Quaternions and Rotations*

(Com S 477/577 Notes)

Yan-Bin Jia

Sep 10, 2013

1 Introduction

The development of quaternions is attributed to W. R. Hamilton in 1843. Legend has it that Hamilton was walking with his wife Helen at the Royal Irish Academy when he was suddenly struck by the idea of adding a fourth dimension in order to multiply triples. Excited by this breakthrough, as the couple passed the Broome Bridge of the Royal Canal, he carved the newfound quaternion equations

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$$

into the stone of the bridge. This event is marked by a plaque at the exact location today. Hamilton spent the rest of his life working on quaternions, which became the first non-commutative algebra to be studied.

Up until now we have learned that a rotation in $\mathbb{R}^3$ about an axis through the origin can be represented by a 3×3 orthogonal matrix with determinant 1. However, the matrix representation seems redundant because only four of its nine elements are independent. Also the geometric interpretation of such a matrix is not clear until we carry out several steps of calculation to extract the rotation axis and angle. Furthermore, to compose two rotations, we need to compute the product of the two corresponding matrices, which requires twenty-seven multiplications and eighteen additions.

Quaternions are very efficient for analyzing situations where rotations in $\mathbb{R}^3$ are involved. A quaternion is a 4-tuple, which is a more concise representation than a rotation matrix. Its geometric meaning is also more obvious as the rotation axis and angle can be trivially recovered. The quaternion algebra to be introduced will also allow us to easily compose rotations. This is because quaternion composition takes merely sixteen multiplications and twelve additions.

2 Quaternion Algebra

The set of quaternions, together with the two operations of addition and multiplication, form a non-commutative ring.[1] The standard orthonormal basis for $\mathbb{R}^3$ is given by three unit vectors

*Sections 2–6 are based on Chapters 3–6 of the book [7] by J. B. Kuipers, and Sections 1 (partially) and 7 are based on the essay by S. Oldenburger [6] who took the course.

[1] For the purpose of this course, you don't really need to know what a ring is although it can be found in a standard algebra text such as the book by Hungerford [4] or Jacobson [5].

$\boldsymbol{i} = (1, 0, 0)$, $\boldsymbol{j} = (0, 1, 0)$, $\boldsymbol{k} = (0, 0, 1)$. A *quaternion* q is defined as the sum of a scalar q_0 and a vector $\boldsymbol{q} = (q_1, q_2, q_3)$; namely,

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

2.1 Addition and Multiplication

Addition of two quaternions acts componentwise. More specifically, consider the quaternion q above and another quaternion

$$p = p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k}.$$

Then we have

$$p + q = (p_0 + q_0) + (p_1 + q_1)\boldsymbol{i} + (p_2 + q_2)\boldsymbol{j} + (p_3 + q_3)\boldsymbol{k}.$$

Every quaternion q has a negative $-q$ with components $-q_i$, $i = 0, 1, 2, 3$.

The product of two quaternions satisfies these fundamental rules introduced by Hamilton:

$$\begin{aligned} \boldsymbol{i}^2 &= \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \\ \boldsymbol{ij} &= \boldsymbol{k} = -\boldsymbol{ji}, \\ \boldsymbol{jk} &= \boldsymbol{i} = -\boldsymbol{kj}, \\ \boldsymbol{ki} &= \boldsymbol{j} = -\boldsymbol{ik}. \end{aligned}$$

Now we can give the product of two quaternions p and q:

$$\begin{aligned} pq &= (p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k})(q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}) \\ &= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) + p_0(q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}) + q_0(p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k}) \\ &\quad + (p_2q_3 - p_3q_2)\boldsymbol{i} + (p_3q_1 - p_1q_3)\boldsymbol{j} + (p_1q_2 - p_2q_1)\boldsymbol{k}. \end{aligned}$$

Whew! It is too long to remember or even to understand what is going on. Fortunately, we can utilize the inner product and cross product of two vectors in $\mathbb{R}^3$ to write the above quaternion product in a more concise form:

$$pq = p_0q_0 - \boldsymbol{p}\cdot\boldsymbol{q} + p_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p}\times\boldsymbol{q}. \tag{1}$$

In the above, $\boldsymbol{p} = (p_1, p_2, p_3)$ and $\boldsymbol{q} = (q_1, q_2, q_3)$ are the vector parts of p and q, respectively.

