CS461 : WRITTEN ASSIGNMENT - 3

Devaishi Tiwari (170101021)

QUATERNIONS in Computer Graphics

1. Brief Introduction

Similar to complex numbers, there are two different ways to understand and define quaternions, the **geometric** and the **algebraic** approach.

1.1 Algebraic Definition

While the complex numbers are obtained by adding the element i to real numbers which satisfies $i^2 = 1$, quaternions are obtained by adding the elements **i, j,** and **k** to the real numbers. Here i, j and k satisfy the following relations,

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

Every quaternion is a real **linear combination** of basis quaternions 1, i, j, and k, that means every quaternion can be uniquely expressible in the form $\mathbf{a + b}\boldsymbol{i}\mathbf{ + c}\boldsymbol{j}\mathbf{ + d}\boldsymbol{k}$.

1.2 Geometric Definition

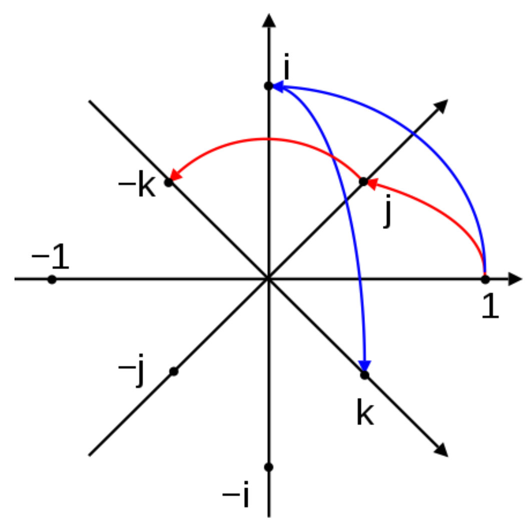
Quaternions can also be considered as a pair of **scalar** and **vector** quantities. It can be represented as,

$$q = s + \vec{v}$$

$$q = [s, \vec{v}]$$

$$q = a + bi + cj + dk$$

The axes can be visualized as shown in the diagram in the right.



---

2. Need for Quaternions

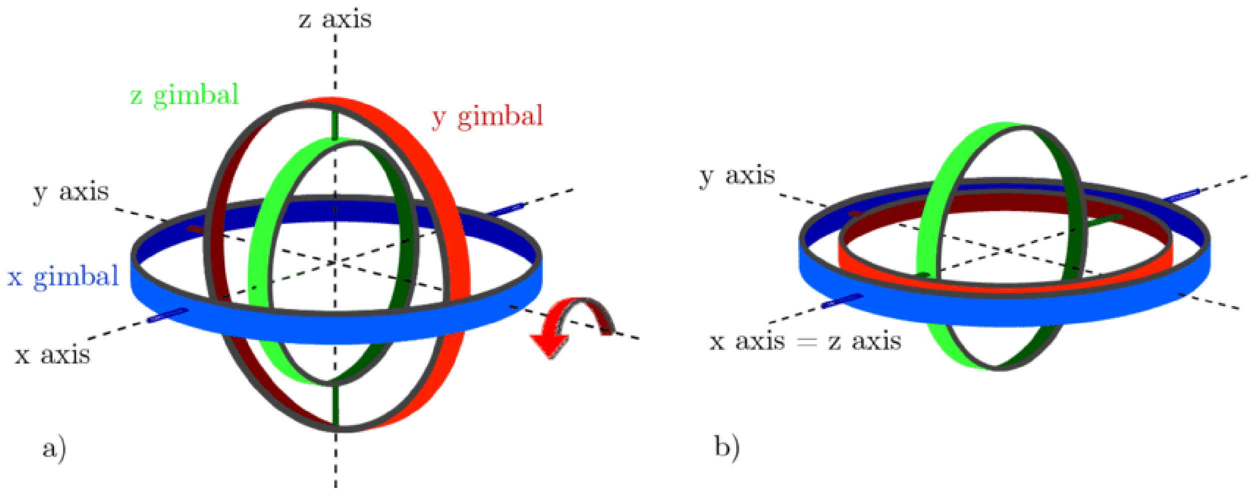
Over the last few years, quaternions are used **instead of matrices** to **rotate 3D characters**. We will go more into the details of the rotation in the application section.

