3D Rotations

Gabriel Taubin
Brown University

3D rotations are used everywhere in computer graphics, computer vision, geometric modeling and processing, and other related areas. However, manipulating them is always confusing, and debugging code that involves 3D rotation is usually time-consuming. Many different, and seemingly unrelated, ways exist to describe 3D rotations—such as orthogonal matrices, turnaround vectors, exponential maps, and quaternions.

Here, I use elementary concepts from algebra, analytic geometry, and calculus to reveal how these and other, less well-known representations are related. (For a definition of some of the concepts I cover here, see the related sidebar.) My main goal is to produce simple, elegant, and efficient code, while paying close attention to the computational cost and, in particular, trying to avoid computing transcendental functions and square roots.

The Rotation Matrix

Let's start with the algebraic representation. We can represent a 3D rotation as an orthogonal 3×3 matrix Q. That is, Q has a transpose equal to its inverse $QQ^t = I$, where I is the identity matrix, and has the unit determinant $|Q| = 1$.

We obtain the result of applying a rotation to a 3D vector v by multiplying the matrix by the vector Qv, which requires nine multiplications and six additions. The set of all such matrices forms an algebraic group with respect to matrix multiplication called the *special orthogonal group*, denoted as $SO(3)$.

Rotation around a Vector

The geometric representation is a more intuitive way of describing a 3D rotation as a turn of angle θ around a unit-length 3D vector u, with the rotation's positive direction specified by the right-hand rule. We denote this rotation as $Q(\theta, u)$.

Because the geometric representation has a more intuitive meaning, it's used for modifying 3D rotations—for example, changing the viewpoint in a 3D viewer or modeling system. However, for applying the same rotation to multiple vectors, the algebraic representation is easier and usually less computationally expensive.

We must first determine the relation between the algebraic representation and this one. Can we go back and forth between the two? More specifically, can we describe every rotation matrix Q as a rotation of a certain angle around a unit vector? If so, how do we determine the vector and the angle? And how do we determine whether the description is unique? Conversely, given a unit vector and an angle, what is the 3D rotation's corresponding algebraic representation?

We start by determining whether the geometric representation is unique: it isn't. The rotation of θ around u is indistinguishable from the rotation of an angle $\theta + 2k\pi$ around the same vector $Q(\theta + 2k\pi, u) = Q(\theta, u)$; this is true for every integer k. In particular, the rotation of angle 2π (360 degrees) around any vector is identical to the identity. In other words, applying such a rotation is equivalent to not doing anything.

Also, the rotation of angle $-\theta$ around the vector $-u$ is identical to the rotation of θ around u. That is, $Q(-\theta, -u) = Q(\theta, u)$.

The Rodrigues Formula

What about computing a 3D rotation's algebraic representation from the geometric representation? The matrix representation of $Q(\theta, u)$ is given by the Rodrigues formula:

$$Q(\theta, u) = I + sU + (1 - c)U^2,$$

where U is the skew-symmetric ($U^t = -U$) matrix

$$U = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}$$

 Published by the IEEE Computer Society 0272-1716/11/$26.00

corresponding to the vector product by u. That is, for every vector v, we have

$$Uv = u \times v = \begin{pmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{pmatrix}.$$

I leave it to you to verify by direct computation that $U^2 = uu^t - I$.

Now, where does the Rodrigues formula come from? We can explain it using plain and simple analytic geometry. Let $Q = Q(\theta, u)$ be the algebraic representation of the 3D rotation that we're looking for, and let v be an arbitrary 3D vector. We want to find an expression for the 3D vector Qv as a function of θ, u, and, of course, v. We start by decomposing v into the sum of two orthogonal 3D vectors

$$v = (uu^t)v + (I - uu^t)v.$$

The first vector, $v_0 = (uu^t)v = \lambda u$, is the orthogonal projection of v onto the line spanned by u. This is u scaled by the inner product $\lambda = u^t v$ of u and v. The second vector is the difference $v_1 = v - v_0$, which is a 3D vector orthogonal to both u and v_0.