Example 1. Suppose the two vectors are given as follows:

$$\begin{aligned} p &= 3 + \boldsymbol{i} - 2\boldsymbol{j} + \boldsymbol{k}, \\ q &= 2 - \boldsymbol{i} + 2\boldsymbol{j} + 3\boldsymbol{k}. \end{aligned}$$

We single out their vector parts $\boldsymbol{p} = (1, -2, 1)$ and $\boldsymbol{q} = (-1, 2, 3)$ and calculate their inner and cross products:

$$\begin{aligned} \boldsymbol{p}\cdot\boldsymbol{q} &= -2, \\ \boldsymbol{p}\times\boldsymbol{q} &= \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 1 & -2 & 1 \\ -1 & 2 & 3 \end{vmatrix} \\ &= -8\boldsymbol{i} - 4\boldsymbol{j}. \end{aligned}$$

By (1) the quaternion product is then

$$\begin{aligned} pq &= 6 - (-2) + 3(-\boldsymbol{i} + 2\boldsymbol{j} + 3\boldsymbol{k}) + 2(\boldsymbol{i} - 2\boldsymbol{j} + \boldsymbol{k}) + (-8\boldsymbol{i} - 4\boldsymbol{j}) \\ &= 8 - 9\boldsymbol{i} - 2\boldsymbol{j} + 11\boldsymbol{k}. \end{aligned}$$

We see that the product of two quaternions is still a quaternion with scalar part $p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q}$ and vector part $p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q}$. The set of quaternions is closed under multiplication and addition. It is not difficult to verify that multiplication of quaternions is distributive over addition. The identity quaternion has real part 1 and vector part $\boldsymbol{0}$.

2.2 Complex Conjugate, Norm, and Inverse

Let $q = q_0 + \boldsymbol{q} = q_0 + q_1 \boldsymbol{i} + q_2 \boldsymbol{j} + q_3 \boldsymbol{k}$ be a quaternion. The *complex conjugate* of q, denoted q^*, is defined as

$$q^* = q_0 - \boldsymbol{q} = q_0 - q_1 \boldsymbol{i} - q_2 \boldsymbol{j} - q_3 \boldsymbol{k}.$$

From the definition we immediately have

$$\begin{aligned} (q^*)^* &= q_0 - (-\boldsymbol{q}) \quad = \quad q, \\ q + q^* &= 2q_0, \\ q^* q &= (q_0 - \boldsymbol{q})(q_0 + \boldsymbol{q}) \\ &= q_0 q_0 - (-\boldsymbol{q}) \cdot \boldsymbol{q} + q_0 \boldsymbol{q} + (-\boldsymbol{q}) q_0 + (-\boldsymbol{q}) \times \boldsymbol{q} \qquad \text{by (1)} \\ &= q_0^2 + \boldsymbol{q} \cdot \boldsymbol{q} \\ &= q_0^2 + q_1^2 + q_2^2 + q_3^2 \\ &= q q^*. \end{aligned}$$

Given two quaternions p and q, we can easily verify that

$$(pq)^* = q^* p^*. \tag{2}$$

The *norm* of a quaternion q, denoted by $|q|$, is the scalar $|q| = \sqrt{q^* q}$. A quaternion is called a *unit quaternion* if its norm is 1. The norm of the product of two quaternions p and q is the product of the individual norms, for we have

$$\begin{aligned} |pq|^2 &= (pq)(pq)^* \\ &= pqq^*p^* \\ &= p|q|^2 p^* \\ &= pp^*|q|^2 \\ &= |p|^2|q|^2. \end{aligned}$$

The *inverse* of a quaternion q is defined as

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

We can easily verify that $q^{-1} q = q q^{-1} = 1$. In the case q is a unit quaternion, the inverse is its conjugate q^*.

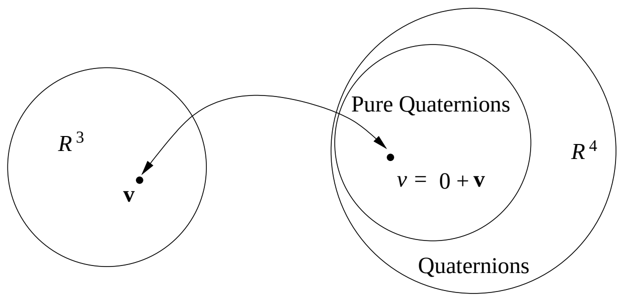
3 Quaternion Rotation Operator

How can a quaternion, which lives in $\mathbb{R}^4$, operate on a vector, which lives in $\mathbb{R}^3$? First, we note that a vector $\boldsymbol{v} \in \mathbb{R}^3$ is a *pure quaternion* whose real part is zero. Let us consider a unit quaternion $q = q_0 + \boldsymbol{q}$ only. That $q_0^2 + \|\boldsymbol{q}\|^2 = 1$ implies that there must exist some angle θ such that

$$\begin{aligned} \cos^2\theta &= q_0^2, \\ \sin^2\theta &= \|\boldsymbol{q}\|^2. \end{aligned}$$

In fact, there exists a unique $\theta \in [0, \pi]$ such that $\cos\theta = q_0$ and $\sin\theta = \|\boldsymbol{q}\|$. The unit quaternion can now be written in terms of the angle θ and the unit vector $\boldsymbol{u} = \boldsymbol{q}/\|\boldsymbol{q}\|$:

$$q = \cos\theta + \boldsymbol{u}\sin\theta.$$



Using the unit quaternion q we define an operator on vectors $\boldsymbol{v} \in \mathbb{R}^3$:

$$\begin{aligned} L_q(\boldsymbol{v}) &= q\boldsymbol{v}q^* \\ &= (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{v} + 2(\boldsymbol{q}\cdot\boldsymbol{v})\boldsymbol{q} + 2q_0(\boldsymbol{q}\times\boldsymbol{v}). \end{aligned} \tag{3}$$

