Hand-Eye Calibration via the Homogeneous Matrix Equation $AX = XB$

Zixing Jiang[*]

December 4, 2024

1 Introduction

A generic solution to the hand-eye calibration problem, whether in the *eye-in-hand* or *eye-to-hand* setup, involves solving a homogeneous matrix equation of the form $AX = XB$ This documentation first shows how to formulate the problem as this equation and then introduces a *rotation-then-translation* method to compute the closed-form solution of this equation (the exact method used in the minimal_handeye_ros2 package).

Notations In this documentation we use the following notation conventions to represent mathematical variables:

- Matrices: *UPPERCASE ITALIC LETTERS*, e.g., A, B, X, T, R. In particular, for matrices whose elements are all the same, we write them in the form of $\lambda_{m\times n}$, where λ is a scalar representing the values of all the elements, m is the number of rows and n is the number of columns, e.g., $0_{1\times 3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.
- Vectors and quaternions written as vectors: **lowercase bold letters**, e.g., $\mathbf{v}$, $\mathbf{t}$.
- Scalars and quaternions: *lowercase italic letters*, e.g., λ, v_x, v_y, v_z, q.

2 Eye-in-hand Setup

In a common *eye-in-hand* setup (Figure 1), the camera is mounted on the wrist of a robotic manipulator. The goal of hand-eye calibration is to estimate the transformation from the manipulator end-effector frame to the wrist-mounted camera frame. To do this, an external marker that can be recognized by the camera is required. The marker should remain stationary during the calibration process.

[*]Email: zixingjiang@outlook.com

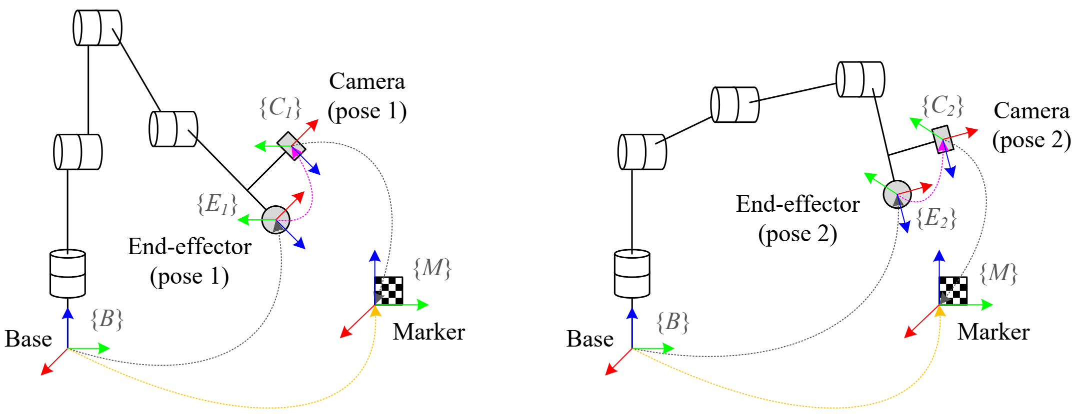

Figure 1: Eye-in-hand setup

The calibration leverages the invariance of the marker's pose relative to the manipulator base. Let $\{B\}$ denote the manipulator base frame, $\{M\}$ denote the marker frame, $\{E\}$ denote the manipulator end-effector frame, and $\{C\}$ denote the camera frame. We have:

$$ {}^{B}T_{M} = {}^{B}T_{E}\,{}^{E}T_{C}\,{}^{C}T_{M} \tag{1} $$

where ${}^{X}T_{Y} \in \mathbb{R}^{4\times 4}$ denote the the homogeneous transformation matrix from frame $\{X\}$ to frame $\{Y\}$. In equation (1), ${}^{B}T_{E}$ can be acquired by robot kinematics, ${}^{C}T_{M}$ can be acquired by camera perception, and ${}^{E}T_{C}$ is what we want to calibrate. Since ${}^{B}T_{M}$ is invariant as the marker is stationary, for pose 1 and pose 2 shown in Figure 1, we have:

$$ {}^{B}T_{E_1}\,{}^{E_1}T_{C_1}\,{}^{C_1}T_{M} = {}^{B}T_{M} = {}^{B}T_{E_2}\,{}^{E_2}T_{C_2}\,{}^{C_2}T_{M} \tag{2} $$