The reason why Quaternions were preferred in computer graphics were as follows,

- Quaternions allow a character to rotate about multiple axes simultaneously, instead of sequentially.
- Quaternions consume less memory and computation time as compared to matrices.
- Quaternions can overcome the common issues with other methods of rotating 3D structures such as **Gimbal Lock**.

2.1 Gimbal Lock

A **Gimbal** is a hardware implementation of euler angles used for rotation in 3D space. **Gimbal lock** situation arises when **two rotational axes of an object point in the same direction.** Due to this orientation, the rotation ends up **losing one degree of freedom**, hence “locking” the system into r**otation in a degenerate 2D space**.



---

3. Mathematics related to Quaternions

We will only discuss a few basic operations like addition and subtraction, multiplication and norm. You can refer to [this link] to know more about mathematics pertaining to quaternions.

3.1 Addition and Subtraction

In quaternion addition and subtraction, the corresponding scalar and vectors from each quaternion are added/subtracted, respectively.

$$q_1 = [s_1, \vec{v}_1]$$
$$q_2 = [s_2, \vec{v}_2]$$
$$q_1 \pm q_2 = [s_1 \pm s_2, \vec{v}_1 \pm \vec{v}_2]$$

3.2 Multiplication

Quaternions can be multiplied by another quaternion or a scalar. Mathematically, the multiplication can be written as,

$$q_1 = [s_1, \vec{v}_1]$$

$$kq_1 = k[s_1, \vec{v}_1] = [ks_1, k\vec{v}_1]$$

$$q_2 = [s_2, \vec{v}_2]$$
$$q_1q_2 = [s_1, \vec{v}_1][s_2, \vec{v}_2]$$
$$q_1q_2 = [s_1s_2 - \vec{v}_1.\vec{v}_2 , s_1\vec{v}_2 + s_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2]$$

Note: The **Cross Product** of two quaternions is equivalent to the cross product of the vector part of quaternions.

$$q_1 = [s_1, \vec{v}_1]$$
$$q_2 = [s_2, \vec{v}_2]$$
$$q_1 \times q_2 = \vec{v}_1 \times \vec{v}_2$$

The **Dot Product** of two quaternions can be written as follows,

$$q_1 = [s_1, \vec{v}_1]$$
$$q_2 = [s_2, \vec{v}_2]$$
$$q_1 . q_2 = s_1s_2 + \vec{v}_1.\vec{v}_2$$

3.3 Norm

Norm for quaternions is similar to that of vectors, it can be represented as,

$$||q|| = (s^2 + |\vec{v}|^2)^{0.5}$$

---

4. Applications

Quaternions have various uses in the sector of science and technology, including but not limited to, **electro mechanics, quantum mechanics, 3D rotation and animation,** and **topology**. Here, we will only discuss 3D rotation and animation in brief.

4.1 Rotation using Quaternions

As mentioned above, quaternions allow a character to rotate about multiple axes simultaneously, instead of sequentially as matrix rotation allows.

In 3D space, according to **Euler's rotation theorem**, any rotation or sequence of rotations of a structure about a fixed point is equivalent to a **single rotation** by a **given angle *θ*** about a **fixed axis** (called the **Euler axis**) that runs through the fixed point.

Using this theorem, we can represent the rotation in 3D space as a combination of a vector $\vec{v}$ and a scalar ***θ.*** This **axis-angle representation** can be encoded as a quaternion,

$$q^{-1} = Cos(\theta/2) - (v_x i + v_y j + v_z k)\ Sin(\theta/2)$$