Here we make two observations. First, the quaternion operator (3) does not change the length of the vector $\boldsymbol{v}$ for

$$\begin{aligned} \|L_q(\boldsymbol{v})\| &= \|q\boldsymbol{v}q^*\| \\ &= |q| \cdot \|\boldsymbol{v}\| \cdot |q^*| \\ &= \|\boldsymbol{v}\|. \end{aligned}$$

Second, the direction of $\boldsymbol{v}$, if along $\boldsymbol{q}$, is left unchanged by the operator L_q. To verify this, we let $\boldsymbol{v} = k\boldsymbol{q}$ and have

$$\begin{aligned} q\boldsymbol{v}q^* &= q(k\boldsymbol{q})q^* \\ &= (q_0^2 - \|\boldsymbol{q}\|^2)(k\boldsymbol{q}) + 2(\boldsymbol{q}\cdot k\boldsymbol{q})\boldsymbol{q} + 2q_0(\boldsymbol{q}\times k\boldsymbol{q}) \\ &= k(q_0^2 + \|\boldsymbol{q}\|^2)\boldsymbol{q} \\ &= k\boldsymbol{q}. \end{aligned}$$

The two observations make us guess that the operator L_q acts like a rotation about $\boldsymbol{q}$. This is made precise by the next theorem.

Before proceeding with the theorem, we remark that the operator L_q is linear over $\mathbb{R}^3$. For any two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^3$ and any $a_1, a_2 \in R$ we can show that

$$L_q(a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2) = a_1L_q(\boldsymbol{v}_1) + a_2L_q(\boldsymbol{v}_2).$$

Theorem 1 *For any unit quaternion*

$$q = q_0 + \boldsymbol{q} = \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2}, \tag{4}$$

and for any vector $\boldsymbol{v} \in \mathbb{R}^3$ *the action of the operator*

$$L_q(\boldsymbol{v}) = q\boldsymbol{v}q^*$$

on $\boldsymbol{v}$ *is equivalent to a rotation of the vector through an angle* θ *about* $\boldsymbol{u}$ *as the axis of rotation.*

Proof Given a vector $\boldsymbol{v} \in \mathbb{R}^3$, we decompose it as $\boldsymbol{v} = \boldsymbol{a} + \boldsymbol{n}$, where $\boldsymbol{a}$ is the component along the vector $\boldsymbol{q}$ and $\boldsymbol{n}$ is the component normal to $\boldsymbol{q}$. Then we show that under the operator L_q, $\boldsymbol{a}$ is invariant, while $\boldsymbol{n}$ is rotated about $\boldsymbol{q}$ through an angle θ. Since the operator is linear, this shows that the image $q\boldsymbol{v}q^*$ is indeed interpreted as a rotation of $\boldsymbol{v}$ about $\boldsymbol{q}$ through an angle θ.

We know from an early reasoning that $\boldsymbol{a}$ is invariant under L_q. So let us focus on the effect of L_q on the orthogonal component $\boldsymbol{n}$. We have

$$\begin{aligned} L_q(\boldsymbol{n}) &= (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{n} + 2(\boldsymbol{q}\cdot\boldsymbol{n})\boldsymbol{q} + 2q_0(\boldsymbol{q}\times\boldsymbol{n}) \\ &= (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{n} + 2q_0(\boldsymbol{q}\times\boldsymbol{n}) \\ &= (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{n} + 2q_0\|\boldsymbol{q}\|(\boldsymbol{u}\times\boldsymbol{n}), \end{aligned}$$

where in the last step above we introduced $\boldsymbol{u} = \boldsymbol{q}/\|\boldsymbol{q}\|$. Denote $\boldsymbol{n}_\perp = \boldsymbol{u}\times\boldsymbol{n}$. So the last equation becomes

$$L_q(\boldsymbol{n}) = (q_0^2 - \|\boldsymbol{q}\|^2)\boldsymbol{n} + 2q_0\|\boldsymbol{q}\|\boldsymbol{n}_\perp. \tag{5}$$

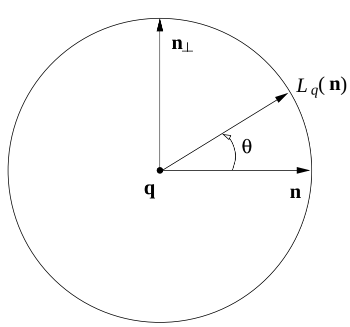
Note that $\boldsymbol{n}_\perp$ and $\boldsymbol{n}$ have the same length:

$$\|\boldsymbol{n}_\perp\| = \|\boldsymbol{n}\times\boldsymbol{u}\| = \|\boldsymbol{n}\|\cdot\|\boldsymbol{u}\|\sin\frac{\pi}{2} = \|\boldsymbol{n}\|.$$

