Teaching Quaternions is not Complex

J. McDonald

DePaul University, Chicago, IL, United States

Abstract

Quaternions are used in many fields of science and computing, but teaching them remains challenging. Students can have a great deal of trouble understanding essentially what quaternions are and how they can represent rotation matrices. In particular, the similarity transform $\mathrm{q}\boldsymbol{v}\mathrm{q}^{-1}$ *which actually achieves rotation, can often be baffling even after they've seen a full derivation. This paper outlines a constructive method for teaching quaternions, which allows students to build intuition about what quaternions are, and why simple multiplication is not adequate to represent a rotation. Through a set of examples, it demonstrates exactly how quaternions relate to rotation matrices, what goes wrong when* $\mathrm{q}\boldsymbol{v}$ *is naively used to rotate vectors, and how the similarity transform fixes the problem.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: General, K.3.2 [Computers and Education]: Computer Science Education

1. Introduction

Since their initial introduction in 1866 by Hamilton [Ham86], quaternions have found applications in many scientific fields including quantum mechanics, electrodynamics, special relativity and computer graphics [dLe96, Hor97, Ima76, Sho85]. Their compact representation of concepts like rotation in three dimensions, and the fact that we can conveniently build smooth interpolation curves between quaternions, make them an important technique for students in many areas of science and computation.

In computer graphics they give us a powerful tool for representing the orientation of an object. Many graphics packages use quaternions for interpolating orientations smoothly to create animations of rotating objects. In addition, such packages often provide extensive scripting tools to allow users to manipulate them programmatically[Sof09, Max09, May09]. Quaternions are also a key tool used in many types of interactive software to control camera movement [Bob03].

Quaternions, therefore, figure prominently in quite a few courses, and teaching them is a challenge that every instructor for such a course has struggled with. Though their arithmetic rules are quite simple and some tools exist to help students visualize quaternions [HFK94], quaternions themselves are often not truly understood until students have worked with them for quite a while. There are several reasons for this:

1. The multiplication rules, while simple, can seem quite abstract and arbitrary to first time students.
2. The method for using quaternions to rotate vectors in 3D is very different from matrix multiplication.
3. Exponential notation, while powerful, can be difficult for graphics students to grasp even if they have had a course in Taylor series.

Part of the problem is that quaternions live in four dimensions. This is just one dimension more than students are used to dealing with, and it is compelling, though difficult, to try to visualize them. Compare this to the fact that students rarely attempt to visualize 4x4 matrices as they live in a 16 dimensional space!

This paper presents a method for introducing students to quaternions that maintains a firm connection to prior knowledge. It assumes that the students are adept in working with rotation matrices, and uses them to motivate the discussion. Finally, the method lays out a series of geometric examples that bring students to discover why vector rotation requires a similarity transform rather than a simple multiplication.

2. Teaching quaternions

Though much work has gone into helping students visualize quaternions and quaternion curves in four dimensions [HFK94], their initial presentation in most texts generally follows the following pattern:

1. Review the basics of complex numbers
2. Define three different roots of -1, called i, j, and k, and define quaternions as a sum $q = w+xi+yj+zk$
3. Present the rules for multiplication, conjugation, etc. Many texts also point out the relationship between the multiplication of i, j, and k, and cross products.
4. Demonstrate that if $q = \cos(\theta/2) + \sin(\theta/2)\,\mathbf{a}$ then $q\mathbf{v}q^{-1}$ gives rotation of $\mathbf{v}$ by θ about the axis $\mathbf{a}$.

This last step is accomplished by calculating the expression for $q\mathbf{v}q^{-1}$ through and equating the final result to the arbitrary rotation matrix, for example, see [Ang03, Par07, vVe04, War97, Wat92].

This is fine for the most adept students, especially if they have had a course in abstract algebra, but many students in graphics have a rough time with the fact that the only time this is tied back to something they know is right at the end. Further, the matrix that it is tied to is the most complicated of rotation matrices. Many students are left wondering why rotation requires such a complicated operation when with matrices it is simply $M\mathbf{v}$.

