Rotation Quaternions, and How to Use Them

D. Rose - May, 2015

Abstract

This paper provides a basic introduction to the use of quaternions in 3D rotation applications. We give a simple definition of quaternions, and show how to convert back and forth between quaternions, axis-angle representations, Euler angles, and rotation matrices. We also show how to rotate objects forward and back using quaternions, and how to concatenate several rotation operations into a single quaternion.

Introduction

Strictly speaking, a quaternion is represented by four elements:

$$\boldsymbol{q} = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 \tag{1}$$

where q_0, q_1, q_2 and q_3 are real numbers, and $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are mutually orthogonal imaginary unit vectors. The q_0 term is referred to as the "real" component, and the remaining three terms are the "imaginary" components. In practice (and for the remainder of this paper), the imaginary notation is implied, and only the four coefficients are used to specify a quaternion, as in equation 2:

$$\boldsymbol{q} = (q_0, q_1, q_2, q_3) \tag{2}$$

Quaternions are a complicated subject. However, in this paper we will restrict ourselves to a subset of quaternions called ***rotation quaternions***. Rotation quaternions are a mechanism for representing rotations in three dimensions, and can be used as an alternative to rotation matrices in 3D graphics and other applications. Using them requires no understanding of complex numbers.

Rotation quaternions are closely related to the axis-angle representation of rotation. We will therefore start with an explanation of the axis-angle representation, and then show how to convert to a quaternion.

Axis-Angle Representation of 3D Rotations

According to Euler's rotation theorem, any 3D rotation (or sequence of rotations) can be specified using two parameters: a unit vector that defines an axis of rotation; and an angle θ describing the magnitude of the rotation about that axis. This is illustrated in Figure 1:

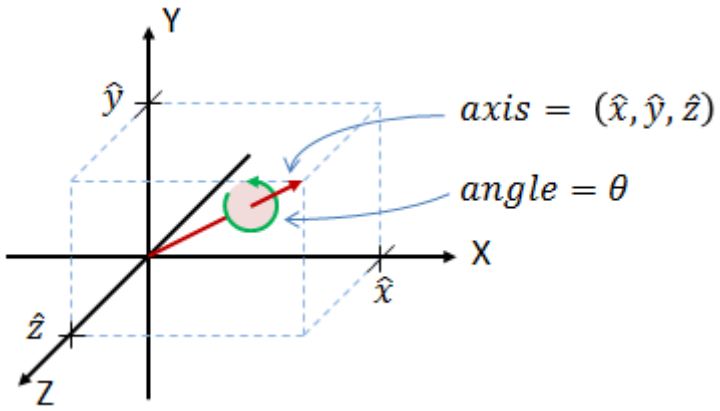

Figure 1: Any 3D rotation can be specified by an axis of rotation and a rotation angle around that axis

An axis-angle rotation can therefore be represented by four numbers as in equation 3:

$$(\theta, \hat{x}, \hat{y}, \hat{z}) \tag{3}$$

where:

$(\hat{x}, \hat{y}, \hat{z})$ is a unit vector that defines the axis of rotation

θ is the amount of rotation around $(\hat{x}, \hat{y}, \hat{z})$

Convert Axis-Angle to Quaternion

A ***rotation quaternion*** is similar to the axis-angle representation. If we know the axis-angle components (θ, x̂, ŷ, ẑ), we can convert to a rotation quaternion q as follows:

$$\boldsymbol{q} = (q_0, q_1, q_2, q_3) \tag{4a}$$

where:

$$q_0 = \cos\left(\frac{\theta}{2}\right) \tag{4b}$$

$$q_1 = \hat{x}\sin\left(\frac{\theta}{2}\right) \tag{4c}$$

$$q_2 = \hat{y}\sin\left(\frac{\theta}{2}\right) \tag{4d}$$

$$q_3 = \hat{z}\sin\left(\frac{\theta}{2}\right) \tag{4e}$$

From these equations we can see that the real term of the quaternion (q_0) is completely determined by the rotation angle, and the remaining three imaginary terms (q_1, q_2 and q_3) are just the three rotation axis vectors scaled by a common factor. One consequence of this representation is that the magnitude of a rotation quaternion (that is, the sum of the squares of all four components) is always equal to one.

