Rotation in 3-dimensions using Quaternions and Euler angles

Research Question:
How are quaternions and Euler angles used in rotating in three dimensions and what are their benefits and drawbacks?

Subject Area: Mathematics
Word Count: 3610

Table of Contents

Introduction

In the current era of technological advancements, copious amount of machines are computerized. A large percentage of these computerized machines also use rotations in order to configure itself. For example, machines in manufacturing industry have rotations encoded in them which allows them to move properly without disruptions. Drones are completely reliant on rotations to maintain their stability and not crash. More and more video games are also made in 3D meaning that it also requires three dimensional rotation in order to provide a more realistic and immersive gameplay. Therefore, rotations are a key part of many industries and sectors in this technologically oriented generation. For the same reason, there have been a lot of progress in the math behind rotations throughout the centuries that led to better and more stable rotation methods that allow for proper rotation in three dimensions.

Rotations in two dimensions can be easily explained. In simple terms, two-dimensional rotation is picking two points on a two-dimensional figure and moving one of the points while keeping the other point stationary, while the two-dimensional object moves with the mobile point. However, the distance between the points must remain the same, such that one is considered as the center of a circle and the other, a point moving along the circumference of the circle. Rotation in three-dimension can be similarly explained, however, instead of the points being related to a circle, it is a sphere in this case. This definition is vague and lacking a solid foundation. Hence, this essay will explain the mathematics behind methods of rotation in three-dimensions, which can be understood in much more depth.

There are various methods of rotating in three dimensions. Two of these methods that this essay will focus on are the two main methods of rotations used in computer graphics, robotics and game development. These methods are using quaternions and Euler angles.

Since, both topics are out of syllabus and not widely taught or heard about outside the world of rotations, resources were very limited. However, some resources such as *3D Game*

Engineering Programming[1] and the research paper from Gregory G. Slabaugh called *Computing Euler angles from a rotation Matrix*[2], both gave detailed information on both topics and how the mathematics behind either rotational methods work.

Quaternions

Quaternions are one of the most widely used and accepted methods of rotation in computer graphics, aeronautics, robotics and many other areas where three dimensional rotation is required. Quaternions were discovered by Sir William Rowan Hamilton[3], who was an Irish mathematician and astronomer. He was very keen on the two dimensional geometric significance of complex numbers in Mathematics and he wanted to extend this concept onto three dimensions. Complex numbers introduced the concept of "unit imaginary number", i, which was defined as $i^2 = -1$. A complex number can be written in the form $a + bi$ where a and b are real numbers and bi is the imaginary part of the number. It was later discovered that the relationship between dimension and number of parts in the number system is not a linear one. Therefore, after multiple trials, Hamilton came up with the general formula for quaternions which is given in the form $a + bi + cj + dk$ where $a, b, c, d \in \mathbb{R}$ and $i, j, k \in \mathbb{I}$.

However, in order for this new number system to work, a few axioms had to be defined. The defining rules of quaternions, known as "Fundamental Formula for Quaternion Multiplication[4]", that Hamilton defined is as such,

$$i^2 = j^2 = k^2 = ijk = -1$$

[1] Jeremiah. *3D Game Engine Programming*. 25 June 2012. 2019 August 23. <https://www.3dgep.com/understanding-quaternions/>.

[2] Slabaugh, Gregory G. *Computing Euler angles from a rotation matrix*. n.d. Document. 05 August 2019. <http://www.gregslabaugh.net/publications/euler.pdf>.

[3] Wilkins, David. *Sir William Rowan Hamilton*. 29 April 2019. Article. 23 August 2019. <https://www.brit annica.com/biography/William-Rowan-Hamilton>.

[4] *Weisstein, Eric W. Quaternion. n.d. Article. 23 August 2019. <http://mathworld.wolfram.com/Quaternion.html>.*

One feature of the three imaginary numbers to be noticed is that even though i, j, and k all have unit value of $\sqrt{-1}$, they are different square roots each mutually perpendicular to each other. This feature is more prominent while graphically representing quaternions with i, j, and k as the three axes, in a similar way to which complex numbers are represented graphically (shown in Diagram 2 (page 11)).

An important property of quaternion is that its multiplication is noncommutative. It means that the order in which the imaginary numbers are multiplied affects the results. For example, $ij \neq ji$. Hence, the axioms[5] defined by Hamilton were as such,

$$ij = k = -ji$$

$$ki = j = -ik$$

$$jk = i = -kj$$

With these axioms defined, the rotations in three dimensions using quaternions can be understood.

Rotation using Quaternions

In order to understand how quaternions can be used in rotating three dimensional object, one must initially understand two dimensional rotation using complex numbers. Rotation in two dimensions use unit vectors in order to avoid the magnitude of the vector being rotated, from being changed. Given that the vector $v = [x, y]$ is a unit vector in a two dimensional plane, the x value is given by $cos\theta$ and the y value is given by $sin\theta$. Diagram 1 visualizes why it is so,

[5] Mason, Matthew T. *Lecture 8. Quaternions*. 2013. Presentation. 20 August 2019. <http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/lectures/Lecture8.pdf>.