Some texts have tried to tie quaternions to matrix multiplication earlier in the development. For example, several texts note that a quaternion $w+xi+yj+zk$ can be represented by a matrix like

$$\begin{bmatrix} w & -z & y & x \\ z & w & -x & y \\ -y & x & w & z \\ -x & -y & -z & w \end{bmatrix}$$

The main problem with jumping immediately to this representation is that, though helpful in allowing students to tie a quaternion to a specific matrix, this matrix is not one of the fundamental transformations that students are used to working with. In addition, there are several such matrices, which all produce valid matrix representations of quaternions, and there is no consensus for a canonical form; each text presents a different matrix representation [Mar05, vVe04]. These correspond to different embeddings of the space of quaternions in the space of 4x4 matrices. For more discussion on this, see Appendix A.

Others have suggested that replacing quaternions with rotors from the Geometric Algebra, can be a good approach [DFM08], and it is true that rotors do solve some of the problems that arise in the development of quaternions and in how it is used to achieve rotation. However, to use this method effectively, one would need to couch the entire graphics curriculum in terms of this algebra, something that many programs are not willing to do, partially because of the fact that it is not yet widely used in industry. One of the major complaints, though, from the Geometric Algebra camp is that quaternions are not "geometric" [DFM08]. This paper addresses this by demonstrating, from the beginning of the construction, the geometric meanings and effects of these objects.

The technique presented in this paper makes the connection between matrices and quaternions earlier by defining the primitive elements, i, j and k in terms of matrices, and allows students to see clearly how rotations about coordinate axes are built. These matrices are far simpler than their more general cousin above, and students have a better opportunity to build intuition. Further, we will see that it is possible to visualize geometrically how quaternions work in the formula $q\mathbf{v}q^{-1}$. To help build student intuition from the ground up, however, we begin the story with 2D rotations and complex numbers. Other texts have done this as well, though they didn't carry the connection to rotation matrices nearly as far as we will here [War97].

3. i is not imaginary

By the time graphics students get to courses that cover quaternions, they will have encountered complex numbers, if only in an algebra course covering the quadratic formula. The problem is that very often the seed for confusion about quaternions is already laid in this foundation with the use of the other name for complex numbers: "imaginary". This name implies that this object i doesn't really exist, but that we'll go ahead and define it to be the square-root of -1 anyway.

Many authors have written about the inappropriateness of this name, and I won't go through a complete rundown of why this is unfortunate. In laying the groundwork for computer graphics, however, we have a wonderful opportunity to clear up this confusion since our students have a lot of experience with matrices. Finding a meaning for i that students can relate to is often a matter of embedding the field of complex numbers into some other set that has meaning for students. In our case we are relating 2D rotations to the unit complex numbers, and so, the logical choice is to use matrices to build this connection.

3.1. Giving meaning to i for graphics students

In the set of 2x2 matrices, we have a subset that act just like the real numbers. These matrices are of the form

$$\begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$$

Students will quickly recognize these as scale matrices, but it is worth pointing out that both addition and subtraction work *exactly* the same for these matrices as for real numbers. Thus, these matrices are a nice copy of the real numbers sitting inside this larger set.

If the students then square the following matrix, they will make a nice discovery:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

The quickest students may recognize this matrix as a rotation by 90°, but in any case, they will discover that its square is the scale matrix -1. Therefore, this matrix, in this new number system is $\sqrt{-1}$. For them, this matrix is the i that was so mysteriously called *imaginary.* One can quickly work out the usual properties for complex numbers from here, but now they have a physical meaning to attach to i: Rotation by 90°.

It is worth pointing out here that this version of i assumes that we are dealing with matrices that act on vectors on the left. I.e.

$$M\mathbf{v}$$

and thus the above matrix represents a right-handed rotation. If the students are used to working with vM, then one can create the same development by taking the transposes of our i matrix.

Of course, the most immediate fact we get is that a unit complex number, $\cos\theta + i\sin\theta$, is then equal to

$$\begin{bmatrix} \cos\theta & 0 \\ 0 & \cos\theta \end{bmatrix} + \begin{bmatrix} 0 & -\sin\theta \\ \sin\theta & 0 \end{bmatrix}$$

which when summed equals the standard matrix for rotation that they know. This makes the connection between rotations and unit complex numbers quite evident, and lays the foundation for quaternions. It is also important to point out at this point the fact that the inverse of this matrix is its complex conjugate $\cos\theta - i\sin\theta$, since negating the coefficient on i simply flips the off-diagonal elements giving us the transpose of the matrix.