Since the axis-angle and quaternion representations contain exactly the same information, it is reasonable to ask why we would bother with the less-intuitive quaternions at all? The answer is that to do anything useful with an axis-angle quantity—such as rotate a set of points that make up some 3D object—we have to perform these trigonometric operations anyway. Performing them ahead of time means that most quaternion operations can be accomplished using only multiplication/division and addition/subtraction, thus saving valuable computer cycles.

Convert Quaternion to Axis-Angle

Given the quaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)$, we can convert back to an axis-angle representation as follows. First, we extract the rotation angle from q_0:

$$\theta = 2\cos^{-1}(q_0) \tag{5}$$

If θ is not zero, we can then find the rotation axis unit vector as follows:

$$(\hat{x}, \hat{y}, \hat{z}) = \left(\frac{q_1}{\sin(\frac{\theta}{2})}, \frac{q_2}{\sin(\frac{\theta}{2})}, \frac{q_3}{\sin(\frac{\theta}{2})}\right) \tag{6}$$

There is one special case in which equation (6) will fail. A quaternion with the value $\boldsymbol{q} = (1,0,0,0)$ is known as the ***identity quaternion***, and will produce no rotation. In this case, equation (5) will produce a rotation angle (θ) of zero, which is what we expect. However, since the axis of rotation is undefined when there is no rotation, equation (6) will generate a divide-by-zero error. Any software implementation should therefore test whether q_0 equals 1.0, and if it does should set $\theta = 0$, and $(\hat{x}, \hat{y}, \hat{z}) = (1, 0, 0)$.

It is worth noting that there are several ways to convert from a quaternion to axis-angle, so don't be too concerned if equations 5 and 6 don't match other sources.

Convert Quaternion to Rotation Matrix

Given the rotation quaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)$, the corresponding rotation matrix is:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{7a}$$

Or equivalently:

$$R = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \tag{7b}$$

Both methods work for all valid unit rotation quaternions, including the identity quaternion.

Convert Rotation Matrix to Quaternion

Given the rotation matrix R:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{8}$$

We can find the equivalent quaternion using two steps.

Step 1: Find the magnitude of each quaternion component. This leaves the sign of each component undefined:

$$|q_0| = \frac{\sqrt{1 + r_{11} + r_{22} + r_{33}}}{4} \tag{9a}$$

$$|q_1| = \frac{\sqrt{1 + r_{11} - r_{22} - r_{33}}}{4} \tag{9b}$$

$$|q_2| = \frac{\sqrt{1 - r_{11} + r_{22} - r_{33}}}{4} \tag{9c}$$

$$|q_3| = \frac{\sqrt{1 - r_{11} - r_{22} + r_{33}}}{4} \tag{9d}$$

Step 2: To resolve the signs, find the largest of q_0, q_1, q_2, q_3 and assume its sign is positive. Then compute the remaining components as shown in the table below. Taking the largest magnitude avoids division by small numbers, which would reduce numerical accuracy.

If q_0 is largest:	If q_1 is largest:	If q_2 is largest:	If q_3 is largest:
$q_1 = \frac{r_{32} - r_{23}}{4q_0}$	$q_0 = \frac{r_{32} - r_{23}}{4q_1}$	$q_0 = \frac{r_{13} - r_{31}}{4q_2}$	$q_0 = \frac{r_{21} - r_{12}}{4q_3}$
$q_2 = \frac{r_{13} - r_{31}}{4q_0}$	$q_2 = \frac{r_{12} + r_{21}}{4q_1}$	$q_1 = \frac{r_{12} + r_{21}}{4q_2}$	$q_1 = \frac{r_{13} + r_{31}}{4q_3}$
$q_3 = \frac{r_{21} - r_{12}}{4q_0}$	$q_3 = \frac{r_{13} + r_{31}}{4q_1}$	$q_3 = \frac{r_{23} + r_{32}}{4q_2}$	$q_2 = \frac{r_{23} + r_{32}}{4q_3}$