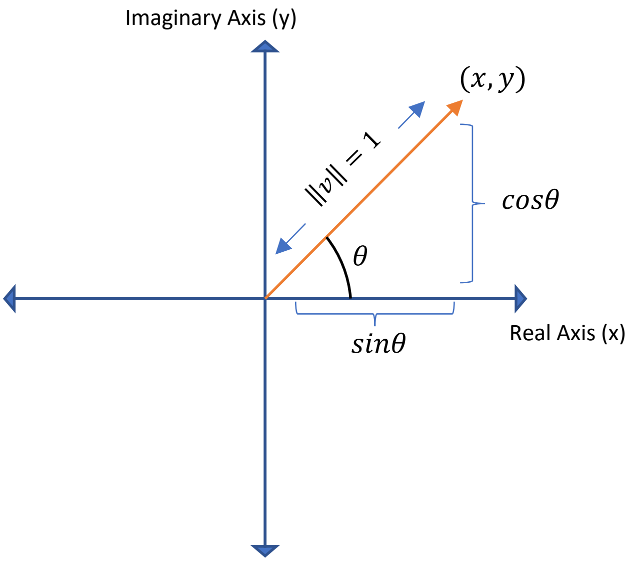

Diagram 1: Argand Diagram

A two dimensional point, such as (x, y) in Diagram 1, can be rotated by a given angle by using complex numbers. Given that the point is given in the coordinate format (x, y) and that it is to be rotated by an angle θ, the following equation can be used to give the coordinates of the final point after rotation, (x', y'),

$$(x' + iy') = (x + iy)(cos\theta + isin\theta)$$

Another notation with coordinates is as follows,

$$(x', y') = (x, y)(cos\theta, sin\theta)$$

The $(cos\theta, sin\theta)$ in the above equation, is known as the *rotor*. The resultant rotated point will have the general formula,

$$(x', y') = (xcos\theta - ysin\theta, xsin\theta + ycos\theta)$$

For example, assume an arbitrary point (3, 1). From intuition, it can be said that rotating the point 180° about the origin will result in the point (-3, -1).

Using the above equation, the rotated point can be found as such,

$$(x', y') = (xcos\theta - ysin\theta, xsin\theta + ycos\theta)$$

$$(x', y') = (3cos180 - 1sin180, 3sin180 + 1cos180)$$

$$(x', y') = ((-3 - 0), (0 + -1))$$

$$\boldsymbol{(x', y') = (-3, -1)}$$

As seen by the above calculation, the results obtained were the expected results.

A very similar method as to the one shown above is used in rotating using quaternions. Rotating in 3-dimensions means that there are three axes and three coordinates. However, quaternions are 4-dimensional points and hence have four axes and four coordinates – one scalar part and 3 imaginary parts. To use quaternions in 3-dimensional rotation, we only need 3 imaginary axes, and hence the real part is considered as 0. Such quaternions are known as pure quaternions. Consider a point on a 3-dimensional space to be rotated as (b, c, d). Then the corresponding pure quaternion can be written as $p = 0 + bi + cj + dk$.

As previously mentioned, any complex number $z = a + bi$, can be written as $z = [a, b]$. Hence, any quaternion q can also be written as $q = [a, \hat{q}]$ where $\hat{q}$ is the vector $[b, c, d]$. It was previously stated that the general formula that defines a unit complex number is given by $z = [cos\theta, sin\theta]$. Using the same principle, a unit quaternion can be defined as $q = [cos\theta, sin\theta\hat{q}]$. Since we stated that $[cos\theta, sin\theta]$ is the rotor used for two dimensional rotation, $[cos\theta, sin\theta\hat{q}]$ can be said to be the rotor for three dimensional rotation. In the rotor, $\hat{q}$ is the axis of rotation. There is no such factor as axis of rotation in two dimensional rotation since a given point can only be rotated about a point. In three dimensions, θ is defined as the value of how much the given point must be rotated with correspondence to the axis of rotation in the anti-clockwise direction and hence the factor of axis of rotation $(\hat{q})$ must be defined.

Therefore, the equation for the rotated vector p', obtained by rotating a three dimensional vector p along an axis of rotation of $\hat{q}$, is given by the following formula,

$$\boldsymbol{p'} = \boldsymbol{qp}$$

This shows that rotation with quaternions involve multiplying them together and hence quaternion multiplication must be thoroughly understood. Given two quaternions,

$q_1 = (a_1 + b_1 i + c_1 j + d_1 k)$ and $q_2 = (a_2 + b_2 i + c_2 j + d_2 k)$,

$(q_1)(q_2)$ is as shown below,

$$(a_1 + b_1 i + c_1 j + d_1 k)(a_2 + b_2 i + c_2 j + d_2 k)$$

$$= a_1a_2 + a_1b_2i + a_1c_2j + a_1d_2k + a_2b_1i - b_1b_2 + b_1c_2k - b_1d_2j + a_2c_1j - b_2c_1k - c_1c_2 + c_1d_2i + a_2d_1k + b_2d_1j - c_2d_1i - d_1d_2$$

$$= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 + a_1b_2i + a_2b_1i + c_1d_2i - c_2d_1i + a_1c_2j - b_1d_2j + a_2c_1j + b_2d_1j + a_1d_2k + b_1c_2k - b_2c_1k + a_2d_1k$$

$$= (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + (a_1b_2 + a_2b_1 + c_1d_2 - c_2d_1)i + (a_1c_2 + a_2c_1 + b_2d_1 - b_1d_2)j + (a_1d_2 + a_2d_1 + b_1c_2 - b_2c_1)k$$