3.2. Applying unit complex numbers to vectors

It is at this point that we run into a subtle point that will carry over into quaternions. How are we going to apply these complex numbers to vectors. If we are only thinking about complex numbers, we interpret a vector (or a point) in 2D space as a complex number and then we multiply using complex arithmetic. This will be a little unusual for the students because they are thinking of matrices at this point, and thus about

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix} \tag{1}$$

but if we think about the vector as a complex number the product becomes

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x & -y \\ y & x \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta & -y\cos\theta + x\sin\theta \\ y\cos\theta + x\sin\theta & x\cos\theta - y\sin\theta \end{bmatrix}$$

which is the same thing as $(x\cos\theta - y\sin\theta) + i\,(y\cos\theta + x\sin\theta)$. And so the product works out the same either way.

The best students may not get confused by this, but usually this discussion is best left to a later time, and simply appeal to the fact that complex multiplication gives the same answer as the rotation matrix. Students will generally accept this here. It is important for instructors to be aware of this issue, however, because one of the sticking points of the quaternion construction is interpreting vectors as a quaternion. This is analogous to what we have done in this section.

4. Quaternions

When we pass to the discussion of quaternions, the question becomes, can we build an analog of the complex numbers that will allow us to represent rotations in 3D? The development of quaternions can proceed in the same manner as for complex numbers, by building roots of a diagonal matrix of -1's, but with several important differences that need to be treated delicately. The most important thing is to keep connecting new ideas to ones students already know. In the case of graphics students students: matrices.

4.1. 4x4 matrices and roots of -1

When building roots of -1 for rotations in 3D there are two possibilities, 3x3 matrices or 4x4 homogeneous matrices. Since the students have usually been working with homogeneous matrices for a while, they may accept without question using them at this point. However, the brighter students may question whether 3x3 matrices wouldn't be easier and more compact.

One can eliminate this possibility quite easily by appealing to the determinant. The determinant of a 3x3 diagonal matrix with -1's on the diagonal is -1, and so a root for this matrix would have determinant $\sqrt{-1}$. Since our matrices have real entries, their determinants are real, and of course, there is no *real* number whose square is -1. If we use 4x4 matrices, we don't have this problem since the determinant would be $+1$.

In any case, we know that homogeneous matrices are needed in general for transformations in 3D, and so it is not a big stretch for students to accept them here.

We begin the story of quaternions in the same way as complex numbers, by considering a copy of the real numbers in the collection of 4x4 matrices:

$$[a] = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix}$$

Students should note that these are not scale matrices for 3D, though they are for 4D space. This is one place in which the development differs from the 2D case. In 3D, they will scale vectors, whose homogeneous coordinate w is 0, but with points, they will multiply all four coordinates by a, including w. When we return the point to its canonical form, $w = 1$, the a divides out and we get exactly the same point we started out with. This observation contains the beginning of why $q\mathbf{v}q^{-1}$is necessary for rotation.

Such homogeneous matrices are a bit unusual, but the students can quickly grasp what they do, and can easily discover that they obey the same rules as the real numbers under matrix arithmetic. The product and sums of two diagonal matrices $[a]$ and $[b]$ are $[a + b]$ and $[ab]$ respectively.

When trying to find a root of the diagonal matrix $[-1]$, we find that there is more than one possibility, unlike in 2D. In fact the following three matrices are all roots of $[-1]$:

$$i = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

$$j = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

$$k = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2)$$

Each of these matrices is analogous to the matrix i from complex numbers, and can be built by choosing pairs of columns (coordinates) to reverse, and one of each pair to negate. These are not all of the different roots of $[-1]$, however, and in fact the others correspond to negating one of the columns in each pair in different ways. The other roots correspond to different ways to embed quaternions in the space of matrices. The three in equation 2 have been chosen so that they match the corresponding rotation matrices most closely. Appendix A explores this more fully.