The reason the sign is ambiguous is that any given rotation has two possible quaternion representations. If one is known, the other can be found by taking the negative of all four terms. This has the effect of reversing both the rotation angle and the axis of rotation. So for all rotation quaternions, (q_0, q_1, q_2, q_3) and ($-q_0$, $-q_1$, $-q_2$, $-q_3$) produce identical rotations. To convert from a rotation matrix to a quaternion, we must arbitrarily pick one of the two possible answers as described in steps 1 and 2.

Convert Euler Angles to Quaternion

Euler angles are a complicated subject, primarily because there are dozens of mutually exclusive ways to define them. Different authors are likely to use different conventions, often without clearly stating the underlying assumptions, which makes it difficult to combine equations and code from more than one source.

In this paper we will use the following definition of Euler angles.

- Tait-Bryan variant of Euler Angles
- Yaw-pitch-roll rotation order, rotating around the z, y and x axes respectively
- Intrinsic rotation (the axes move with each rotation)
- Active (otherwise known as alibi) rotation (the point is rotated, not the coordinate system)
- Right-handed coordinate system with right-handed rotations

This is a common convention, and most people find it the easiest to visualize. For a more thorough discussion of Euler angles, see this paper.

Given the above definition, we can convert from Euler angles to Quaternion as follows:

$$q_0 = c\left(\frac{u}{2}\right)c\left(\frac{v}{2}\right)c\left(\frac{w}{2}\right) + s\left(\frac{u}{2}\right)s\left(\frac{v}{2}\right)s\left(\frac{w}{2}\right) \tag{10a}$$

$$q_1 = s\left(\frac{u}{2}\right)c\left(\frac{v}{2}\right)c\left(\frac{w}{2}\right) - c\left(\frac{u}{2}\right)s\left(\frac{v}{2}\right)s\left(\frac{w}{2}\right) \tag{10b}$$

$$q_2 = c\left(\frac{u}{2}\right)s\left(\frac{v}{2}\right)c\left(\frac{w}{2}\right) + s\left(\frac{u}{2}\right)c\left(\frac{v}{2}\right)s\left(\frac{w}{2}\right) \tag{10c}$$

$$q_3 = c\left(\frac{u}{2}\right)c\left(\frac{v}{2}\right)s\left(\frac{w}{2}\right) - s\left(\frac{u}{2}\right)s\left(\frac{v}{2}\right)c\left(\frac{w}{2}\right) \tag{10d}$$

where:

u = roll angle
v = pitch angle
w = yaw angle
c() = cosine function
s() = sine function

Equations 10a-d work for all values of Euler angle, including the condition of gimbal lock, where the pitch angle equals +90° or −90°.

Convert Quaternion to Euler Angles

Equations 11a through 11c show how to convert from quaternions to Euler angles:

$$roll = u = \tan^{-1}\left(\frac{2(q_0q_1 + q_2q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) = \text{atan2}[2(q_0q_1 + q_2q_3),\ q_0^2 - q_1^2 - q_2^2 + q_3^2] \tag{11a}$$

$$pitch = v = \sin^{-1}(2(q_0q_2 - q_1q_3)) \quad = \text{asin}\,[2(q_0q_2 - q_1q_3)] \tag{11b}$$

$$yaw = w = \tan^{-1}\left(\frac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right) = \text{atan2}[2(q_0q_3 + q_1q_2),\ q_0^2 + q_1^2 - q_2^2 - q_3^2\,] \tag{11c}$$

Gimbal Lock

Equations 11a through 11c are the general solution for extracting Euler angles from a quaternion. But in the special case where the pitch angle is +90° or −90°, the arguments for 11a and 11c will all be zero, for which the atan2() function is undefined.