Since ${}^{E_1}T_{C_1} = {}^{E_2}T_{C_2} = {}^{E}T_{C}$ as the camera frame and end-effector frame are relatively stationary, equation (2) can be simplified as:

$$ {}^{B}T_{E_1}\,{}^{E}T_{C}\,{}^{C_1}T_{M} = {}^{B}T_{E_2}\,{}^{E}T_{C}\,{}^{C_2}T_{M} \tag{3} $$

By pre-multiplying ${}^{B}T_{E_2}^{-1}$ and post-multiplying ${}^{C_1}T_{M}^{-1}$ on both sides of equation (3), we have:

$$ {}^{B}T_{E_2}^{-1}\,{}^{B}T_{E_1}\,{}^{E}T_{C} = {}^{E}T_{C}\,{}^{C_2}T_{M}\,{}^{C_1}T_{M}^{-1} \tag{4} $$

of the form $AX = XB$.

3 Eye-to-hand Setup

In a common *eye-to-hand* setup (Figure 2), the camera is installed externally. The goal of hand-eye calibration is to estimate the transformation from the manipulator base frame to the external camera frame. To do this, a marker mounted on the manipulator wrist that can be recognized by the camera is required. The marker should remain relatively stationary to the manipulator end-effector during the calibration process.

The calibration leverages the invariance of the marker's pose relative to the manipulator end-effector. Similar to the eye-in-hand setup, we have:

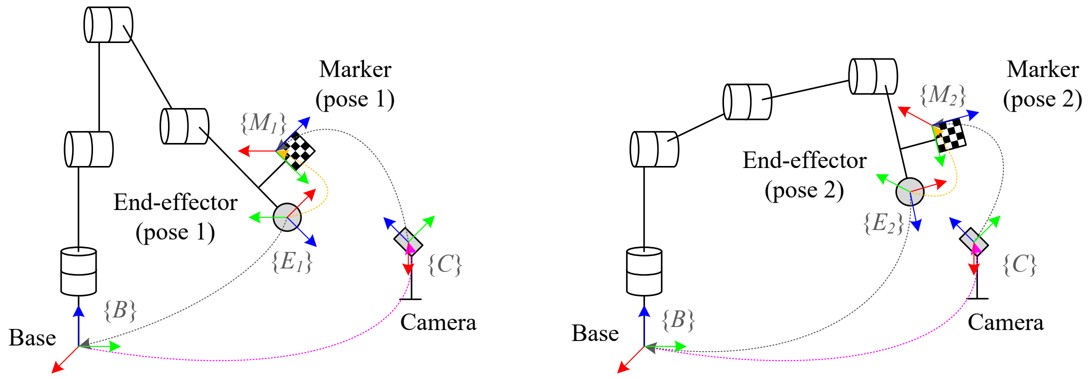

Figure 2: Eye-to-hand setup

$$ {}^{E}T_{M} = {}^{E}T_{B}\,{}^{B}T_{C}\,{}^{C}T_{M} \tag{5}$$

where ${}^{E}T_{B}$ can be acquired by robot kinematics, ${}^{C}T_{M}$ can be acquired by camera perception, and ${}^{B}T_{C}$ is what we want to calibrate. Since ${}^{E}T_{M}$ is invariant as the marker is relatively stationary to the manipulator end-effector, for pose 1 and pose 2 shown in Figure 2, we have:

$$ {}^{E_1}T_{B}\,{}^{B}T_{C}\,{}^{C}T_{M_1} = {}^{E_2}T_{B}\,{}^{B}T_{C}\,{}^{C}T_{M_2} \tag{6}$$

By pre-multiplying ${}^{E_2}T_{B}^{-1}$ and post-multiplying ${}^{C}T_{M_1}^{-1}$ on both sides of equation (6), we have:

$$ {}^{E_2}T_{B}^{-1}\,{}^{E_1}T_{B}\,{}^{B}T_{C} = {}^{B}T_{C}\,{}^{C}T_{M_2}\,{}^{C}T_{M_1}^{-1} \tag{7}$$

of the form $AX = XB$.

4 Solve $AX = XB$

In this section, we will show how to solve the equations of the form $AX = XB$ derived in the previous sections.