Finally, we rewrite (5) into the form

$$\begin{aligned} L_q(\boldsymbol{n}) &= \left(\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}\right)\boldsymbol{n} + \left(2\cos\frac{\theta}{2}\sin\frac{\theta}{2}\right)\boldsymbol{n}_\perp \\ &= \cos\theta\boldsymbol{n} + \sin\theta\boldsymbol{n}_\perp. \end{aligned}$$

Namely, the resulting vector is a rotation of $\boldsymbol{n}$ through an angle θ in the plane defined by $\boldsymbol{n}$ and $\boldsymbol{n}_\perp$. See the figure below. This vector is clearly orthogonal to the rotation axis.



□

We substitute the unit quaternion form (4) into (3) to obtain the resulting vector from rotating a vector $\boldsymbol{v}$ about the axis $\boldsymbol{u}$ through θ:

$$\begin{aligned} L_p(\boldsymbol{v}) &= \left(\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}\right)\boldsymbol{v} + 2\left(\boldsymbol{u}\sin\frac{\theta}{2}\cdot\boldsymbol{v}\right)\boldsymbol{u}\sin\frac{\theta}{2} + 2\cos\frac{\theta}{2}\left(\boldsymbol{u}\sin\frac{\theta}{2}\times\boldsymbol{v}\right) \\ &= \cos\theta\cdot\boldsymbol{v} + (1-\cos\theta)(\boldsymbol{u}\cdot\boldsymbol{v})\boldsymbol{u} + \sin\theta\cdot(\boldsymbol{u}\times\boldsymbol{v}). \end{aligned} \tag{6}$$

Example 2. Consider a rotation about an axis defined by $(1,1,1)$ through an angle of $2\pi/3$. About this axis, the basis vectors $\boldsymbol{i}, \boldsymbol{j}$, and $\boldsymbol{k}$ generate the same cone when rotated through 2π. We define a unit vector

$$\boldsymbol{u} = \frac{1}{\sqrt{3}}(1,1,1).$$

Let the rotation angle $\theta = 2\pi/3$. The quaternion q defines the rotation is then given as

$$\begin{aligned} q &= \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2} \\ &= \frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{k}. \end{aligned}$$

Let us compute the effect of rotation on the basis vector $\boldsymbol{i} = (1,0,0)$. We obtain the resulting vector using (6):

$$\begin{aligned} \boldsymbol{v} &= -\frac{1}{2}\begin{pmatrix}1\\0\\0\end{pmatrix} + \left(1+\frac{1}{2}\right)\cdot\frac{1}{\sqrt{3}}\cdot\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} + \frac{\sqrt{3}}{2}\cdot\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}\times\begin{pmatrix}1\\0\\0\end{pmatrix} \\ &= \begin{pmatrix}-\frac{1}{2}\\0\\0\end{pmatrix} + \begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\\\frac{1}{2}\end{pmatrix} + \begin{pmatrix}0\\\frac{1}{2}\\-\frac{1}{2}\end{pmatrix} \\ &= \boldsymbol{j}. \end{aligned}$$

The rotation of $\boldsymbol{v}$ under the operator L_q can also be interpreted from the perspective of an observer attached to the vector $\boldsymbol{v}$. What he sees happening is that the coordinate frame rotates through the angle $-\theta$ about the same axis defined by the quaternion.

Theorem 2 *For any unit quaternion*

$$q = q_0 + \boldsymbol{q} = \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2},$$

and for any vector $\boldsymbol{v} \in \mathbb{R}^3$ the action of the operator

$$L_{q^*}(\boldsymbol{v}) = q^*\boldsymbol{v}(q^*)^* = q^*\boldsymbol{v}q$$

is a rotation of the coordinate frame about the axis $\boldsymbol{u}$ through an angle θ while $\boldsymbol{v}$ is not rotated.

Equivalently, the operator L_{q^*} rotates the vector $\boldsymbol{v}$ with respect to the coordinate frame through an angle $-\theta$ about $\boldsymbol{q}$.

The quaternion operator $L_q(\boldsymbol{v}) = q\boldsymbol{v}q^*$ may be interpreted as a *point or vector rotation* with respect to the (fixed) coordinate frame. The quaternion operator $L_{q^*}(\boldsymbol{v}) = q^*\boldsymbol{v}q$ may be interpreted as a *coordinate frame rotation* with respect to the (fixed) space of points.