This is the dreaded "gimbal lock." It occurs because, at a pitch angle of +90° and −90°, the yaw and roll axes of rotation are aligned with each other in the world coordinate system, and therefore produce the same effect. This means there is no unique solution: any orientation can be described using an infinite number of yaw and roll angle combinations.

To handle the gimbal lock condition, we must first use equation 11b to determine whether the pitch angle is +π/2 or −π/2 radians. Then we set either roll or yaw to zero and solve for the other as follows:

$If\ pitch = +\frac{\pi}{2}$ (radians)	$If\ pitch = -\frac{\pi}{2}$ (radians)
$roll = 0$ $yaw = -2atan2(q_1, q_0)$	$roll = 0$ $yaw = 2atan2(q_1, q_0)$

Note that while Euler angles are susceptible to gimbal lock, quaternions and rotation matrices are not.

Rotating a Point using Quaternions

This section defines quaternion multiplication and inversion, and shows how they are used to perform a rotation.

Quaternion Multiplication

The ***product*** of two quaternions:

$$\boldsymbol{t} = \boldsymbol{rs}$$

$$(t_0, t_1, t_2, t_3) = (r_0, r_1, r_2, r_3) \times (s_0, s_1, s_2, s_3)$$

Is defined as:

$$t_0 = (r_0 s_0 - r_1 s_1 - r_2 s_2 - r_3 s_3) \tag{12a}$$

$$t_1 = (r_0 s_1 + r_1 s_0 - r_2 s_3 + r_3 s_2) \tag{12b}$$

$$t_2 = (r_0 s_2 + r_1 s_3 + r_2 s_0 - r_3 s_1) \tag{12c}$$

$$t_3 = (r_0 s_3 - r_1 s_2 + r_2 s_1 + r_3 s_0) \tag{12d}$$

Quaternion multiplication is associative, but (except for some special cases) is not commutative. Therefore if ***a***, ***b*** and ***c*** are quaternions, then:

$$(\boldsymbol{ab})\boldsymbol{c} = \boldsymbol{a}(\boldsymbol{bc})$$

$$\boldsymbol{ab} \neq \boldsymbol{ba}$$

Quaternion Inversion

The ***inverse*** of a quaternion is obtained by negating the imaginary components:

$$\boldsymbol{q}^{-1} = (\ q_0, -q_1, -q_2, -q_3\ ) \tag{13}$$

Quaternion Rotation

We can rotate a point (*x, y, z*) by the quaternion ***q*** using the following three steps:

Step 1: Convert the point to be rotated into a quaternion by assigning the point's coordinates as the quaternion's imaginary components, and setting the quaternion's real component to zero. If (*x, y, z*) is the point to be rotated, then it is converted to a quaternion as follows:

$$\boldsymbol{p} = (p_0, p_1, p_2, p_3) = (\,0, x, y, z\,) \tag{14}$$

Step 2: Perform the rotation. Quaternion rotation requires two multiplications.

$$\text{For active rotation:} \quad \boldsymbol{p'} = \boldsymbol{q}^{-1}\boldsymbol{pq} \tag{15a}$$

$$\text{For passive rotation:} \quad \boldsymbol{p'} = \boldsymbol{qpq}^{-1} \tag{15b}$$

where:

$\boldsymbol{p}$ contains the point to be rotated (see step 1)
$\boldsymbol{q}$ is a rotation quaternion
$\boldsymbol{q}^{-1}$ is the inverse of $\boldsymbol{q}$
$\boldsymbol{p'}$ contains the coordinates of the rotated point

Active rotation is when the point is rotated with respect to the coordinate system, and passive rotation is when the coordinate system is rotated with respect to the point. The two rotations are opposite from each other.