4.1 Decomposition

Note that A, B, and X are all (or products of) homogeneous transformation matrices, we can expand them in terms of rotation matrices $R \in \mathbb{R}^{3\times3}$ and translation vectors $\mathbf{t} \in \mathbb{R}^3$.

$$ AX = XB \tag{8a}$$

$$ \Leftrightarrow \begin{bmatrix} R_A & \mathbf{t}_A \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_X & \mathbf{t}_X \\ 0_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} R_X & \mathbf{t}_X \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_B & \mathbf{t}_B \\ 0_{1\times3} & 1 \end{bmatrix} \tag{8b}$$

$$ \Leftrightarrow \begin{bmatrix} R_A R_X & R_A\mathbf{t}_X + \mathbf{t}_A \\ 0_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} R_X R_B & R_X\mathbf{t}_B + \mathbf{t}_X \\ 0_{1\times3} & 1 \end{bmatrix} \tag{8c}$$

Hence, we can decompose the homogeneous matrix equation (8a) into two equations: a matrix equation depending on rotation:

$$R_A R_X = R_X R_B \tag{9}$$

and a vector equation depending both on rotation and translation:

$$(R_A - I)\,\mathbf{t}_X = R_X \mathbf{t}_B - \mathbf{t}_A \tag{10}$$

where $I \in \mathbb{R}^{3\times 3}$ is the identify matrix. Particularly, R_A and R_B are similar matrices and therefore have the same eigenvalues because of $R_A = R_X R_B R_X^{-1}$ implied by equation (9).

A well-known property of three-dimensional rotation matrices is that one of their eigenvalues is equal to 1[1]. Let $\mathbf{n}_B$ be an eigenvector of R_B associated with eigenvalue 1[2], then we have $R_B \mathbf{n}_B = \mathbf{n}_B$. Post-multiplying $\mathbf{n}_B$ on both sides of equation (9) yields:

$$R_A R_X \mathbf{n}_B = R_X R_B \mathbf{n}_B = 1 R_X \mathbf{n}_B \tag{11}$$

of the eigenvalue problem form $A\mathbf{v} = \lambda\mathbf{v}$. Therefore, $R_X \mathbf{n}_B$ is equal to $\mathbf{n}_A$, the eigenvector of R_A associated with eigenvalue 1.

$$\mathbf{n}_A = R_X \mathbf{n}_B \tag{12}$$

To conclude, solving homogeneous matrix equation (8a) is equivalent to solve matrix equation (12) and vector equation (10).

4.2 Solve Rotation then Translation

Equations (12) and (10) are of the form:

$$\mathbf{v}' = R\mathbf{v} \tag{13a}$$
$$(K - I)\mathbf{t} = R\mathbf{p} - \mathbf{p}' \tag{13b}$$

where $R \in \mathbb{R}^{3\times 3}$ and $\mathbf{t} \in \mathbb{R}^3$ are parameter to be estimated (rotation and translation), $\mathbf{v}$, $\mathbf{v}'$, $\mathbf{p}$, $\mathbf{p}'$ are $\mathbb{R}^3$ vectors, $K \in \mathbb{R}^{3\times 3}$ is an orthogonal matrix and $I \in \mathbb{R}^{3\times 3}$ is the identity matrix. In this document, we introduce a *rotation-then-translation* technique to solve these two equations. That is, first solve R through (13a), and then substitute it into (13b) to solve $\mathbf{t}$.

Solve Rotation Solving (13a) can be cast into minimizing the following positive error function over the constraint that R should be a $\mathbb{R}^{3\times 3}$ rotation matrix:

$$\begin{aligned} \min_R \quad & f_1(R) = ||\mathbf{v}' - R\mathbf{v}||^2 \\ \text{s.t.} \quad & R^T R = I \\ & det(R) = 1 \end{aligned} \tag{14}$$

[1]Obviously, for a vector $\mathbf{n}$ that is collinear with the rotation axis of a rotation matrix R, the eigenvalue equation $R\mathbf{n} = \lambda\mathbf{n}$ holds with $\lambda = 1$. Therefore $\mathbf{n}$ is an eigenvector of R associated with eigenvalue 1.

[2]We can easily find such a unnormalized eigenvector for a rotation matrix by adopting the vector part of equation (81) in the Appendix.

