b. Using the result in part *a*, prove that

$$|q_1 q_2|=|q_1||q_2| .$$

8. Consider a quaternion $q=q_0 O+q_1 i+q_2 j+q_3 k$.

a. Show that

$$q=(q_0 O+q_1 i)+(q_2 O+q_3 i)j .$$

b. Conclude from part *a* that every quaternion can be represented by a pair of complex numbers.

c. Show that for any complex number $\alpha=aO+bi$

$$j\alpha=\alpha^* j .$$

d. Let a,b,c,d be complex numbers. Using part *c*, show that

$$(a+bj)(c+dj)=(ac-bd^*)+(ad+bc^*)j .$$

9. Let q be an arbitrary quaternion. Define

$$e^q=O+q+\frac{q^2}{2!}+\frac{q^3}{3!}+\cdots$$

a. Show that if N is a unit vector, then

$$e^{\lambda N}=cos(\lambda)O+sin(\lambda)N .$$

b. Conclude from part *a* and Proposition 3.4 that a vector v can be rotated around a unit vector N through the angle ϕ by sandwiching v between $q=e^{\phi N/2}$ and $q^*=e^{-\phi N/2}$.

10. Consider two unit vectors v_0,v_1. Let ϕ be the angle between v_0 and v_1 and let $w=\dfrac{v_0\times v_1}{|v_0\times v_1|}$.

a. Using the results of Exercise 9, show that:

i. $slerp(v_0,v_1,t)=e^{t\phi w/2}\, v_0\, e^{-t\phi w/2}$

ii. $v_1^* v_0 = e^{\phi w}$

b. Conclude from part *a* that

$$slerp(v_0, v_1, t) = (v_1^* v_0)^{t/2} \; v_0 \left(v_0^* v_1 \right)^{t/2} .$$

11. In this exercise we explore the effect of sandwiching on points in affine space.

a. Show that for any quaternions q, a, b

$$S_q(a+b) = S_q(a) + S_q(b) .$$

b. Show that:

i. $S_{q(N,\theta)}(O) = O$

ii. $-S_N(O) = -O$

iii. $S_{\sqrt{c}\,O}(O) = c\,O$

c. Conclude from parts *a,b* that if *P* is a point in affine space, then

i. $S_{q(N,\phi/2)}(P)$ rotates the point *P* by the angle ϕ around the line *L* through the origin parallel to the vector *N*.

ii. $-S_N(P)$ is not the mirror image of *P* in the plane through the origin perpendicular to the unit normal vector *N*, (For the correct sandwiching formula for mirror image on points, see Exercise 15.). Give a geometric interpretation of $-S_N(P)$.

iii. $S_{\sqrt{c}\,O}(P)$ scales the mass of *P* by the factor *c*, but leaves the location of *P* unchanged.

12. Let $p = p_4 O + p_1 i + p_2 j + p_3 k$ and $q = q_4 O + q_1 i + q_2 j + q_3 k$. Show that:

$$\begin{aligned} pq = (p_4 q_4 - p_1 q_1 - p_2 q_2 - p_3 q_3) O + (p_4 q_1 + p_1 q_4) i + (p_4 q_2 + p_2 q_4) j + (p_4 q_3 + p_3 q_4) k \\ + (p_2 q_3 - p_3 q_2) i + (p_3 q_1 - p_1 q_3) j + (p_1 q_2 - p_2 q_1) k \end{aligned}$$

13. Let $p = (p_1, p_2, p_3, p_4)$ and $q = (q_1, q_2, q_3, q_4)$ be quaternions, and let

$$L(q) = \begin{pmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q4 \end{pmatrix} \quad \text{and} \quad R(q) = \begin{pmatrix} q_4 & -q_3 & q_2 & -q_1 \\ q_3 & q_4 & -q_1 & -q_2 \\ -q_2 & q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{pmatrix} .$$

a. Using Exercise 12, verify that:

i. $q\,p = (p_1, p_2, p_3, p_4) * L(q)$

ii. $p\,q = (p_1, p_2, p_3, p_4) * R(q)$

b. Conclude from part *a* that

$$S_q(p) = (p_1, p_2, p_3, p_4) * \left(L(p) * R(q^*) \right) .$$

c. By direct computation, verify that

$$L(p)*R(q^*)=\begin{pmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2+2q_3q_4 & 2q_1q_3-2q_2q_4 & 0 \\ 2q_1q_2-2q_3q_4 & q_4^2-q_1^2+q_2^2-q_3^2 & 2q_2q_3+2q_1q_4 & 0 \\ 2q_1q_3+2q_2q_4 & 2q_2q_3-2q_1q_4 & q_4^2-q_1^2-q_2^2+q_3^2 & 0 \\ 0 & 0 & 0 & q_1^2+q_2^2+q_3^2+q_4^2 \end{pmatrix}$$