In any case, it is easy for the students to check that a simple set of rules governs multiplication of these special matrices, namely the usual quaternion product rules:

$$ij = k,\ jk = i,\ ki = j$$
$$i^2 = j^2 = k^2 = -1 \quad (3)$$

and the fact that the product rules for i, j, and k follow the same rules as the cross-product. So, we define a quaternion as the sum $q = w + xi + yj + zk$, where w, x, y and z are real numbers. At this point we can note what some other texts do, that this quaternion corresponds to the matrix

$$\begin{bmatrix} w & -z & y & x \\ z & w & -x & y \\ -y & x & w & z \\ -x & -y & -z & w \end{bmatrix} \quad (4)$$

Though in this case, students get a clear view of exactly where this matrix comes from.

Amongst the various properties of quaternions that are usually discussed at this point, it is worth tying conjugation back to this matrix. Since $\overline{q} = w - xi - yj - zk$, a modicum of inspection will show the students that the corresponding matrix is the transpose of q's matrix. Since the inverse of a rotation matrix is its transpose, this will help reinforce the idea that for unit quaternions, which will represent rotations, $q^{-1} = \overline{q}$.

Note, that as described in Appendix A, the specific form of matrix 4 will depend on our choices of i, j and k. The specific choices of these matrices here were made to strengthen the analogy between i, j and k and rotation matrices. It is at this point, though, that we begin to leave matrix notation and work exclusively with quaternion notation. We will however tie it back to matrices from time to time to reinforce intuition when building rotations.

4.2. Quaternions and rotations

Before we get to general quaternion multiplication, it is useful to investigate rotations with some simple examples. This not only gets the students used to the rules for quaternion multiplication, but it also allows the students to discover, on their own, the rules that will govern quaternion rotations.

The first two examples investigate what happens with the naive attempt to use the direct analog of complex rotation in section 3.2 only this time in the context of quaternions. Of course, the effort will fail, but it is in *how* it fails that the students will build

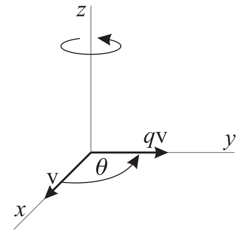

Figure 1: *The first rotation*

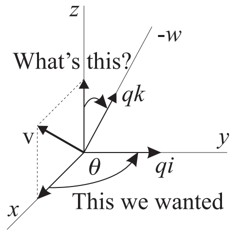

Figure 2: *The second rotation*

their intuition for the real formula: $q = \cos(\theta/2) + \sin(\theta/2)\,\mathbf{a}$ and its application to a vector $q\mathbf{v}q^{-1}$.

The first thing to note, is that just like we converted vectors in the 2D case to complex numbers, we will do the same here. The only difference is that a quaternion has four dimensions and a vector only 3. But appealing to homogeneous coordinates, and looking at the form of a quaternion $q = w + xi + yj + zk = w + \mathbf{v}$ gives us the answer. We will let a vector be represented by a quaternion with $w = 0$. So, $\langle 2, 3, 1\rangle = 0 + 2i + 3j + k$. With this method of representing vectors as quaternions we are ready to give rotation a try.

4.2.1. Example 1, first attempt

For the first example, we begin with the simplest of quaternion rotations, and we apply it to a very simple vector. Consider a rotation of by an angle θ about the z-axis. If we follow the pattern suggested to us by complex numbers, we might take a guess at trying the quaternion

$$q = \cos\theta + \sin\theta\,\langle 0, 0, 1\rangle = \cos\theta + k\sin\theta$$

The first vector we will attempt to rotate is a simple one as well $\mathbf{v} = \langle 1, 0, 0\rangle$, which when converted to a quaternion by attaching $w = 0$, simply becomes i. Clearly, when rotated by an angle of θ about the z-axis, this vector should be transformed to $\langle \cos\theta, \sin\theta, 0\rangle$.

In the complex case, we could rotate a vector by simply multiplying, so we try it here and see what we get.

$$\begin{aligned} q * \mathbf{v} &= (\cos\theta + k\sin\theta) * i \\ &= i\cos\theta + ki\sin\theta \\ &= i\cos\theta + j\sin\theta = \langle \cos\theta, \sin\theta, 0\rangle \end{aligned}$$

Just as expected. So for the moment quaternions are following exactly in the footsteps of the complex case, see Figure 1 where the case of $\theta = \pi/2$ is shown.