Since, the Quaternion $(a_1 + b_1 i + c_1 j + d_1 k)$ can be represented as a point on a four dimensional plane with one real axis and three imaginary axes, i, j and k, it can be simplified to quadruplets of coordinates (a_1, b_1, c_1, d_1). Therefore, multiplication of two quaternions given in coordinates can be calculated as such,

$$(a_1, b_1, c_1, d_1)(a_2, b_2, c_2, d_2)$$

$$= (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2, a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2, a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2, a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)$$

This multiplication is long and time consuming and hence, taking out like terms and simplifying it further, the same result of multiplying two quaternions can be represented using dot and cross products of vectors[6]. In order to do so, the quaternions must be separated into two components, the real and imaginary part. Say v_1 represents the imaginary part and therefore, $v_1 = (b_1, c_1, d_1)$. Hence, the quaternion (a_1, b_1, c_1, d_1) can be written as (a_1, v_1).Therefore, quaternions multiplication can be shortened to as follows,

$$(a_1, v_1)(a_2, v_2) = (a_1a_2 - v_1 \cdot v_2, a_1v_2 + a_2v_1 + v_1 \times v_2)$$

The above equation is a simplification that involves dot and cross products of vectors. Expanding the right hand side of the above equation will result in the same long equation that was calculated before, while multiplying the two quaternions as a whole. The expansion is as follows,

$$v_1 \cdot v_2 = (b_1, c_1, d_1) \cdot (b_2, c_2, d_2)$$

$$= b_1b_2 + c_1c_2 + d_1d_2$$

$$\therefore a_1a_2 - v_1 \cdot v_2 = a_1a_2 - (b_1b_2 + c_1c_2 + d_1d_2)$$

$$= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 \longrightarrow$$ **Scalar part of the quaternion**

$$v_1 \times v_2 = \begin{pmatrix} b_1 \\ c_1 \\ d_1 \end{pmatrix} \times \begin{pmatrix} b_2 \\ c_2 \\ d_2 \end{pmatrix}$$

$$= \begin{pmatrix} c_1d_2 - c_2d_1 \\ d_1b_2 - d_2b_1 \\ b_1c_2 - b_2c_1 \end{pmatrix}$$

$$a_1v_2 = \begin{pmatrix} a_1b_2 \\ a_1c_2 \\ a_1d_2 \end{pmatrix}$$

[6] Baker, Martin John. *Maths - Quaternion Arithmetic*. 2017. Document. <https://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternio ns/arithmetic/index.htm>.

$$a_2 v_1 = \begin{pmatrix} a_2 b_1 \\ a_2 c_1 \\ a_2 d_1 \end{pmatrix}$$

$$\therefore a_1 v_2 + a_2 v_1 + v_1 \times v_2 = \begin{pmatrix} a_1 b_2 \\ a_1 c_2 \\ a_1 d_2 \end{pmatrix} + \begin{pmatrix} a_2 b_1 \\ a_2 c_1 \\ a_2 d_1 \end{pmatrix} + \begin{pmatrix} c_1 d_2 - d_1 c_2 \\ d_1 b_2 - b_1 d_2 \\ b_1 c_2 - c_1 b_2 \end{pmatrix}$$

$$= \begin{pmatrix} a_1 b_2 + a_2 b_1 + c_1 d_2 - d_1 c_2 \\ a_1 c_2 + a_2 c_1 + d_1 b_2 - b_1 d_2 \\ a_1 d_2 + a_2 d_1 + b_1 c_2 - c_1 b_2 \end{pmatrix} \longrightarrow$$ **Imaginary or Vector part of the quaternion**

Writing both the parts in the above calculation in coordinate form will result in a 4-dimensional coordinate as shown below,

$$(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2, a_1 b_2 + a_2 b_1 + c_1 d_2 - c_2 d_1, a_1 c_2 + a_2 c_1 + b_2 d_1 - b_1 d_2, a_1 d_2 + a_2 d_1 + b_1 c_2 - b_2 c_1)$$

As seen, the result of expanding the simplified equation gives the same result from the long multiplication method. With the use of these imaginary numbers and axioms, quaternions finally became useful in rotations in three dimensions.