.

d. Conclude from parts *b,c* that if $q=q(N,\phi/2)$, then

$$Rot(N,\phi)=\begin{pmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2+2q_3q_4 & 2q_1q_3-2q_2q_4 \\ 2q_1q_2-2q_3q_4 & q_4^2-q_1^2+q_2^2-q_3^2 & 2q_2q_3+2q_1q_4 \\ 2q_1q_3+2q_2q_4 & 2q_2q_3-2q_1q_4 & q_4^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}.$$

14. Let $q=q_4O+q_1i+q_2j+q_3k$.

a. By direct computation show that

i. $q\,i\,q^*=\left(q_4^2+q_1^2-q_2^2-q_3^2,\,2q_1q_2+2q_3q_4,\,2q_1q_3-2q_2q_4\right)$

ii. $q\,j\,q^*=\left(2q_1q_2-2q_3q_4,\,q_4^2-q_1^2+q_2^2-q_3^2,\,2q_2q_3+2q_1q_4\right)$

iii. $q\,k\,q^*=\left(2q_1q_3+2q_2q_4,\,2q_2q_3-2q_1q_4,\,q_4^2-q_1^2-q_2^2+q_3^2\right)$

b. Conclude from part *a* that if $q=q(N,\phi/2)$, then

$$Rot(N,\phi)=\begin{pmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2+2q_3q_4 & 2q_1q_3-2q_2q_4 \\ 2q_1q_2-2q_3q_4 & q_4^2-q_1^2+q_2^2-q_3^2 & 2q_2q_3+2q_1q_4 \\ 2q_1q_3+2q_2q_4 & 2q_2q_3-2q_1q_4 & q_4^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}.$$

15. Let $L(q)$ and $R(q)$ be as in Exercise 11, and let *N* be a unit vector.

a. By direct computation, verify that

$$L(N)*R(N)=\begin{pmatrix} 1-2N_1^2 & -2N_1N_2 & -2N_1N_3 & 0 \\ -2N_1N_2 & 1-2N_2^2 & -2N_2N_3 & 0 \\ -2N_1N_3 & -2N_2N_3 & 1-2N_3^2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

b. Let $Mir_{3\times 3}(N)$ be the 3×3 matrix that mirrors vectors in the plane perpendicular to *N*. Using part *a*, show that:

$$Mir_{3\times 3}(N)=\begin{pmatrix} 1-N_1^2 & -2N_1N_2 & -2N_1N_3 \\ -2N_1N_2 & 1-N_2^2 & -2N_2N_3 \\ -2N_1N_3 & -2N_2N_3 & 1-N_3^2 \end{pmatrix}$$

c. Let $P=(p_1,p_2,p_3,1)$ be a point in affine space and let

$$Mir_{4\times 4}(N)=\begin{pmatrix} Mir_{3\times 3}(N) & 0 \\ 0 & -1 \end{pmatrix}=\begin{pmatrix} 1-2N_1^2 & -2N_1N_2 & -2N_1N_3 & 0 \\ -2N_1N_2 & 1-2N_2^2 & -2N_2N_3 & 0 \\ -2N_1N_3 & -2N_2N_3 & 1-2N_3^2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}..$$

Show that the mirror image of *P* in the plane through the origin perpendicular to the unit normal *N* is given by

$(p_1,p_2,p_3,-1)*Mir_{4\times 4}(N)$.

d. Conclude from parts *a* and *c* the mirror image of a point *P* in the plane through the origin *O* perpendicular to the unit normal *N* is given by sandwiching $P-2O$ between two copies of *N*.