We continue exploring with this example, to see if this naive approach continues to work, by rotating a more general vector, such as $\mathbf{v} = \langle 1, 0, 1\rangle$, if the rotation is working properly, we should get $\langle \cos\theta, \sin\theta, 1\rangle$. In this case, when we convert $\mathbf{v}$ to a quaternion we get $i + j$, and so, the product for rotation gives us

$$\begin{aligned} q * \mathbf{v} &= (\cos\theta + k\sin\theta) * (i + k) \\ &= i\cos\theta + ki\sin\theta + k\cos\theta + k^2\sin\theta \\ &= i\cos\theta + j\sin\theta + k\cos\theta - \sin\theta \end{aligned}$$

Here we've run into a problem. First, the result doesn't correspond to a vector because the real part w is not zero, it is $-\sin\theta$. Second, when we analyze the terms in the result, we see that we did get part of the rotation that we wanted, namely $i\cos\theta + j\sin\theta$. The problem is that we got more than we wanted. The k component of $\mathbf{v}$ was transformed into $k\cos\theta - \sin\theta$, which is itself a rotation by $-\theta$ towards the $-w$-axis! The effect of this in 3D is to shorten the z-coordinate.

The case of $\theta = \pi/2$ is shown in Figure 2, where we have broken out the two components of the vector $\mathbf{v}$ and have shown the effect of the rotation on each. The diagram projects 4D space into 3D by choosing a direction in 3D for the w-axis. This doesn't cause confusion in this case, since $\mathbf{v}$, its projections to the coordinate axes, and the rotation qi all have $w = 0$. The only vector that has a non-zero w-coordinate is $qk = -w$.

As the figure clearly shows, the quaternion multiplication gave us an unexpected "extra" rotation that we don't want. We can even get a hint as to where this comes from if we look back at the matrix for i, and build the matrix that corresponds to $\cos\theta + i\sin\theta$ as a quaternion

$$\cos\theta + k\sin\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

This matrix has two 2x2 sub-matrices. In the upper left, we have precisely the terms for the rotation by θ about z. In the lower right hand corner, we have a rotation by $-\theta$ in the z,w-plane, exactly as we are getting in the quaternion multiplication, which is caused directly by the form of k as a 4x4 root of -1. The next example will explore how to fix this.

4.2.2. Example 2, fixing the extra rotation

The example in the last section shows students that simple quaternion multiplication with a unit quaternion came close to giving us the desired rotation, but also gave us an extra unwanted factor. The examples in this section will allow them to explore and find a way to fix this extra factor. The key is in discovering what the order of the multiplication does to the result, and also why the inverse is necessary. Of course we are leading them to the true method of $q\mathbf{v}q^{-1}$, but these examples will show what the q^{-1} on the right is doing geometrically.

We begin by looking at what happens when we reverse the order of the multiplication. After all, quaternion multiplication is not commutative, just like matrix multiplication. As it turns out, the result is quite instructive. We will use the same example before that was causing problems

$$\begin{aligned}\mathbf{v}q &= (i+k)*(\cos\theta + k\sin\theta) \\ &= i\cos\theta + ik\sin\theta + k\cos\theta + k^2\sin\theta \\ &= i\cos\theta - j\sin\theta + k\cos\theta - \sin\theta\end{aligned}$$

Taking a close look at what this did, we see that it reversed the desired rotation: $i \to i\cos\theta - j\sin\theta$, but the extra rotation stayed the same!

With this, many students can probably guess what we have to do. Inverting q will now reverse both of the rotations.

$$\begin{aligned}\mathbf{v}q^{-1} &= (i+k)*(\cos\theta - k\sin\theta) \\ &= i\cos\theta - ik\sin\theta + k\cos\theta - k^2\sin\theta \\ &= i\cos\theta + j\sin\theta + k\cos\theta + \sin\theta\end{aligned}$$

Now, the extra rotation is by positive θ, and the desired rotation is in the same direction as before.