The equation $\boldsymbol{p'} = \boldsymbol{qp}$, works for the simplest case for rotating with quaternions. This case is when the vector being rotated is orthogonal to the axis of rotation. Say that the 3-dimensional vector, $\hat{p} = (0,0,2)$ (or pure quaternion $p = (0,0,0,2)$) is to be rotated at an angle of 180° along the i-axis. Since the unit quaternion that lies along the i-axis has the equation $(0,1,0,0)$, the axis of rotation can be said to be $\hat{q} = 0 + 1i + 0j + 0k$ or simply $\hat{q} = i$. Therefore, the rotated point p' is given by the following formula

$$p' = qp$$

$$p' = [cos\theta, \hat{q}sin\theta][0, \hat{p}]$$

This equation can be quickly simplified using the shorthand multiplication method discussed previously to formulate the equation shown below,

$$p' = [(cos\theta)(0) - \hat{q}sin\theta \cdot \hat{p}, \hat{p}cos\theta + (0)(\hat{q}sin\theta) + \hat{q}sin\theta \times \hat{p}]$$

$$p' = [(cos180)(0) - \hat{q}sin180 \cdot \hat{p}, \hat{p}cos180 + (0)(\hat{q}sin180) + \hat{q}sin180 \times \hat{p}]$$

The parts of the above equation are solved below,

$$(cos180)(0) - \hat{q}sin180 \cdot \hat{p} = (cos180)(0) - \hat{q}sin180 \cdot \hat{p} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = 0$$

$$\hat{p}cos180 = -1 \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix}$$

$$(0)(\hat{q}sin180) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\hat{q}sin180 \times \hat{p} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Therefore, substituting all the calculations above into the previous equation results in an equation as follows,

$$p' = \left[0, \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right]$$

$$\boldsymbol{p'} = [\mathbf{0}, \mathbf{0}, \mathbf{0}, -\mathbf{2}]$$

Therefore, from the previous calculation, the result of rotating the vector $p = [0,0,0,2]$ 180° about the z-axis results in the vector $p' = [0,0,0,-2]$. This rotation is visualized in Diagram 2 below.

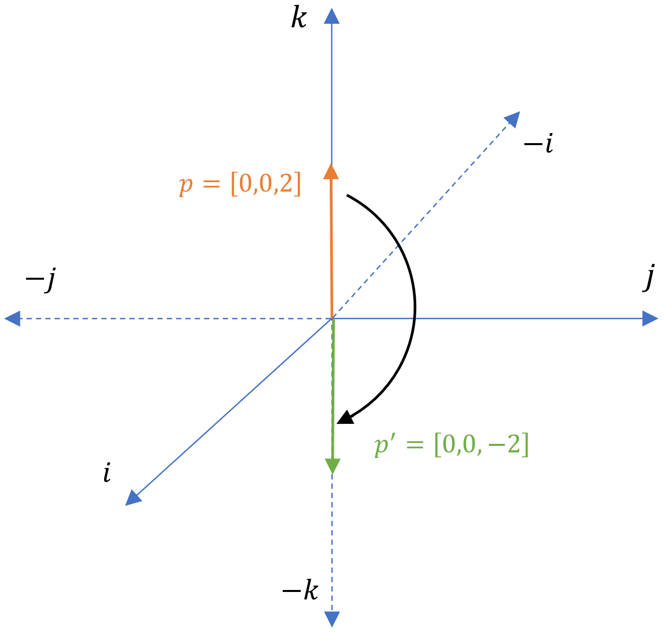

Diagram 2: Rotation of p *to* p′

However, this method of quaternion rotation only takes the simplest method into consideration. There could be cases where the axis of rotation is not orthogonal to the vector being rotated. In those cases a slightly different formula must be used. The formula that Hamilton came up with was $p' = qpq^{-1}$. This equation is only slightly different from the above equation in that it is further multiplied by the inverse of q. The inverse of q or q^{-1} is defined by the following equation,

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

In the above equation, q^* is called the quaternion conjugate. A quaternion conjugate is another quaternion with the same scalar or real part and negative vector or imaginary part. Therefore, for the general formula for the rotor $q = [cos\theta, vsin\theta]$, the conjugate will be $q^* = [cos\theta, -vsin\theta]$. $\|q\|$ is the magnitude of the quaternion q. It can be found using Pythagoras

Theorem, $||q|| = \sqrt{cos\theta^2 + (bsin\theta)^2 + (csin\theta)^2 + (dsin\theta)^2}$, since Pythagoras Theorem works in all dimensions. However, for unit quaternions with magnitude 1, $q^{-1} = q^*$.

Using the definition for q used previously, the rotation would result in the vector being rotated twice as much. In order to compensate this error, the angle was divided by 2. Hence, the rotor for rotations, where the axis of rotation is not orthogonal to the vector being rotated, is $q = \left[\cos\frac{\theta}{2}, \hat{q}sin\frac{\theta}{2}\right]$.