Solving this matrix-variable minimization problem directly is difficult because it involves an orthogonality constraint, which is non-convex. Fortunately, we can convert it into a convex vector-variable minimization problem by representing the rotation R in terms of its equivalent unit quaternion.

Let $q \in \mathbb{H}_1 = \{q \in \mathbb{H} : ||q|| = 1\}$ be a unit quaternion representing the same rotation as R, and $\mathbf{q} \in \mathbb{R}^4$ be its vector representation. According to (75) in the Appendix, The rotation transform $R\mathbf{v}$ in equation (13a) can be written in the equivalent form of quaternion multiplication $\mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}}$, where

$$\mathbf{v}_q = \begin{bmatrix} 0 & \frac{v_x}{||\mathbf{v}||^2} & \frac{v_y}{||\mathbf{v}||^2} & \frac{v_z}{||\mathbf{v}||^2} \end{bmatrix}^T \tag{15}$$

is the vector representation of a pure imaginary unit quaternion constructed from vector $\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$. Then we can rewrite the error function f_1 (14) in terms of unit quaternion q:

$$\begin{aligned}
f_1(R) &= ||\mathbf{v}' - R\mathbf{v}||^2 && \text{(16a)}\\
\Rightarrow g_1(\mathbf{q}) &= \left|\left|\mathbf{v}'_q - \mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}}\right|\right|^2 && \text{(16b)}\\
&= \left|\left|\mathbf{v}'_q - \mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}}\right|\right|^2 ||\mathbf{q}||^2 && \text{(16c)}\\
&= \left|\left|\mathbf{v}'_q * \mathbf{q} - \mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}} * \mathbf{q}\right|\right|^2 && \text{(16d)}\\
&= \left|\left|\mathbf{v}'_q * \mathbf{q} - \mathbf{q} * \mathbf{v}_q\right|\right|^2 && \text{(16e)}\\
&= \left|\left|Q(\mathbf{v}'_q)\mathbf{q} - W(\mathbf{v}_q)\mathbf{q}\right|\right|^2 && \text{(16f)}\\
&= \left|\left|\left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)\mathbf{q}\right|\right|^2 && \text{(16g)}\\
&= \left(\left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)\mathbf{q}\right)^T \left(\left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)\mathbf{q}\right) && \text{(16h)}\\
&= \mathbf{q}^T \left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)^T \left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)\mathbf{q} && \text{(16i)}
\end{aligned}$$

where $\mathbf{v}'_q$, similar to (15), is the vector representation of a pure imaginary unit quaternion constructed from vector $\mathbf{v}'$, Q is the pre-multiplication matrix of quaternion multiplication (see (54) in the Appendix), and W is the post-multiplication matrix of quaternion multiplication (see (55) in the Appendix).

Step-wise explanation:

- From (16b) to (16c), we leveraged the property that the unit quaternion has a norm of 1: $||q|| = 1$.
- From (16c) to (16d), we leveraged the norm identity (68) in the Appendix.
- From (16d) to (16e), we eliminated $\bar{\mathbf{q}} * \mathbf{q}$ as $\mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}} * \mathbf{q} = W(\bar{\mathbf{q}} * \mathbf{q})\,(\mathbf{q} * \mathbf{v}_q)$, where $W(\bar{\mathbf{q}} * \mathbf{q})$, according to (60), (62), and (64) in the Appendix, is equal to the identity matrix I (note that $q * \bar{q} = 1$ as q is a unit quaternion).
- From (16e) to (16f), we adopted the matrix notation of quaternion multiplication (56).

For brevity, let:

$$\mathcal{A} = \left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right)^T \left(Q(\mathbf{v}'_q) - W(\mathbf{v}_q)\right) \tag{17}$$

Note that the $\mathbb{R}^{4\times 4}$ matrix $\mathcal{A}$ is symmetric as $\mathcal{A}^T = \mathcal{A}$.