So, we can fix the unwanted rotation that occurs in $q\mathbf{v}$ by multiplying on the right by q^{-1}. The first multiplication gives us the desired rotation plus an unwanted rotation, the second rotates again by the desired rotation and then undoes the unwanted one in the z,w-plane. The only problem is that now we've rotated by 2θ, but this is easy to correct if we change our quaternion to

$$q = \cos\left(\frac{\theta}{2}\right) + k\sin\left(\frac{\theta}{2}\right)$$

4.2.3. An interactive demonstration

It is far easier for students to grasp the geometric meaning of the last example if they are provided with an interactive tool that allows them to experiment with the different possibilities. Figure 3 shows just such a tool, that allows students to vary the angle of rotation for the quaternion in the above example.

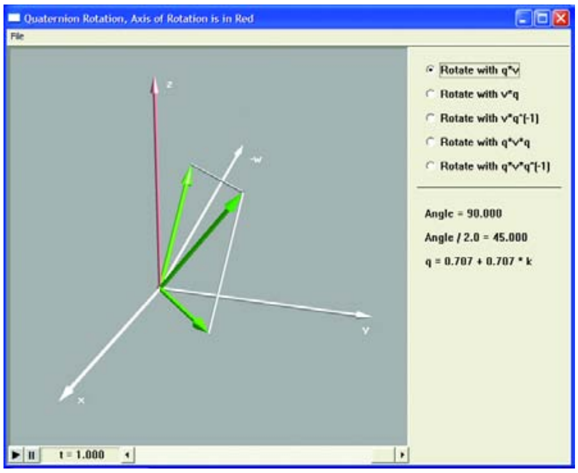

Figure 3: *Tool for exploring quaternion rotation*

This interactive demo uses a projection from 4D to 3D much like the one in Figure 2. It provides an interactive means of exploring the spherical linear interpolation from $q_0 = 1$ to $q_1 = \sqrt{2}/2 + \sqrt{2}/2k$, and allows students to choose some of the different methods of multiplying the vector $\langle 1, 0, 0\rangle$ by q. The last one in the list is, of course, the full rotation, which is the only one in which the component of the vector along the rotation axis doesn't move.

By trying out the different possibilities, and using the animation controls provided, the user can explore the effect of both left and right multiplication by both q and its inverse. This tool helps reinforce the geometric meaning of the rotation. The Windows executable for this tool will be provided for use at the following address: `http://facweb.cdm.depaul.edu/quat`.

4.3. Quaternion multiplication

Now that the students have built up some solid intuition about quaternion multiplication and rotation, we can proceed in the same manner as most texts in presenting some of the other operations of quaternions. Of primary interest is multiplication. The rules in equation 3 can be used to derive the general rule for the quaternion product. It is most elegant in vector notation: by writing $q_n = w_n + \mathbf{v}_n$ where $\mathbf{v}_n = x_n i + y_n j + z_n k$ for $n = 0, 1$, we can derive as usual,

$$q_0 q_1 = (w_0 w_1 - \mathbf{v_0} \bullet \mathbf{v_1}) + (w_0 \mathbf{v}_1 + w_1 \mathbf{v_0} + \mathbf{v}_0 \times \mathbf{v_1}) \quad (5)$$

It can be interesting for students to discuss the fact that the product of two matrices like equation 4 can be decomposed into familiar operations with a highly geometric interpretation.

More importantly, this expression for multiplication allows us to derive the final form of quaternion rotation. To rotate a vector by an angle θ about an axis $\mathbf{a}$, we use the quaternion $q = \cos(\theta/2) + \sin(\theta/2)\,\mathbf{a}$, just like in the simpler case above. We then apply it to the vector by the similarity transform $q\mathbf{v}q^{-1}$. The first thing to note here is that q has the same effect as we saw earlier with our simple example. It rotates the part of $\mathbf{v}$ that is perpendicular to $\mathbf{a}$ correctly by $\theta/2$, but at the same time it rotates the component of $\mathbf{v}$ that is parallel to $\mathbf{a}$ towards the $-w$ axis. Therefore, multiplying on the other side by q^{-1} finishes the desired rotation and undoes the rotation of $\mathbf{a}$.