4 Quaternion Operator Sequences

Let p and q be two unit quaternions. We first apply the operator L_p to the vector $\boldsymbol{u}$ and obtain the vector $\boldsymbol{v}$. To $\boldsymbol{v}$ we then apply the operator L_q and obtain the vector $\boldsymbol{w}$. Equivalently, we apply the composition $L_q \circ L_p$ of the two operators:

$$\begin{aligned}
\boldsymbol{w} &= L_q(\boldsymbol{v}) \\
&= q\boldsymbol{v}q^* \\
&= q(p\boldsymbol{u}p^*)q^* \\
&= (qp)\boldsymbol{u}(qp)^* \\
&= L_{qp}(\boldsymbol{u}).
\end{aligned}$$

Because p and q are unit quaternions, so is the product qp. Hence the above equation describes a rotation operator whose defining quaternion is the product of the two quaternions p and q. The axis and angle of the composite rotation is given by the product qp.

Similarly, consider the quaternion operators $L_{p^*}(\boldsymbol{u}) = p^*\boldsymbol{u}p$ and $L_{q^*}(\boldsymbol{v}) = q^*\boldsymbol{v}q$ which carry out rotations of the coordinate system determined by quaternions p and q, respectively. Then the quaternion product pq defines an operator $L_{(pq)^*}$, which represents a sequence of operators L_{p^*} followed by L_{q^*}. The axis and angle of rotation of $L_{(pq)^*}$ are those represented by the quaternion product pq.

Example 3. We now use the quaternion method to find the axis and angle of the composite rotation in the Satellite tracking example from the notes titled "Space Rotations". Recall that the tracking application takes a rotation about the z-axis through a bearing angle α followed by a rotation about the new y-axis through an elevation angle β. After these two rotations, the new x-axis points toward the satellite. The two rotations are respectively described by the two quaternions below:

$$\begin{aligned}
p &= \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}\boldsymbol{k}, \\
q &= \cos\frac{\beta}{2} + \sin\frac{\beta}{2}\boldsymbol{j}.
\end{aligned}$$

Since we are rotating the coordinate frame, the two operators L_{p^*} and L_{q^*} are applied sequentially. The composite rotation operator is $L_{(pq)^*}$, which transforms coordinates in the station frame to those in the tracking frame. And the quaternion describing the composition rotation is the product pq which is as follows.

$$\begin{aligned}
pq &= \left(\cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}\boldsymbol{k}\right)\left(\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\boldsymbol{j}\right) \\
&= \cos\frac{\alpha}{2}\cos\frac{\beta}{2} + \cos\frac{\alpha}{2}\sin\frac{\beta}{2}\boldsymbol{j} + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\boldsymbol{k} + \sin\frac{\alpha}{2}\sin\frac{\beta}{2}(\boldsymbol{k}\times\boldsymbol{j}) \\
&= \cos\frac{\alpha}{2}\cos\frac{\beta}{2} - \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\boldsymbol{i} + \cos\frac{\alpha}{2}\sin\frac{\beta}{2}\boldsymbol{j} + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\boldsymbol{k}.
\end{aligned}$$

The axis of the composite rotation is defined by the vector

$$\boldsymbol{v} = \left(-\sin\frac{\alpha}{2}\sin\frac{\beta}{2},\ \cos\frac{\alpha}{2}\sin\frac{\beta}{2},\ \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\right). \tag{7}$$

And the angle of rotation θ satisfies

$$\begin{aligned}
\cos\frac{\theta}{2} &= \cos\frac{\alpha}{2}\cos\frac{\beta}{2}, \\
\sin\frac{\theta}{2} &= \|\boldsymbol{v}\|.
\end{aligned}$$

The cosine is same as obtained in Section 3 of the handouts titled "Rotation in the Space" for we have

$$
\begin{aligned}
\cos\theta &= 2\cos^2\frac{\theta}{2} - 1 \\
&= 2\cos^2\frac{\alpha}{2}\cos^2\frac{\beta}{2} - 1 \\
&= 2\frac{\cos\alpha + 1}{2}\cdot\frac{\cos\beta + 1}{2} - 1 \\
&= \frac{\cos\alpha\cos\beta + \cos\alpha + \cos\beta - 1}{2}.
\end{aligned}
$$

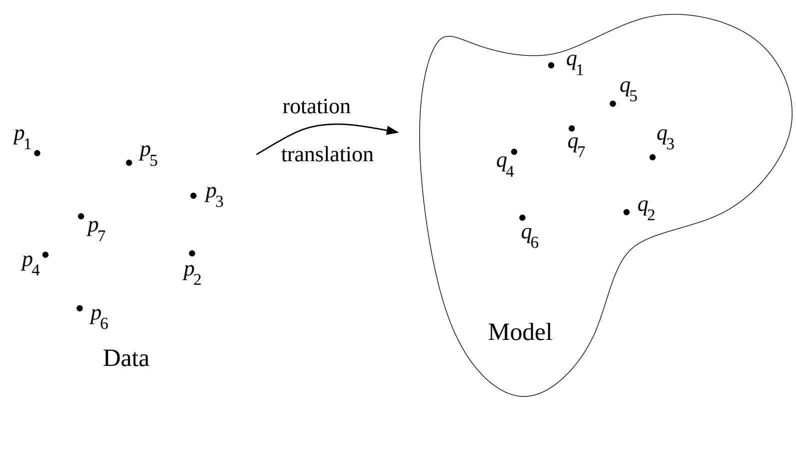
Note that the rotation axis and angle in that section transforms coordinates in the tracking frame to those in the station frame. This explains why the axis $\boldsymbol{v}$ in (7) is opposite to the one obtained in that section while the angle is the same.