Since multiplication by a matrix is a linear operation, we have $Qv = Qv_0 + Qv_1$, and the problem reduces to finding expressions for Qv_0 and Qv_1 as functions of θ, u, and v. Because the rotation is around u, and v_0 and u are colinear, we have $Qv_0 = v_0$. For the other term, let $v_2 = u \times v_1 = u \times v$. Because u is the unit length, and u and v_1 are orthogonal vectors, the 3D vectors v_1 and v_2 are also orthogonal to each other and of the same length. In the plane spanned by v_1 and v_2, we can write the rotation of θ of v_1 around the vertical direction u as

$$Qv_1 = cv_1 + sv_2.$$

Combining these expressions achieves

$$Qv = Qv_0 + Qv_1 = (uu^t)v + c(I - uu^t)v + su \times v.$$

Finally, rearranging terms and remembering that $U = uu^t - I$, we obtain the desired result:

$$\begin{aligned} Qv &= v + su \times v + (1 - c)(uu^t - I)v \\ &= (I + sU + (1 - c)U^2)v. \end{aligned}$$

So far, we've seen that we can represent a 3D rotation specified by θ and u as a rotation matrix $Q = Q(\theta, u)$. We also know that mapping $(\theta, u) \mapsto Q(\theta, u)$ isn't one-to-one because rotations around the same vector that differ by 2π produce identical rotation matrices. But can we describe all rotation matrices $Q \in SO(3)$ as the rotation of a certain angle around a unit vector? If so, how can we determine a suitable vector and angle?

Background Information

Here are five ways to specify a rotation as a turn of angle θ around a unit-length vector u.

- *The Rodrigues formula*. Here, we specify θ by its sine, $s = \sin(\theta)$, and cosine, $c = \cos(\theta)$, and we specify u explicitly. We can specify θ in radians or degrees.
- *Exponential maps*. We specify θ by an unconstrained 3D vector, $v = \theta u$, equal to the product of θ, specified in radians by the unit-length axis of rotation u.
- *Quaternions*. We specify θ by a 3D vector, $v = su$, of a length less than 1 and equal to the product of the sine of half the angle of the rotation, $\sin(\theta/2)$, and the unit-length rotation axis u.
- *Cayley's rational parameterization*. We specify θ by an unconstrained 3D vector, $v = -\mu u$, equal to the product of the negative of the tangent of half the angle of the rotation, $\mu = \tan(\theta/2)$, and the unit-length axis of rotation u.
- *Two unit-length vectors*. Here, θ is the angle formed by the two unit-length vectors, u_0 and u_1; u is the vector product, $u_0 \times u_1$, of the two vectors normalized to the unit length (see Figure 1 in the main article). The rotation transforms u_0 to u_1.

The Exponential Map

To answer the previous question, I introduce the exponential map, which is defined by a power series:

$$v \mapsto e^V = I + \sum_{n=1}^{\infty} \frac{1}{n!} V^n \,, \tag{1}$$

where V is the skew-symmetric matrix defined by $v(V = \theta U)$. This is a well-defined formula because the power series on the right-hand side converges absolutely to a finite square matrix for every square matrix V, not only for skew-symmetric matrices. I leave it to you to verify that e^V is an orthogonal matrix when V is skew-symmetric.

The Rodrigues formula turns out to be an efficient way to evaluate the exponential map. In fact, we'll show now that $e^V = Q(\theta, u)$ for $v = \theta u$ (the value of u is irrelevant if $\theta = 0$). It follows from equation 1 that

$$e^V = e^{\theta U} = I + \sum_{n=1}^{\infty} \frac{\theta^n}{n!} U^n \,.$$

From the identity $U^3 + U = 0$, which you can verify by direct expansion, it follows that $U^{2k+1} = (-1)^k U$ for $k \geq 0$, and $U^{2k} = -(-1)^k U^2$ for $k \geq 1$.