Then we can write the error function g_1 (16i) in a quadratic form, where $\mathcal{A}$ is its associated symmetric matrix:

$$g_1(\mathbf{q}) = \mathbf{q}^T \mathcal{A} \mathbf{q} \tag{18}$$

We should minimize (18) over the constraint that q should be a unit quaternion:

$$\begin{aligned} \min_{\mathbf{q}} \quad & g_1(\mathbf{q}) = \mathbf{q}^T \mathcal{A} \mathbf{q} \\ \text{s.t.} \quad & 1 - \mathbf{q}\cdot\mathbf{q} = 0 \end{aligned} \tag{19}$$

This constrained minimization problem has an elegant closed-form solution, as the associated symmetric matrix $\mathcal{A}$, according to the spectral theorem[3], can be diagonalized as:

$$\mathcal{A} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}^T \tag{20}$$

where $\lambda_i \in \mathbb{R}, i \in \{1,2,3,4\}$ are real eigenvalues of $\mathcal{A}$, and $\mathbf{u}_i \in \mathbb{R}^4, i \in \{1,2,3,4\}$ are normalized eigenvectors associated to λ_i. Matrix $\begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}$ is orthogonal as unit eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$ form an orthogonal basis. Plugging (20) into (18) yields:

$$\begin{aligned} g_1(\mathbf{q}) &= \mathbf{q}^T \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}^T \mathbf{q} && \text{(21a)} \\ &= \left(\begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}^T \mathbf{q}\right)^T \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}^T \mathbf{q}\right) && \text{(21b)} \\ &= \begin{bmatrix} \mathbf{u}_1\cdot\mathbf{q} \\ \mathbf{u}_2\cdot\mathbf{q} \\ \mathbf{u}_3\cdot\mathbf{q} \\ \mathbf{u}_4\cdot\mathbf{q} \end{bmatrix}^T \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1\cdot\mathbf{q} \\ \mathbf{u}_2\cdot\mathbf{q} \\ \mathbf{u}_3\cdot\mathbf{q} \\ \mathbf{u}_4\cdot\mathbf{q} \end{bmatrix} && \text{(21c)} \\ &= \sum_{i=1}^{4} \lambda_i \left(\mathbf{u}_i \cdot \mathbf{q}\right)^2 && \text{(21d)} \end{aligned}$$

Since the norm is preserved by multiplying an orthogonal matrix, we have:

$$\left|\left|\begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 \end{bmatrix}^T \mathbf{q}\right|\right|^2 = \sum_{i=1}^{4} \left(\mathbf{u}_i\cdot\mathbf{q}\right)^2 = ||\mathbf{q}||^2 = 1 \tag{22}$$

[3]Recommended texts on spectral theorem and symmetric matrix: Harvard Math's and UC Davis Math's lecture notes.

With (22), we can view (21d) as a weighted sum of λ_i, where the sum of the weights $\sum_{i=1}^{4} (\mathbf{u}_i \cdot \mathbf{q})^2 = 1$. Without loss of generality, let $\lambda_i, i \in \{1, 2, 3, 4\}$ be a ordered permutation, i.e., $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$. Obviously (21d) is bounded by:

$$\lambda_1 \leq \sum_{i=1}^{4} \lambda_i (\mathbf{u}_i \cdot \mathbf{q})^2 \leq \lambda_4 \tag{23}$$

The minimum value is obtained when the weights are

$$(\mathbf{u}_i \cdot \mathbf{q})^2 = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases} \tag{24}$$

which implies $\mathbf{q} = \pm \mathbf{u}_1$. Hence, we can conclude that:

The solution to the minimization problem (19) is a unit eigenvector corresponding to the smallest eigenvalue of its associated symmetric matrix $\mathcal{A}$.

which leads to the closed-form solution introduced by (Faugeras and Hebert, 1986) for finding the best rotation between two sets of 3D features. Since unit quaternions double cover three-dimensional rotations, it doesn't matter whether we choose $\mathbf{q} = \mathbf{u}_1$ or $\mathbf{q} = -\mathbf{u}_1$. After determining the unit quaternion $\mathbf{q}$, we can use formula (72) in the Appendix to find the corresponding rotation matrix R, which is the closed-form solution of equation (13a).