5 Application: 3-D Shape Registration

An important problem in model-based recognition is to find the transformation of a set of data points that yields the best match of these points against a shape model. The process is often referred to as *data registration*. The data points are typically measured on a real object by range sensors, touch sensors, etc., and given in Cartesian coordinates. The quality of a match is often described as the total squared distance from the data points to the model. When multiple shape models are possible, the one that results in the least total distance is then recognized as the shape of the object.

Quaternions are very effective in solving the above least-squares-based registration problem. Let us begin with a formulation of the problem in 3D. Let $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ be a set of data points. We assume that $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ are to be matched against the points $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ on a shape model. Namely, the *correspondences* between the data points and those on the model have been predetermined. Then the problem is to find a rotation, represented by an orthogonal matrix R with $\det(R) = 1$, and a translation $\boldsymbol{b}$ as the solution to the following minimization:

$$\min_{R,\boldsymbol{b}} \sum_{i=1}^{n} \|R\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_i\|^2. \tag{8}$$



We begin by computing the centroids of the two sets of points:

$$
\begin{aligned}
\bar{\boldsymbol{p}} &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{p}_i; \\
\bar{\boldsymbol{q}} &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{q}_i.
\end{aligned}
$$

The relative coordinates of all the points to their centroids are obtained as, for $1 \le i \le n$,

$$
\begin{aligned}
\boldsymbol{p}_i' &= \boldsymbol{p}_i - \bar{\boldsymbol{p}}; \\
\boldsymbol{q}_i' &= \boldsymbol{q}_i - \bar{\boldsymbol{q}}.
\end{aligned}
$$

Clearly, we have

$$
\sum_{i=1}^{n}\boldsymbol{p}_i' = \sum_{i=1}^{n}\boldsymbol{p}_i - n\bar{\boldsymbol{p}} = \sum_{i=1}^{n}\boldsymbol{p}_i - n\cdot\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{p}_i = 0; \tag{9}
$$

$$
\sum_{i=1}^{n}\boldsymbol{q}_i' = \sum_{i=1}^{n}\boldsymbol{q}_i - n\bar{\boldsymbol{q}} = \sum_{i=1}^{n}\boldsymbol{q}_i - n\cdot\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{q}_i = 0. \tag{10}
$$

Let us rewrite the objective function in (8) in terms of $\bar{p}, \bar{q}, \boldsymbol{p}_i', \boldsymbol{q}_i'$:

$$
\begin{aligned}
\sum_{i=1}^{n}\|R\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_i\|^2 &= \sum_{i=1}^{n}\|R\boldsymbol{p}_i' - \boldsymbol{q}_i' + R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}\|^2 \\
&= \sum_{i=1}^{n}(R\boldsymbol{p}_i' - \boldsymbol{q}_i' + R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b})\cdot(R\boldsymbol{p}_i' - \boldsymbol{q}_i' + R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}) \\
&= \sum_{i=1}^{n}\|R\boldsymbol{p}_i' - \boldsymbol{q}_i'\|^2 + \left(2\sum_{i=1}^{n}(R\boldsymbol{p}_i' - \boldsymbol{q}_i')\right)\cdot(R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}) + n\|R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}\|^2 \\
&= \sum_{i=1}^{n}\|R\boldsymbol{p}_i' - \boldsymbol{q}_i'\|^2 + 2\left(R\sum_{i=1}^{n}\boldsymbol{p}_i' - \sum_{i=1}^{n}\boldsymbol{q}_i'\right)\cdot(R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}) + n\|R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}\|^2 \\
&= \sum_{i=1}^{n}\|R\boldsymbol{p}_i' - \boldsymbol{q}_i'\|^2 + n\|R\bar{\boldsymbol{p}} - \bar{\boldsymbol{q}} + \boldsymbol{b}\|^2, \qquad \text{by (9) and (10).}
\end{aligned}
$$

The minimizing translation $\boldsymbol{b}$ should make the second term in the last equation above zero, yielding:

$$
\boldsymbol{b} = \bar{\boldsymbol{q}} - R\bar{\boldsymbol{p}}. \tag{11}
$$

Thus we have decomposed the problem of data registration into two phases: the first of which determines its optimal translation, as given by equation (11), and the second of which determines the optimal rotation of the set $\{\boldsymbol{p}_i\}$. Note that every point $\boldsymbol{p}_i$ is transformed into $R(\boldsymbol{p}_i - \bar{\boldsymbol{p}}) + \bar{\boldsymbol{q}}$ before matching against $\boldsymbol{q}_i$. Equivalently, to find the best match of the two point sets $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{q}_i\}$, we first translate $\{\boldsymbol{p}_i\}$ to let their centroid coincide with that of $\{\boldsymbol{q}_i\}$, and then rotate about the common centroid.