By splitting the previous series expansion into even and odd terms, replacing these identities, and rearranging terms, we obtain

$$e^{\theta U} = I + \left(\sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+1)!} \right) U + \left(1 - \sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k}}{(2k)!} \right) U^2 .$$

The two power series of the real number θ in this expression are nothing but the power series expansions of $s = \sin(\theta)$ and $c = \cos(\theta)$, respectively. This follows from Euler's formula $e^{j\theta} = \cos\theta + j\sin\theta$. (This formula should be familiar to readers exposed to signal processing.) We can consider Euler's formula to define c and s. Here, $e^{j\theta}$ is the exponential of a complex number, which is defined by the same power series expansion as the exponential of a real number or of the square matrix V showed previously, and V is the square root of -1:

$$e^{j\theta} = \sum_{n=0}^{\infty} \frac{(j\theta)^n}{n!} = \left(\sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k}}{(2k)!} \right) + j \left(\sum_{k=0}^{\infty} \frac{(-1)^k \theta^{2k+1}}{(2k+1)!} \right).$$

The Logarithm Map

The exponential map of Equation 1 with the entire euclidean 3D space as the domain is onto (surjective). That is, we can represent every rotation as an exponential of a skew-symmetric matrix. However, this map isn't one-to-one. I won't prove these facts here; I leave the task to you. But I'll show you how to compute the inverse map.

First, we restrict the domain to a smaller open subset of $\mathbb{R}^3$, where the exponential map is one-to-one. To determine such a subset, we note that the map $\theta \mapsto e^{\theta U}$ with fixed u is 2π-periodic. For $\pi \le \theta \le 2\pi$, we have

$$e^{\theta U} = e^{(2\pi-\theta)(-U)}.$$

That is, the rotation of θ around u is equal to the rotation of angle $2\pi - \theta$ around $-u$. When we restrict the domain to the open ball of radius π,

$$\Omega_\pi = \{v : \|v\| < \pi\} = \{\theta u : 0 \le \theta < \pi \text{ , } \|u\| = 1\},$$

the exponential map becomes one-to-one, but we can't represent rotations of angle π. The image of Ω_π through the exponential map—that is, the set of rotations of an angle less than π around an arbitrary unit-length vector—is an open neighborhood of the identity in the Lie group of 3D rotations $SO(3)$.

The set of rotations of angle π around an arbitrary u are those satisfying

$$|Q + I| = 0.$$

This is because, in such a case, any vector u_1 orthogonal to u satisfies $Qu_1 = -u_1$. That is, it's an eigenvalue of Q corresponding to the eigenvalue -1.

On the set of rotations of an angle less than π, the exponential map has a well-defined inverse: the logarithm map. To compute the logarithm of Q of an angle less than π, we can follow these steps:

$$\begin{cases} c = \left(1 - \text{trace}(Q) - 1\right)/2 \\ V = \left(Q - Q^t\right)/2 \\ s = \|v\| \\ u = v/s \\ \theta = \text{angle}_{[0,\pi)}(s,c). \end{cases}$$

Where do these steps come from? Since Q is in the exponential map's range, we can write it as $Q = Q(\theta, u) = I + sU + (1 - c)U^2$. Likewise, because the transpose of Q is equal to the inverse of Q, we have $Q^t = Q - 1 = Q(-\theta, u) = I - sU + (1 - c) U^2$. Subtracting Q^t from Q, we obtain the skew-symmetric matrix $V = (Q - Q^t)/2 = sU$. So, $s = \|v\|$ must be equal to the norm of v. This number can be equal to zero but doesn't have to be.

If s isn't equal to zero, we can obtain u by normalizing v to $u = v/s$. But we haven't yet uniquely determined the rotation's angle because there are two angles, θ and $\pi - \theta$, between 0 and π for which the sine attains the same value. To uniquely identify the angle, we must determine whether θ's cosine is positive or negative. We can compute this from Q's trace:

$$\begin{aligned} \text{trace}(Q) &= \text{trace}(I) + s\,\text{trace}(U) + (1-c)\,\text{trace}(U^2) \\ &= 3 + 0 + (1-c)(-2) \\ &= 2c + 1. \end{aligned}$$

This is because $\text{trace}(U^2) = \text{trace}(uu^t - I) = \|u\|^2 - 3 = -2$. It follows that $c = (1 - \text{trace}(Q) - 1)/2$.