Thus we can derive the equality

$$q\mathbf{v}q^{-1} = \mathbf{v}_\perp \cos\theta + (\mathbf{v}_\perp \times \mathbf{a}) \sin\theta + \mathbf{v}_\parallel$$

where $\mathbf{v}_\parallel = (\mathbf{v} \bullet \mathbf{a})\,\mathbf{a}$, is the component of $\mathbf{v}$ parallel to $\mathbf{a}$, and $v_\perp = \mathbf{v} - (\mathbf{v} \bullet \mathbf{a})\,\mathbf{a}$, is the component perpendicular to $\mathbf{a}$. This derivation can proceed in the usual manner, using equation 5 with a good dose of trigonometry and linear algebra, see [vVe04]. With the developments above, the students will have far better intuition to help them understand this derivation.

5. Conclusion

The technique presented here provides an introductory path to quaternions that reinforces their geometric nature, and allows students to visualize their functionality geometrically at each step. By connecting quaternions to matrices early in the development, it is easier for graphics students to grasp the geometric meaning of the operations. Also, breaking the derivation of the quaternion rotation formula into a series of elementary examples, students can gain a better understanding of how this formula actually accomplishes its job.

As to next steps, there are several possibilities. Certainly, there are opportunities for building better visualizations of the operations outlined here. The interactive demo discussed in 4.2.3 is a first step, but extending it to be a visualization of arbitrary quaternion rotation would be helpful. Quaternions remain an important part of many fields, and the techniques that use them are varied and often quite subtle. It may be possible to find simple geometric insights into more subtle quaternion processes that will help students more readily accept them and use them effectively.

Appendix A

One of the confusing points of quaternions is that they have several different possible representations in the matrix ring. This is not something that needs to be discussed in an initial introduction to quaternions, but it is something that may arise, especially when inquisitive students begin exploring multiple texts on the subject. This appendix gives a brief outline of the different representations, and why we've chosen the one in equation 2.

For example, in [vVe04] and [Arv91] they use, as we have done here,

$$q = \begin{bmatrix} w & -z & y & x \\ z & w & -x & y \\ -y & x & w & z \\ -x & -y & -z & w \end{bmatrix} \tag{6}$$

However, in [Mar05] and [Wei08] they use what seems to be a bit more popular representation

$$q = \begin{bmatrix} w & -x & -y & -z \\ x & w & -z & y \\ y & z & w & -x \\ z & -y & x & w \end{bmatrix}^T$$

The transpose is because they are using post-matrix multiplication for vectors and points, i.e. $\mathbf{v}M$. Notice that one of the main differences here is the swapping of the x's and z's. It is important to note however, that even with the coordinates in the same place some representations will use a different pattern of negative signs.

The specific form of the quaternion matrix is determined by the choice of matrices for equation 2, and as we mentioned in section 4.1 a matrix representation begins with a choice of three compatible roots of the 4x4 matrix [-1]. There are three factors that determine the form of the matrices for i, j and k. We build the root by starting with the identity matrix:

1. In the identity matrix, we need to choose two of the three coordinates x, y and z to swap, the remaining coordinate will be swapped with w. This will suffice, since to be a root of [-1], $Row_m \bullet Col_n$ must equal -1 if $n = m$ or 0 otherwise. To keep these roots simple, we do this by letting (m, i) and (i, m) be ± 1 or vice versa. We eliminate w from our consideration since it doesn't correspond to a coordinate in the 3D space of vectors. This choice determines an axis for a 3D rotation and an "axis" for the "extraneous" rotation discussed above. There are three ways to choose this pair, these correspond to the three elements i, j and k.
2. In each of the two pairs of coordinates that were swapped in 1) we choose one to negate. This chooses a direction for each rotation. There are four ways to choose these two negations.
3. Once we do this, we need to identify each of the three roots with one of the primitive quaternion elements i, j and k.

It is the third of these that causes the swap of x and z above, and which also causes the most confusion when looking at different embeddings. Because in these two matrices we have two very different choices for i:

$$i_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \; i_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

These correspond to different choices in both steps 2 and 3.