By the reasoning so far, the optimal rotation can be solved from the formulation below:

$$\min_R \sum_{i=1}^{n} \|R\boldsymbol{p}'_i - \boldsymbol{q}'_i\|^2. \tag{12}$$

Here we present an exact solution to (12) as described in [3] using quaternions. An equivalent quaternion-based solution is given in [2]. The version of matching two curves (or surfaces), also assuming pointwise correspondences, is solved exactly in [8] in a somewhat similar manner without the use of quaternions.

First, we rewrite the summation in (12) as follows:

$$\begin{aligned}\sum_{i=1}^{n} \|R\boldsymbol{p}'_i - \boldsymbol{q}'_i\|^2 &= \sum_{i=1}^{n}(R\boldsymbol{p}'_i \cdot R\boldsymbol{p}'_i) - 2\sum_{i=1}^{n}(R\boldsymbol{p}'_i \cdot \boldsymbol{q}'_i) + \sum_{i=1}^{n} \boldsymbol{q}'_i \cdot \boldsymbol{q}'_i \\ &= \sum_{i=1}^{n} \left(\|\boldsymbol{p}'_i\|^2 + \|\boldsymbol{q}'_i\|^2\right) - 2\sum_{i=1}^{n} R\boldsymbol{p}'_i \cdot \boldsymbol{q}'_i.\end{aligned}$$

The first summand in the last equation above does not depend on the rotation, so we need only minimize the second summand. Equivalently, this can be done through a maximization:

$$\max_R \sum_{i=1}^{n} R\boldsymbol{p}'_i \cdot \boldsymbol{q}'_i. \tag{13}$$

The rotation matrix R has nine entries, only four of which are independent due to the orthogonality and unit determinant of R. Instead, we represent rotations using unit quaternions. Essentially, we find the unit quaternion q that maximizes

$$\sum_{i=1}^{n}(q\boldsymbol{p}'_i q^*) \cdot \boldsymbol{q}'_i. \tag{14}$$

Here we view quaternions as vectors in $\mathbb{R}^4$. Let $q = (q_0, q_1, q_2, q_3)^T$ and $q^* = (q_0, -q_1, -q_2, -q_3)^T$. Also, the points $\boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_n$ and $\boldsymbol{q}'_1, \ldots, \boldsymbol{q}'_n$ are viewed as 4-tuples with $\boldsymbol{p}'_i = (0, p'_{i1}, p'_{i2}, p'_{i3})^T$ and $\boldsymbol{q}'_i = (0, q'_{i1}, q'_{i2}, q'_{i3})^T$ by a slight abuse of notation.

Applying the definition of quaternion product, it is not difficult to show that

$$(q\boldsymbol{p}'_i q^*) \cdot \boldsymbol{q}'_i = (q\boldsymbol{p}'_i) \cdot (\boldsymbol{q}'_i q). \tag{15}$$

Next, we intend to rewrite the summands in (14) as matrix products. For this purpose, we define matrices

$$P_i = \begin{pmatrix} 0 & -p'_{i1} & -p'_{i2} & -p'_{i3} \\ p'_{i1} & 0 & p'_{i3} & -p'_{i2} \\ p'_{i2} & -p'_{i3} & 0 & p'_{i1} \\ p'_{i3} & p'_{i2} & -p'_{i1} & 0 \end{pmatrix} \quad \text{and} \quad Q_i = \begin{pmatrix} 0 & -q'_{i1} & -q'_{i2} & -q'_{i3} \\ q'_{i1} & 0 & -q'_{i3} & q'_{i2} \\ q'_{i2} & q'_{i3} & 0 & -q'_{i1} \\ q'_{i3} & -q'_{i2} & q'_{i1} & 0 \end{pmatrix},$$

for $1 \leq i \leq n$. Then the quaternion products $q\boldsymbol{p}'_i$ and $\boldsymbol{q}'_i q$ are equivalent to the matrix products $P_i q$ and $Q_i q$. We thus have

$$\sum_{i=1}^{n}(q\boldsymbol{p}'_i q^*) \cdot \boldsymbol{q}'_i = \sum_{i=1}^{n}(q\boldsymbol{p}'_i) \cdot (\boldsymbol{q}'_i q) \qquad \text{from (15)}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} (P_i q) \cdot (Q_i q) \\
&= \sum_{i=1}^{n} q^T P_i^T Q_i q \\
&= q^T \left(\sum_{i=1}^{n} P_i^T Q_i \right) q.
\end{aligned}
$$