For example, given the vector $p = [5,6,7]$ to be rotated by an angle of 240° about the axis of rotation of $V = [2,2,2]$, the rotated point, p' can be calculated as follows. Firstly, we need to normalize the axis to ensure that multiplying by it does not affect the magnitude of p. In order to do this, each value in the coordinate must be divided by the magnitude of the vector as shown below,

$$||V|| = \sqrt{2^2 + 2^2 + 2^2} = \sqrt{12}$$

$$v_{norm} = \left[\frac{2}{\sqrt{12}}, \frac{2}{\sqrt{12}}, \frac{2}{\sqrt{12}}\right]$$

In order to rotate this vector, it must be written as a quaternion. Assume v_{quat} is the unit quaternion. Therefore,

$$v_{quat} = \left[\cos\frac{\theta}{2}, v_{norm}\sin\frac{\theta}{2}\right]$$

$$v_{quat} = \left[\cos 120, \frac{2}{\sqrt{12}}\sin 120, \frac{2}{\sqrt{12}}\sin 120, \frac{2}{\sqrt{12}}\sin 120\right]$$

$$v_{quat} = \left[-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]$$

Now, the inverse of v_{quat} or v_{quat}^{-1} must be calculated. Since, v_{quat} is a unit quaternion, v_{quat}^{-1} is the conjugate of v_{quat}. The conjugate of v_{quat} is $v_{quat}^{*} = \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$. Hence, the following formula can be used to calculate p',

$$p' = v_{quat} p v_{quat}^{*}$$

Since, quaternion multiplication is non-commutative, the multiplication must be done in the same order as shown in the formula. Therefore,

$$v_{quat} p = \left[-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] [0,5,6,7]$$

$$= \left[0 - \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] \cdot [5,6,7], -\frac{1}{2}[5,6,7] + \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] \times [5,6,7]\right]$$

Calculating part by part,

$$\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] \cdot [5,6,7] = 9$$

$$-\frac{1}{2}[5,6,7] = [-2.5, -3, -3.5]$$

$$\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right] \times [5,6,7] = [0.5, -1, 0.5]$$

Combining all the parts,

$$v_{quat} p = \left[0 - 9, [-2.5, -3, -3.5] + [0.5, -1, 0.5]\right]$$

$$v_{quat} p = [-9, -2, -4, -3]$$

Therefore,

$$v_{quat} p v_{quat}^{*} = [-9, -2, -4, -3] \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$$

$$= \left[\frac{9}{2} - [-2, -4, -3] \cdot \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right], -9\left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right] + -\frac{1}{2}[-2, -4, -3]\right.$$

$$\left. + [-2, -4, -3] \times \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]\right]$$

Again, calculating by parts,

$$\frac{9}{2} - [-2, -4, -3] \cdot \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right] = 0$$

$$-9\left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right] = [4.5, 4.5, 4.5]$$

$$-\frac{1}{2}[-2, -4, -3] = [1, 2, 1.5]$$

$$[-2, -4, -3] \times \left[-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right] = [0.5, 0.5, -1]$$

Combining all the parts,

$$v_{quat} p v_{quat}^{*} = \left[0, [4.5, 4.5, 4.5] + [1, 2, 1.5] + [0.5, 0.5, -1]\right]$$

$$= [0, 6, 7, 5]$$

Therefore, $\boldsymbol{p'} = [\mathbf{0}, \mathbf{6}, \mathbf{7}, \mathbf{5}]$**.**

Euler Angles

Euler angles are another method of rotating in three dimension. Instead of one single rotation over an arbitrary axis using quaternions, Euler angle rotates a given angle around each of the three axis in a specific order. To be more specific, the three axes x, y and z are given corresponding angles α, β and γ. Diagram 3 below gives a visual representation of this,

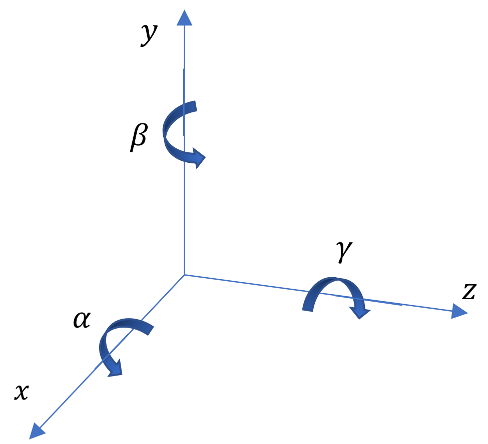

Diagram 3: Euler angles and corresponding Axes

There are two main methods of rotating using Euler angles. The methods are using pure Euler angles and the Tait-Bryan method. The math behind both subdivisions of rotation using Euler angles are the same, however, pure Euler angles start and end the rotation about the same axis. For example, the possibilities are XYX, XZX, YXY, YZY, ZXZ and ZYZ[7] where X, Y and Z represent x-, y- and z-axis respectively. Trait-Bryan method rotates about each axis once. For example, the possibilities are XYZ, XZY, YXZ, YZX, ZXY and ZYX[8]. If the vector to be rotated is initially rotated about the x-axis by an angle α, it rotates the other two axes along with it. The next rotation can be rotated about the newly rotated axes or about the initial position of the axes. These two methods are conveniently called rotated axes and fixed axes respectively. The Tait-Bryan method uses the concept of rotated axes for its rotations. Since, the most commonly used method is using Tait-Bryan angles, it will be the focus of this essay. There are defined rotation matrices[9] for each of the axes and they are shown below,

For rotating about the x-axis with angle α,

[7] Allgeuer, Philipp and Sven Behnke. *Fused Angles and the Deficiencies of Euler Angles*. October 2018. 12 October 2019. <https://www.ais.uni-bonn.de/papers/IROS_2018_Allgeuer.pdf>.