16. Show that:

a $q(N,\phi)^{-1}=q(N,\phi)^*=q(N,-\phi)=q(-N,\phi)$

b. $rot(N,\phi)^{-1}=rot(N,\phi)^T=rot(N,-\phi)=rot(-N,\phi)$

17. Let *v* be a vector and *q* be a unit quaternion. Show that

a. $S_{-q}(v)=S_q(v)$.

b. If $S_q(v)$ rotates the vector *v* around the unit vector *N* by the angle θ, then $S_{q^*}(v)$ rotates the vector *v* around the unit vector *N* by the angle $-\theta$.

18. Let v,w be unit vectors.

a. Show that:

i. $v\,w$ is a unit quaternion.

ii. $S_{v\,w}(w)$ lies in the plane of the vectors v,w .

iii. *v* bisects the angle between *w* and $S_{v\,w}(w)$.

b. Prove that mirroring a vector *u* in the plane perpendicular to *w* and then mirroring the resulting vector in the plane perpendicular to *v* is equivalent to rotating the vector *u* around the vector $v\times w$ by twice the angle between *w* and *v* (see also Exercise 20).

19. In this exercise we will study the composite of two rotations.

a. Show that if *p* and *q* are unit quaternions, then pq is a unit quaternion.

b. Using part *a*, show that

$q(N_1,\phi_1)\,q(N_2,\phi_2)=q(N,\phi)$,

where

i. $\cos(\phi/2)=\cos(\phi_1/2)\cos(\phi_2/2)-\sin(\phi_1/2)\sin(\phi_2/2)(N_1\cdot N_2)$

ii. $\sin(\phi/2)N=\cos(\phi_2/2)\sin(\phi_1/2)N_1+\cos(\phi_1/2)\sin(\phi_2/2)N_2$
$\qquad\qquad+\sin(\phi_1/2)\sin(\phi_2/2)N_1\times N_2$

c. Conclude from part *b* that

$q(N,\phi_1)\,q(N,\phi_2)=q(N,\phi_1+\phi_2)$.

d. Using parts *b,c* show that

i. $rot(N_2,\phi_1) * rot(N_1,\phi_2)=rot(N,\phi)$

ii. $rot(N,\phi_1) * rot(N,\phi_2)=rot(N,\phi_1+\phi_2)$.

20. In this exercise we will show that every rotation is the composite of two reflections. Let N be a unit vector and let u,v be two unit vectors in the plane perpendicular to N with $\mathrm{sgn}(u,v,N)>0$ and $\angle(u,v)=\phi/2$. Show that

a. $q(N,\phi/2)=-v\,u$

b. $S_{q(N,\phi/2)}(w)=(v\,u)\,w\,(uv)$

c. $S_{q(N,\phi/2)}=S_v\circ S_u$

d. Conclude from part *c* that every rotation is the composite of two reflections.

21. In this exercise we will develop formulas for some non-conformal affine transformations (see Chapter 12) using quaternions.

a. Show that:

i. $(N\cdot v)O=\dfrac{-(vN+Nv)}{2}$

ii. $N\times v=\dfrac{Nv-vN}{2}$

b. Using the results of part *a*, verify that

. i. *Orthogonal Projection -- onto the plane perpendicular to the unit normal N*

$$v_{new}=\frac{v-S_N(v)}{2}$$

ii. Parallel *Projection -- onto the plane perpendicular to the unit normal N in the direction parallel to w*

$$v_{new}=v-\frac{v\,N\,w+N\,v\,w}{w\,N+N\,w}$$

iii. *Non-Uniform Scaling -- along the direction w by the scale factor c*

$$v_{new} = \frac{v - S_w(v)}{2} + c\,\frac{v + S_w(v)}{2}$$

22. In this exercise we will show how to use sandwiching to compute perspective projection. Let $0 < \theta < \pi$, and let

- S = the plane through the point $O + \cot(\theta)N \equiv T_{q(N,-\theta)}(N)$ perpendicular to the unit normal N
- $E = O + (\cot(\theta) - \csc(\theta))N =$ eye point (at a distance $\csc(\theta)$ from the plane S)
- P = a point in 3-dimensions

a. Let

-- $v_\perp$ = a vector perpendicular to N

-- $v_{||}$ = a vector parallel to N

Using Propositions 3.2 and 3.3, show that

-- $q(N,-\theta/2)\ v_\perp\ q(N,-\theta/2)$ is the identity on $v_\perp$

-- $q(N,-\theta/2)\ v_{||}\ q(N,-\theta/2)$ rotates $v_{||}$ by the angle $-\theta$ in the plane of O,N

b. Using part *a*, show that

- $q(N,-\theta/2)\ (P-E)\ q(N,-\theta/2)$ is a mass-point, where:

-- the point is located at the perspective projection of the point *P* from the eye point *E* onto the plane *S*;

-- the mass is equal to $d\,sin(\theta)$, where *d* is the distance of the point *P* from the plane through the eye point *E* perpendicular to the unit normal *N*.