On the other hand, if $Q - Q^t = 0$, or equivalently if $s = 0$, because $s^2 + c^2 = 1$, we have $c = 1$ or $c = -1$. The $c = 1$ case corresponds to $\theta = 0$, with the identity matrix $Q = I$. The case $c = -1$ corresponds to a rotation of angle $\theta = \pi$. In this case, the rotation matrix has this form:

$$Q = I + sU + (1 - c)U^2 = I + 2(uu^t - I) = 2uu^t - I.$$

By rearranging terms, we obtain

$$\frac{1}{2}(Q+I) = uu^t = \begin{pmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{pmatrix}.$$

Because $uu^t = (-u)(-u)^t$, it is clear now that if u is a solution to this problem, then so is $-u$.

Finally, we can obtain the magnitudes of the components of u by computing the square roots of this matrix's diagonal elements, and the signs from the off-diagonal elements:

$$\begin{cases} u_x = \sqrt{(Q_{11}+1)/2} \\ \text{if } Q_{12} < 0 \text{ and } Q_{13} < 0 \text{ then} \\ \quad u_x = -u_x \\ u_y = \sqrt{(Q_{22}+1)/2} \\ \text{if } Q_{12} < 0 \text{ and } Q_{23} < 0 \text{ then} \\ \quad u_y = -u_y \\ u_z = \sqrt{(Q_{33}+1)/2} \\ \text{if } Q_{13} < 0 \text{ and } Q_{23} < 0 \text{ then} \\ \quad u_z = -u_z\,. \end{cases}$$

Quaternions

Since in the Rodrigues formula we have $s^2 + c^2 = 1$, let's consider the following map as a candidate to replace the exponential map:

$$\begin{cases} \Omega_1 \to SO(3) \\ su \mapsto I + sU + (1-c)U^2\,, \end{cases}$$

where Ω_1 is the open unit ball in $\mathbb{R}^3$, $0 \leq s < 1$, $c = \sqrt{1-s^2}$, and $|u| = 1$. Rather than the rotation's angle, we use the angle's sine as the magnitude for the parameter vector $v = su$. This parameterization lets us avoid computing θ's sine and cosine. It is also one-to-one, and we need only one square root to evaluate it:

$$\begin{cases} \Omega_1 \to SO(3) \\ v \mapsto I + V + \dfrac{1-\sqrt{1-\|v\|^2}}{\|v\|^2} V^2\,, \end{cases} \tag{2}$$

where $V = sU$ is the skew-symmetric matrix corresponding to $v = su$.

The only potential problem is that, because c is nonnegative here, the image of Ω_1 through this parameterization is the set of rotations of an angle less than $\pi/2$ and therefore isn't an onto map. To solve this problem, we take the magnitude $s = \|v\|$ of a vector in Ω_1 as $\sin(\theta/2)$ rather than $\sin(\theta)$. Because

$$\begin{cases} 1-\cos(\theta) = 2\sin(\theta/2)^2 = 2s^2 \\ \sin(\theta) = 2\sin(\theta/2)\cos(\theta/2) = 2sc\,, \end{cases}$$

the parameterization

$$\begin{cases} \Omega_1 \to SO(3) \\ su \mapsto I + 2scU + 2s^2U^2 \end{cases} \tag{3}$$

covers the same open set of rotations as the exponential map. We can safely compute c as $\sqrt{1-s^2}$ because, in this case, $0 \leq \sin(\theta/2)$ and $\cos(\theta/2) \leq 1$. Once again, by combining terms, we can evaluate this parameterization with a single square root:

Quaternions are particularly popular in computer graphics because only four parameters are needed to represent a rotation, as opposed to nine for matrices.

$$\begin{cases} \Omega_1 \to SO(3) \\ v \mapsto I + 2\sqrt{1-\|v\|^2}\,V + 2V^2\,. \end{cases}$$

How do we compute this parameterization's inverse? I leave this as another exercise for you, but with two hints. Computing the norm of v from the trace of Q is easy. If this number is zero, Q is the identity matrix. Otherwise, you can compute v normalized to the unit length from the skew-symmetric part $(Q - Q^t)/2$ of Q.