[8] Ibid.

[9] Slabaugh, Gregory G. *Computing Euler angles from a rotation matrix*. n.d. Document. 05 August 2019. <http://www.gregslabaugh.net/publications/euler.pdf>.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

For rotating about the y-axis with angle β,

$$R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix}$$

For rotating about z-axis with angle γ,

$$R_z(\gamma) = \begin{bmatrix} cos\gamma & sin\gamma & 0 \\ -sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These matrices have to be multiplied in the reverse order of desired rotation to obtain a single rotation matrix that includes all the rotation. For example, given that a vector is to be rotated an angle of α about the x-axis, then by an angle of β about the y-axis and finally an angle of γ about the z-axis. Then the rotation matrix for this specific rotation will be given by the formula $R_{xyz} = R_z(\gamma)R_y(\beta)R_x(\alpha)$. This calculation must be done in the exact order as shown below due to the non-commutative property of matrices,

Firstly $R_y(\beta)$ must be multiplied by $R_x(\alpha)$,

$$R_y(\beta)R_x(\alpha) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

$$= \begin{bmatrix} cos\beta & sin\beta sin\alpha & sin\beta cos\alpha \\ 0 & cos\alpha & -sin\alpha \\ -sin\beta & cos\beta sin\alpha & cos\beta cos\alpha \end{bmatrix}$$

Then, $R_z(\gamma)$ must be multiplied by the matrix calculated above in order to obtain the rotation matrix,

$$R_{xyz} = \begin{bmatrix} cos\gamma & sin\gamma & 0 \\ -sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\beta & sin\beta sin\alpha & sin\beta cos\alpha \\ 0 & cos\alpha & -sin\alpha \\ -sin\beta & cos\beta sin\alpha & cos\beta cos\alpha \end{bmatrix}$$

$$= \begin{bmatrix} cos\gamma cos\beta & sin\alpha sin\beta cos\gamma - cos\alpha sin\gamma & cos\gamma sin\beta cos\alpha + sin\gamma sin\alpha \\ cos\beta sin\gamma & sin\alpha sin\beta sin\gamma + cos\alpha sin\gamma & cos\alpha sin\beta sin\gamma - sin\alpha cos\gamma \\ -sin\beta & sin\alpha cos\beta & cos\alpha cos\beta \end{bmatrix}$$

This matrix can then be multiplied by any given vector that is to be rotated in the order of XYZ. This is the most common rotational convention and is called XYZ convention.

Rotation using Euler Angles

The sample rotation with the quaternions that was shown as an example can also be done using Euler angles. Given that a vector $v = [0, 0, 2]$ is to be rotated by an angle of 180° about the x-axis, the variables in the above rotation matrix can be substituted in with the following values, $\alpha = 180°$, $\beta = 0°$ and $\gamma = 0°$. Therefore, the rotation matrix will have the following values,

$$\begin{bmatrix} cos\gamma cos\beta & sin\alpha sin\beta cos\gamma - cos\alpha sin\gamma & cos\gamma sin\beta cos\alpha + sin\gamma sin\alpha \\ cos\beta sin\gamma & sin\alpha sin\beta sin\gamma + cos\alpha sin\gamma & cos\alpha sin\beta sin\gamma - sin\alpha cos\gamma \\ -sin\beta & sin\alpha cos\beta & cos\alpha cos\beta \end{bmatrix}$$

$$= \begin{bmatrix} cos0cos0 & sin180sin0cos0 - cos180sin0 & cos0sin0cos180 + sin0sin180 \\ cos0sin0 & sin180sin0sin0 + cos180sin0 & cos180sin0sin0 - sin180cos0 \\ -sin0 & sin180cos0 & cos180cos0 \end{bmatrix}$$

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

Now the vector $p = [0, 0, 2]$ can be multiplied by the rotation matrix above in order to obtain the rotated point p'. Therefore,

$$p' = pR_{xyz}$$

$$p' = [0 \quad 0 \quad 2] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$p' = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{-2}]$$

Even though quaternions and Euler angles are both valid and useful methods of rotating in three dimensions, they both have drawbacks and benefits. Diagram 4 below represents the rotation example above,