It is easy to verify that each matrix $P_i^T Q_i$ is symmetric, so is the 4×4 matrix

$$
M = \sum_{i=1}^{n} P_i^T Q_i.
$$

Thus M has real eigenvalues only, say, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$.[2] Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$ be the corresponding orthogonal unit eigenvectors. Eigenvectors corresponding to different eigenvalues must be orthogonal to each other. Multiple eigenvectors corresponding to the same eigenvalue are chosen to be orthogonal to each other. The quaternion q is a linear combination of these eigenvectors:

$$
q = \alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \alpha_3 \boldsymbol{v}_3 + \alpha_4 \boldsymbol{v}_4.
$$

Therefore we have

$$
\begin{aligned}
q^T M q &= (\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \alpha_3 \boldsymbol{v}_3 + \alpha_4 \boldsymbol{v}_4)^T M (\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \alpha_3 \boldsymbol{v}_3 + \alpha_4 \boldsymbol{v}_4) \\
&= (\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \alpha_3 \boldsymbol{v}_3 + \alpha_4 \boldsymbol{v}_4) \cdot (\lambda_1 \alpha_1 \boldsymbol{v}_1 + \lambda_2 \alpha_2 \boldsymbol{v}_2 + \lambda_3 \alpha_3 \boldsymbol{v}_3 + \lambda_4 \alpha_4 \boldsymbol{v}_4) \\
&= \lambda_1 \alpha_1^2 + \lambda_2 \alpha_2^2 + \lambda_3 \alpha_3^2 + \lambda_4 \alpha_4^2.
\end{aligned}
$$

The product $q^T M q$ achieves its maximum when $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = \alpha_4 = 0$. Therefore, the unit quaternion q that maximizes (14) is *the eigenvector that corresponds to the largest eigenvalue of the matrix M*. It describes the optimal rotation for (12), i.e, for data registration.

When the corresponding points $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ are unknown, a well-known method called the Iterative Closest Point (ICP) [1] solves the registration problem. Given a set of data points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$, the ICP algorithm finds the initial corresponding points $\boldsymbol{q}_1^{(0)}, \ldots, \boldsymbol{q}_n^{(0)}$ as the closest points on the surface model to $\boldsymbol{p}_1^{(0)} = \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n^{(0)} = \boldsymbol{p}_n$, respectively. Then it applies the introduced quaternion-based method to determine the rotation and translation that best match $\{\boldsymbol{p}_i^{(0)}\}$ with $\{\boldsymbol{q}_i^{(0)}\}$. The second iteration applies the just found transformation to every $\boldsymbol{p}_i^{(0)}$, obtaining $\boldsymbol{p}_i^{(1)}$, and then determines its new corresponding point $\boldsymbol{q}_i^{(1)}$ on the model as the closest point to $\boldsymbol{p}_i^{(1)}$. Recompute the best rotation and translation using quaternions, and so on. The algorithm stops when the change in the new transformation becomes small enough.

6 Other Applications of Quaternions

In physics, quaternions are correlated to the nature of the universe at the level of quantum mechanics. They lead to elegant expressions of the Lorentz transformations, which form the basis of

[2]Multiplicities of the eigenvalues are counted.

the modern theory of relativity. In signal processing, Quaternion Fourier Transform (QFT) is a powerful tool. The QFT restores the lost commutative property at the cost of no longer being a division algebra. It can be used, for instance, to embed a watermark in a color image. Other applications of QFT include face recognition (jointly with Quaternion Wavelet Transform) and voice recognition [6].

7 Quaternions vs Homogeneous Coordinates

Homogeneous coordinates are introduced to make translation multiplicative, along with scaling and rotation. They are convenient in representing points, lines, and planes, and fundamental for studying projections. Like quaternions, homogeneous coordinates are 4-tuples. This suggests that there might be a way of doing scaling and translation using some sort of quaternion operator. As of now, no such way has been found as quaternions and their rotation operators are algebraically incompatible with homogeneous coordinates.

References

[1] P. J. Besl and N. D. McKay. *A method for registration of 3-D shapes. IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[2] O. D. Faugeras and M. Hebert. *The representation, recognition, and locating of 3-D objects. International Journal of Robotics Research*, 5(3):27–52, 1986.

[3] B. K. P. Horn. *Closed-form solution of absolute orientation using unit quaternions. Journal of Optical Society of America A*, 4(4):629–642, 1987.

[4] T. W. Hungerford. *Algebra*. Springer-Verlag, 1974.

[5] N. Jacobson. *Basic Algebra*. W. H. Freeman & Co.,1985.

[6] S. Oldenburger. *Applications of Quaternions*. Written project of the course "Problem Solving Techniques in Applied Computer Science" (Com S 477/577), Department of Computer Science, Iowa State University, 2005.

[7] J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1999.

[8] J. T. Schwartz and M. Sharir. *Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. International Journal of Robotics Research*, 6(2):29–44, 1987.