It is interesting to observe that this is the parameterization of the special orthogonal group $SO(3)$ associated with the quaternions. Quaternions are particularly popular in computer graphics because only four parameters are needed to represent a rotation, as opposed to nine for matrices. Also, the rotations' composition corresponds to the product of quaternions, although the exponential map uses only three unconstrained parameters.

A quaternion is a pair (a, b), where $a \in \mathbb{R}$ is a scalar and $b \in \mathbb{R}^3$ is a vector. The formula product of two quaternions (a_1, b_1) and (a_2, b_2) is

$$(a_1, b_1)\cdot(a_2 b_2) = \left(a_1 a_2 - b_1^t b_2,\, a_1 b_2 + a_2 b_1 + b_1 \times b_2\right).$$

If $Q = Q(\theta, u)$ is the rotation of angle $0 \leq \theta < \pi$ around u, and v is an arbitrary 3D vector, the following identity is satisfied:

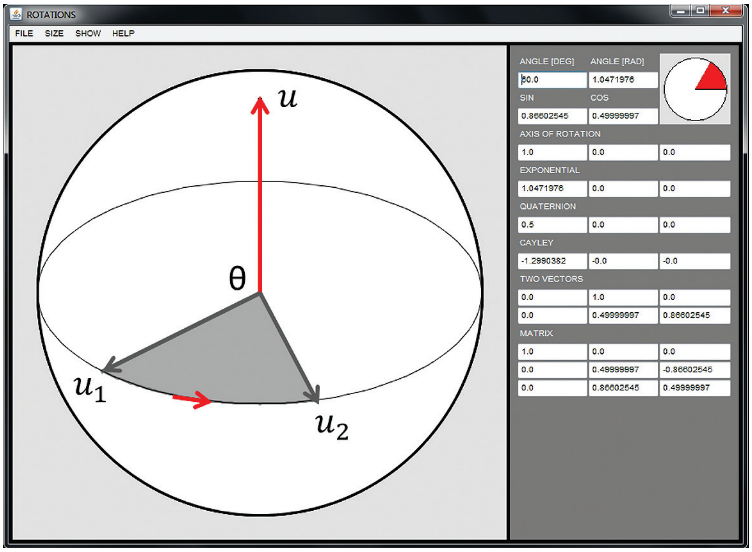

Figure 1. A screenshot of the software program I've written to go along with this article. To access the software, go to http://mesh.brown.edu/rotations.

$$(0, Qv) = (c, su) \cdot (0, v) \cdot (c, -su),$$

where $c = \cos(\theta/2)$ and $s = \sin(\theta/2)$. You can verify this identity by direct computation.

Cayley's Rational Parameterization

To avoid computing square roots, we can use the following less-known parameterization of the set of rotations of an angle less than π, due to Cayley:

$$\begin{cases} \mathbb{R}^3 \to SO(3) \\ w \mapsto (I+W)^{-1}(I-W) = (I-W)(I+W)^{-1}, \end{cases}$$

where W is the skew-symmetric matrix corresponding to the 3D vector w. Because $|I - W| = |I + W| = 1 + \|w\|^2$, this function is well defined for any w. Surprisingly, this parameterization's inverse uses the same formula: if Q is a rotation of an angle less than π, we have $|I + Q| \neq 0$ and $W = (I + Q)^{-1}(I - Q)$.