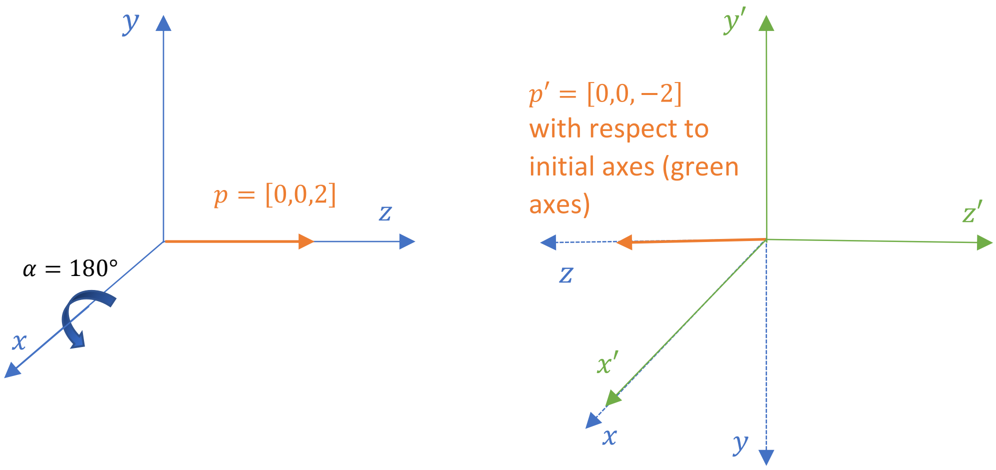

Diagram 4: Visualization of sample rotation using Euler angles

In Diagram 4 above, the graph on the left represents the initial position of the axes and vector before rotation. The blue axes on the graph on the right represent the initial axes after being rotated 180° about the x-axis. The green axes on the represent the initial axes in its initial position and was shown to show how vector $p = [0,0,2]$ becomes $p' = [0,0,-2]$ with respect to the initial position of the axes.

Benefits and Drawbacks of either methods of rotation

Of the two methods of rotating objects in three dimensional space, both have their own advantages and disadvantages. A huge advantage brought by quaternions that is also a huge drawback of Euler angles is the problem of Gimbal Lock. As the name suggests, the problem occurs when rotating objects along Gimbals instead of linear axes. Gimbals, as shown in Diagram 5, are rings that rotate in a fixed direction that replace axes. Gimbal lock is when the gimbals lose a degree of rotation when two of the axes align with each other. The reason for

Gimbal lock being a problem is due to how Euler angles work. Since, Euler angles use rotation matrices in order to execute rotation, it is non commutative as mentioned before. This feature of Euler angle is a huge contributor to the Gimbal lock.

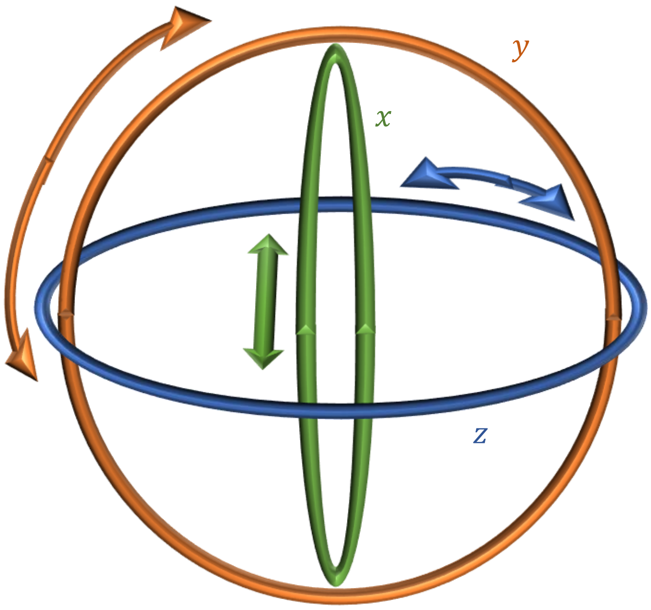

Diagram 5: The Three Gimbal Axes and their direction of rotation

As shown in Diagram 5, the green ring rotates along the x-axis, the orange ring rotates along the y-axis and the blue ring rotates along the z-axis. Since, x is the inner most axis, it can rotate without affecting any of the other axes. When the y axis rotates, it rotates the x axis along with it. In the same way, when the z axis rotates, it rotates both the other two axes along with it. Therefore, say the y axis is rotated 90° clockwise or anticlockwise from its initial position shown in Diagram 5. Following that rotation would result in the x axis and the z axis being parallel to or aligned with each other, as shown in Diagram 6,

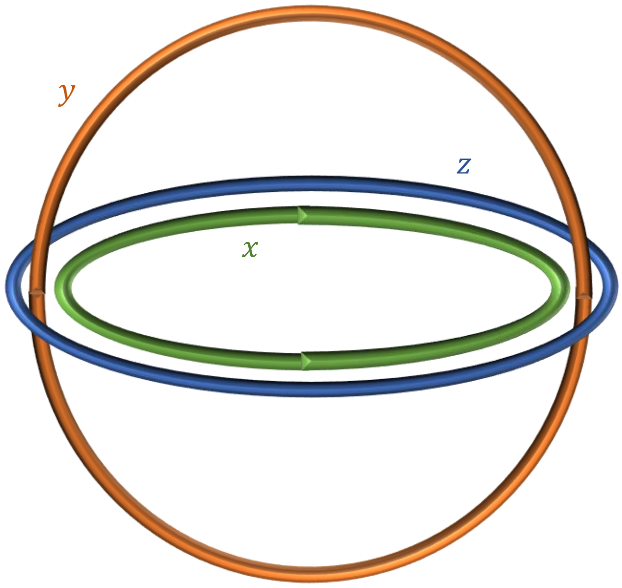

Diagram 6: Two Axes Parallel to each other causing Gimbal Lock

Now, from Diagram 6, it can be seen that rotating the x axis and rotating the z axis would result in the same rotation. Therefore, now the rotation only has two degrees of freedom, one from rotation along y axis and one from rotations along either x or z axis. This is what is known as a Gimbal lock. The benefit of quaternions is that the axes rotate independent of each other. That is, in the case of Euler angles, as seen above in Diagrams 5 and 6, rotating the z axis will rotate the x and y axes along with it. Since in quaternion rotation a given point or vector is rotated with respect to the axes rather than the whole axes being rotated, it can avoid Gimbal lock.