(Hint: See Figure 10)

c. Conclude from part *b* that a perspective projection in 3-dimensions corresponds to a simple rotation in 4-dimensions.

d. Let $q = q(N,-\theta/2) = (q_1, q_2, q_3, q_4)$. Using part *b* and the definition of the matrices $L(q)$ and $R(q)$ in Exercise 13, show that the 4×4 rotation matrix corresponding to perspective projection is given by

$$Persp(N,\theta) = L(q) * R(q) = \begin{pmatrix} 1-2q_1^2 & -2q_1q_2 & -2q_1q_3 & -2q_1q_4 \\ -2q_1q_2 & 1-2q_2^2 & -2q_2q_3 & -2q_2q_4 \\ -2q_1q_3 & -2q_2q_3 & 1-2q_3^2 & -2q_3q_4 \\ 2q_1q_4 & 2q_2q_4 & 2q_3q_4 & 2q_4^2-1 \end{pmatrix}$$

e. Generalize the result in part *b* to arbitrary position of the eye point *E* and the plane of projection *S*. (Hint: Perspective projection and translation commute.)

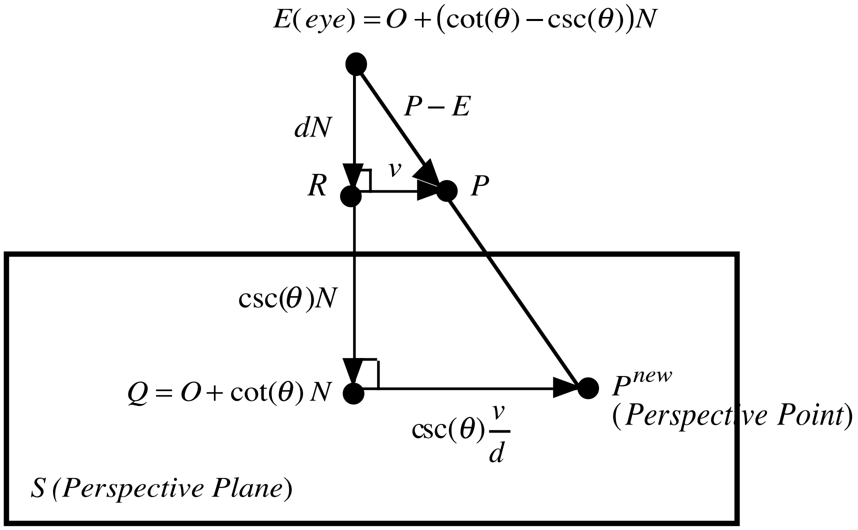

$E(eye) = O + (\cot(\theta) - \csc(\theta))N$
dN
$P - E$
v
R
P
$\csc(\theta)N$
$Q = O + \cot(\theta)\, N$
P^{new}
($Perspective\ Point$)
$\csc(\theta)\frac{v}{d}$
$S\ (Perspective\ Plane)$

Figure 10: By similar triangles, the point

$$P^{new} = O + \cot(\theta)N + \csc(\theta)\frac{v}{d} = E + \csc(\theta)N + \csc(\theta)\frac{v}{d}$$

is the perspective projection of the point P from the eye point E onto the plane S through the point $Q = O + \cot(\theta)\, N$ perpendicular to the unit normal N.

Programming Projects

15.1 *Quaternion Algorithms*

Implement quaternion multiplication.

a. Use quaternion multiplication to:

   i. rotate vectors in 3-dimensions.

   ii. mirror vectors in 3-dimensions

   iii. project points from an eye point E into a perspective plane S (see Exercise 22)

b. Compare these quaternion techniques to the corresponding matrix methods for speed and accuracy.

15.2, *Key Frame Animation*

Give some examples of key frame animation, using quaternions together with spherical linear interpolation (SLERP).