To establish this parameterization's relation to the other parameterizations, we observe that

$$(I+W)^{-1} = I - \frac{1}{1+\|w\|^2} W + \frac{1}{1+\|w\|^2} W^2$$

follows from the identity $W^3 + \|w\|^2 W = 0$. So,

$$Q = (I+W)^{-1}(I-W) = I - \frac{2}{1+\|w\|^2} W + \frac{2}{1+\|w\|^2} W^2,$$

which we can evaluate without square roots. If we write $w = -\mu u$ with $|u| = 1$ in the previous equation, we obtain

$$Q = I + \frac{2\mu}{1+\mu^2} U + \frac{2\mu^2}{1+\mu^2} U^2,$$

which is equal to Equation 3's parameterization, with $\mu = \tan(\theta/2)$.

To apply one of these rotations to a vector v, we can follow these steps:

$$\begin{cases} \mu = 2/\left(1+\|w\|^2\right) \\ v_1 = w \times v \\ v_2 = w \times v_1 \\ Qv = v - \mu(v_1 - v_2), \end{cases} \qquad (4)$$

which require 19 multiplications and 14 additions. On the other hand, we can construct Q by following these steps:

$$\begin{cases} \mu = 2/\left(1+\|w\|^2\right) \\ Q = I - \mu W \\ Q = Q + \mu\left(ww^t - I\right), \end{cases}$$

which require 22 multiplications and 14 additions.

As I mentioned before, multiplying a 3 × 3 matrix by a vector requires nine multiplications and six additions. So, if the same rotation has to be applied to many vectors, the cost of constructing Q and applying the rotation by matrix multiplication is roughly half of that for applying Equation 4 to each vector. However, if the rotation has to be applied to one or two vectors before w changes, Equation 4 is less expensive.

Rotation from Two Unit Vectors

Another way to specify a rotation is from two linearly independent unit-length vectors u_1 and u_2, as a rotation $Q = Q(u_1, u_2)$ that transforms u_1 to $u_2 = Qu_1$. Because an infinite number of such rotations exist, we further require that the rotation minimizes the turning angle among all such rotations. This additional constraint turns out to lead to a unique solution. This is a rotation around the unit vector u normal to the plane spanned by the two vectors—that is, $u_1 \times u_2$ normalized to unit length—and the angle θ is the angle formed by the two vectors. Figure 1 shows a screenshot of the software program I've written to go along with this article. To access the software, go to http://mesh.brown.edu/rotations.

We can use the Rodrigues formula to compute $Q = Q(\theta, u)$ without explicitly computing the rotation's angle. This is because we can determine u and θ's sine and cosine directly from the two vectors:

$$\begin{cases} c = u_1^t u_2 \\ s = \sqrt{1-c^2} \\ u = u_1 \times u_2 \, / \, s \, . \end{cases}$$

We can extend this with continuity to the case $u_1 = u_2$ by defining Q as the identity matrix. However, we can't extend it to the case $u_1 = -u_2$ because it lacks uniqueness. For every u orthogonal to u_1, the rotation $Q(\pi, u)$ transforms u_1 into u_2, and no rotation of angle less than π transforms u_1 to u_2.

But we can also compute $Q(u_1, u_2)$ without evaluating square roots. Let $v = u_1 \times u_2$ and $c = u_1^t u_2$. Because $v = su$, we have $I + V = I + sU$ and

$$(1-c)U^2 = \frac{1-c}{s^2}V^2 = \frac{1-c}{1-c^2}V^2 = \frac{1}{1+c}V^2 ,$$

and it follows that

$$Q(u_1, u_2) = I + V + \frac{1}{1+c^2}V^2 .$$

Note the relation between this formula and the one in Equation 2. Here, we avoid computing the square root because we can compute the angle's cosine independently as an inner product.

Another way to compute $Q(u_1, u_2)$ without square roots is by using Cayley's rational parameterization. We can verify that when $Q = (I + W)^{-1}(I - W)$, we can determine the vector w defining this rotation as

$$w = (u_1 \times u_2) \, / \, (1 - u_1^t u_2) .$$

As you can see, despite these parameterizations' apparent differences, they're all closely related. In a future article, I'll explore applications of these methods, particularly to digital geometry-processing problems.

Gabriel Taubin *is a professor in Brown University's School of Engineering and the editor in chief of this magazine. Contact him at taubin@brown.edu.*