A benefit of Euler angles over quaternions is the ability to comprehend it. As seen by the sample rotation for both quaternions and Euler angles, the mathematics behind rotation using Euler angles is simpler and easily comprehensible to a high-schooler, and on the opposite end of the spectrum, it takes time and in-depth and widespread research in order to properly comprehend the concept of rotating with quaternions. Euler angle only uses basic trigonometry and matrix multiplication. However, in order to understand how to use quaternions in three dimensional rotation, first the concept of quaternion as a four dimensional number system must be understood. There are also a lot of axioms and defined rules that need to be learned before completely understanding how to use quaternions for rotating in three dimensions. The factor of Euler angle involving 3x3 matrix multiplication increases the storage space required to store the rotations when compared to quaternion rotations, since they only use a 4-part number system.

However, even though counter intuitive, once understood, quaternions are much easier to compute compared to Euler angles, due to the fact that the simplified multiplication for quaternions lends itself to a much quicker calculation of the rotation. This is also a huge factor while storing rotations in game development or animations since when rotating using Euler angles, rotating along each axis must be individually considered. This is far more information than rotating about a single arbitrary axis. Since the concept of rotating in 3-dimensions using four dimensional number system is quite hard to understand for people with no specific knowledge about quaternions, Euler angles are used to represent rotations in the user interface of game development software, such as Unity[10], as shown in Diagram 7. However, the rotations are stored as quaternions in order to optimize the storage space used.

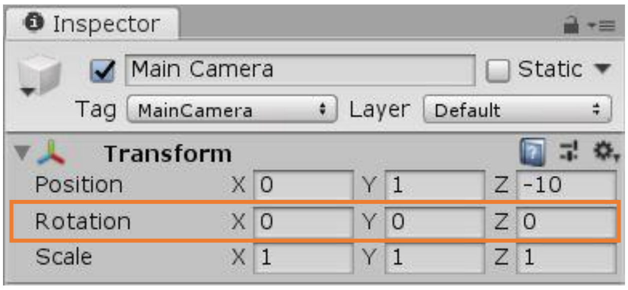

Diagram 7: Rotation in Unity Game Development Software

Another factor that brings up comparison between Euler angles and quaternions is interpolation. Interpolation is smoother while using quaternions compared to Euler angles and hence makes quaternions more preferential in animations and game development. However, the math behind interpolation using quaternions and Euler angles are too complicated and would require an essay of its own. Therefore, considering that the focus of the essay is not on how interpolation with quaternions and Euler angles work, it is not discussed any further.

[10] Unity Technologies. (2019). *Unity*.

Conclusion

Therefore, this essay has answered the research question – **“How are quaternions and Euler angles used in rotation in three dimensions and what are the benefits and drawbacks of both methods?”.** Quaternions and Euler angles have their own “rotors” that allow for them to be used in rotating in three dimension. The significant difference in the math behind both methods of rotations directly correlates to their drawbacks and benefits discussed. However, in the current digitalized society where animations and rotations are prevalent throughout, knowing both methods of rotations can give an upper hand for programmers. As seen in the example of Unity Technologies’ game development software, knowledge of both methods can lead to a great advantage in software development.

Bibliography

Jeremiah. *3D Game Engine Programming*. 25 June 2012. 2019 August 23. <https://www.3dgep.com/understanding-quaternions/>.

Slabaugh, Gregory G. *Computing Euler angles from a rotation matrix*. n.d. Document. 05 August 2019. <http://www.gregslabaugh.net/publications/euler.pdf>.

Wilkins, David. *Sir William Rowan Hamilton*. 29 April 2019. Article. 23 August 2019. <https://www.brit annica.com/biography/William-Rowan-Hamilton>.

Weisstein, Eric W. Quaternion. n.d. Article. 23 August 2019. <http://mathworld.wolfram.com/Quaternion.html>.

Mason, Matthew T. *Lecture 8. Quaternions*. 2013. Presentation. 20 August 2019. <http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/lectures/Lecture8.pdf>.

Baker, Martin John. *Maths - Quaternion Arithmetic*. 2017. Document. <https://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/arithmetic/index.htm>.

Allgeuer, Philipp and Sven Behnke. *Fused Angles and the Deficiencies of Euler Angles*. October 2018. 12 October 2019. <https://www.ais.uni-bonn.de/papers/IROS_2018_Allgeuer.pdf>.

Slabaugh, Gregory G. *Computing Euler angles from a rotation matrix*. n.d. Document. 05 August 2019. <http://www.gregslabaugh.net/publications/euler.pdf>.

Unity Technologies. (2019). *Unity*.