Note that since quaternion multiplication is associative, it doesn't matter if we perform the multiplications in the order $(\boldsymbol{qp})\boldsymbol{q}^{-1}$ or $\boldsymbol{q}(\boldsymbol{pq}^{-1})$.

Step 3: Extract the rotated coordinates from $\boldsymbol{p'}$:

$$\boldsymbol{p'} = (0, x', y', z') \tag{16}$$

The rotated quaternion $\boldsymbol{p'}$ will have four elements as does any quaternion. However, the real element will always equal zero. The 3D coordinates of the rotated point (x', y', z') are therefore just the imaginary components of $\boldsymbol{p'}$.

Properties of Quaternions

The following are some useful properties of quaternions. Up until now, this paper has discussed only rotation quaternions. However, rotation quaternions are only a subset of all possible quaternions, just as rotation matrices are a subset of all possible 3x3 matrices. The following properties apply to all quaternions unless otherwise specified.

1. The length (magnitude) of a quaternion is $|q| = \sqrt{qq^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$
2. A quaternion is a "unit" quaternion if $|q| = 1$.
3. All rotation quaternions must be unit quaternions.
4. The quaternion $\boldsymbol{q} = (1, 0, 0, 0)$ is the ***identity quaternion***. It represents no rotation. If $\boldsymbol{q}$ is an arbitrary quaternion and $\boldsymbol{i}$ is the identity quaternion, then $\boldsymbol{qi} = \boldsymbol{iq} = \boldsymbol{q}$.
5. The ***conjugate*** of a quaternion is $q^* = (\,q_0, -q_1, -q_2, -q_3\,)$
6. The inverse of a quaternion is $q^{-1} = \frac{q^*}{|q|^2}$. The product of a quaternion and its inverse is the identity quaternion: $\boldsymbol{qq}^{-1} = \boldsymbol{q}^{-1}\boldsymbol{q} = (1, 0, 0, 0)$. Note that for this special case, quaternion multiplication is commutative.
7. For rotation quaternions, the inverse equals the conjugate. So for rotation quaternions, $\boldsymbol{q}^{-1} = \boldsymbol{q}^* = (\,q_0, -q_1, -q_2, -q_3\,)$.
8. Inverting or conjugating a rotation quaternion has the effect of reversing the axis of rotation, which modifies it to rotate in the opposite direction from the original. That is, if a

point is rotated to a new position using $\boldsymbol{q}$, then rotating it again using $\boldsymbol{q}^{-1}$ or $\boldsymbol{q}^{*}$ will return it to its original location.

9. Any given rotation has two possible quaternion representations. If one is known, the other can be found by taking the negative of all four terms. This has the effect of reversing both the rotation angle and the axis of rotation. So if $\boldsymbol{q}$ is a rotation quaternion, then $\boldsymbol{q}$ and $-\boldsymbol{q}$ will produce the same rotation.
10. A rotation of $\boldsymbol{q}_a$ followed by a rotation of $\boldsymbol{q}_b$ can be combined into the single rotation $\boldsymbol{q}_c = \boldsymbol{q}_b\boldsymbol{q}_a$. This can be extended to an arbitrary number of rotations. Note that the order matters (because quaternion multiplication is not commutative).
11. Quaternion multiplication is associative: $(\boldsymbol{ab})\boldsymbol{c} = \boldsymbol{a}(\boldsymbol{bc})$
12. Quaternion multiplication is not commutative: $\boldsymbol{ab} \neq \boldsymbol{ba}$

Related Pages

Rotations in Three-Dimensions: Euler Angles and Rotation Matrices. Describes a commonly used set of Tait-Bryan Euler angles, and shows how to convert from Euler angles to a rotation matrix and back.

Rotation Conversion Tool. An on-line utility that converts between Euler Angles, Quaternions, Axis-Angle, and Rotation Matrix representations.

Comments and error reports may be sent to the following address. We may post comments of general interest. Be sure to identify the page you are commenting on.

EngineeringNotes@DancesWithCode.net

Home