The curious thing about this is that the first one corresponds to rotation about the x-axis, while the second one corresponds to rotation around the z-axis. Then how could both of them serve as i in a representation of quaternions? The answer lies in looking more closely at how quaternions are multiplied with vectors. We all know that we interpret the vector as a quaternion with $w = 0$, but when talking about multiplication and the matrix form of the quaternion, many texts, for example [Arv91], multiply the quaternion matrix and the vector

$$\begin{bmatrix} q_w & -q_z & q_y & q_x \\ q_z & q_w & -q_x & q_y \\ -q_y & q_x & q_w & q_z \\ -q_x & -q_y & -q_z & q_w \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

This, however, is incorrect, because the vector has been converted into a *quaternion*. So, it should be

$$\begin{bmatrix} q_w & -q_z & q_y & q_x \\ q_z & q_w & -q_x & q_y \\ -q_y & q_x & q_w & q_z \\ -q_x & -q_y & -q_z & q_w \end{bmatrix} \begin{bmatrix} w & -z & y & x \\ z & w & -x & y \\ -y & x & w & z \\ -x & -y & -z & w \end{bmatrix}$$

And so, when we swap coordinates in the representation for the quaternion, they also get swapped in the quaternion form of the vector! The first of these will only produce the desired result if the representation in equation 6 is used, where they match the corresponding transform matrices.

Note also that this is also the only way that $q\mathbf{v}q^{-1}$ makes sense in matrix form. For if $\mathbf{v}$ is a vector, then it is a 4x1 matrix, and we cannot multiply a 4x1 matrix on the right by a 4x4 matrix q^{-1}. This is not something we necessarily want to point out to students at the beginning as it is a subtle point, and the intuitive idea in most texts has value for building intuition, but only if the matrices for i, j and k match the desired rotations. This is precisely why we chose the matrices we did in equation 2.

References

[Ang03] Angel E.: *Interactive Computer Graphics, A Top-Down Approach with OpenGL, Third Edition.* Addison-Wesley, 2003.

[Arv91] Arvo J.: *Graphics Gems II.* Academic Press, 1991.

[Bob03] Bobick N., Rotating Objects Using Quaternions. Gamasutra, 2003. Accessed 12/9/2008. <http://www.gamasutra.com/features/19980703/quaternions_01.htm>

[dLe96] De Leo S: Quaternions and Special Relativity. *Journal of Mathematical Physics*, vol. 37, 1996, pp. 2955-2968.

[DFM08] Dorst L., Fontijne D., Mann S.: *Geometric Algebra for Computer Science, An Object Oriented Approach to Geometry.* Morgan Kaufmann, 2008.

[Ham86] Hamilton W. R.: *Elements of Quaternions.* Longmans, Green & Co., 1886.

[HFK94] Hart J. C., Francis G. K., Kauffman L. H.: Visualizing Quaternion Rotation. ACM Transactions on Graphics, vol. 13, no. 3, 1994, pp. 256-276.

[Hor97] Horn B. K. P.: Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America*, vol. 4, April 1997, pp. 629-642.

[Ima76] Imaeda K.: A New Formulation of Classical Electrodynamics. Il Nuovo Cimento, vol. 32, no. 1, 1976.

[Mar05] Marsh D.: *Applied Geometry for Computer Graphics and CAD, Second Edition.* Springer, 2005.

[Max09] 3DS Max 2009. Software by Autodesk corp., 2009.

[May09] Autodesk Maya 2009. Software by Autodesk corp., 2009.

[Par07] Parent R.: *Computer Animation, Algorithms and Techniques, Second Edition.* Morgan Kaufmann, 2007.

[Sho85] Shoemake K.: Animating Rotation with Quaternion Curves. *Computer Graphics,* ACM SIGGRAPH, vol. 19, no. 3, 1985, pp. 245-254.

[Sof09] Softimage XSI. Software by Autodesk corp., 2009.

[vVe04] Van Verth J. M.: *Essential Mathematics for Games and Interactive Applications.* Morgan Kaufmann, 2004.

[War97] Ward J. P.: From Complex Numbers to Quaternions. *Teaching Mathematics and its Applications,* vol. 16, no. 2, 1997, pp. 79-84.

[Wat92] Watt A., Watt M.: *Advanced Animation and Rendering Techniques.* ACM Press, 1992.

[Wei08] Weisstein E. W.: Quaternion, *MathWorld – A Wolfram Web Resource.* http://mathworld.wolfram.com/Quaternion.html, 2008.