Solve Translation Once the rotation R is determined, we can substitute it into equation (13b) to solve the translation $\mathbf{t}$. If the matrix $(K - I)$ is invertible, we can easily obtain a closed-form solution:

$$\mathbf{t} = (K - I)^{-1} (R\mathbf{p} - \mathbf{p}') \tag{25}$$

If $(K - I)$ is not invertible, we can use its Moore-Penrose pseudoinverse to get an approximate solution:

$$\mathbf{t} = (K - I)^{+} (R\mathbf{p} - \mathbf{p}') \tag{26}$$

which gives a "best fit" solution to the least-squares problem:

$$\min_{\mathbf{t}} \quad f_2(\mathbf{t}) = ||(K - I)\,\mathbf{t} - (R\mathbf{p} - \mathbf{p}')||^2 \tag{27}$$

For brevity, let $\mathcal{B} = K - I \in \mathbb{R}^{3\times 3}$. The pseudoinverse $\mathcal{B}^{+}$ can be computed as follows:

First, decompose $\mathcal{B}$ into its Singular Value Decomposition (SVD):

$$\mathcal{B} = U \Sigma V^T \tag{28}$$

where U and V are orthogonal matrices, and Σ is the diagonal matrix containing the singular values of $\mathcal{B}$:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \tag{29}$$

Then, taking the reciprocal of the none-zero singular values in Σ (if any singular value is 0, leave it unchanged):

$$\Sigma^+ = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 \\ 0 & 0 & \frac{1}{\sigma_3} \end{bmatrix} \tag{30}$$

Finally, the pseudoinverse $\mathcal{B}^+$ can be computed using the SVD:

$$\mathcal{B}^+ = U\Sigma^+V^T \tag{31}$$

4.3 Solve Multiple $AX = XB$

So far, we have been able to solve the $AX = XB$ equation obtained from a pair of robot pose data. In actual calibration, in order to improve accuracy, multiple pairs of robot pose data are often collected. Suppose we record $n + 1$ poses of the robot, by pairing two adjacent poses, we will get n equations:

$$A_1X = XB_1 \tag{32a}$$
$$A_2X = XB_2 \tag{32b}$$
$$A_3X = XB_3 \tag{32c}$$
$$\vdots$$
$$A_nX = XB_n \tag{32d}$$

By decomposing them, we get $2n$ equations:

$$\mathbf{v}_1' = R\mathbf{v}_1 \tag{33a}$$
$$\mathbf{v}_2' = R\mathbf{v}_2 \tag{33b}$$
$$\mathbf{v}_3' = R\mathbf{v}_3 \tag{33c}$$
$$\vdots$$
$$\mathbf{v}_n' = R\mathbf{v}_n \tag{33d}$$

and

$$(K_1 - I)\mathbf{t} = R\mathbf{p}_1 - \mathbf{p}_1' \tag{34a}$$
$$(K_2 - I)\mathbf{t} = R\mathbf{p}_2 - \mathbf{p}_2' \tag{34b}$$
$$(K_3 - I)\mathbf{t} = R\mathbf{p}_3 - \mathbf{p}_3' \tag{34c}$$
$$\vdots$$
$$(K_n - I)\mathbf{t} = R\mathbf{p}_n - \mathbf{p}_n' \tag{34d}$$

Solving these $2n$ equations can be cast into minimizing these two positive error functions (minimize f_1 then f_2):

$$\begin{aligned} \min_R \quad & f_1(R) = \sum_{i=1}^{n} ||\mathbf{v}_i' - R\mathbf{v}_i||^2 \\ \text{s.t.} \quad & R^TR = I \\ & det(R) = 1 \end{aligned} \tag{35}$$

and

$$\min_{\mathbf{t}} \quad f_2(\mathbf{t}) = \sum_{i=1}^{n} ||(K_i - I)\,\mathbf{t} - (R\mathbf{p}_i - \mathbf{p}'_i)||^2 \tag{36}$$

Minimize f_1 Using the same techniques in (16) and (17), we can rewrite the error function f_1 (35) in terms of a unit quaternion:

$$f_1(R) = \sum_{i=1}^{n} ||\mathbf{v}'_i - R\mathbf{v}_i||^2 \tag{37a}$$

$$\Rightarrow g_1(\mathbf{q}) = \sum_{i=1}^{n} \mathbf{q}^T \mathcal{A}_i \mathbf{q} \tag{37b}$$

$$= \mathbf{q}^T \left(\sum_{i=1}^{n} \mathcal{A}_i \right) \mathbf{q} \tag{37c}$$

Let $\mathcal{A} = \sum_{i=1}^{n} \mathcal{A}_i \in \mathbb{R}^{4\times 4}$, then we can transform the matrix-variable minimization problem (35) into the same vector-variable minimization problem as (19):

$$\begin{aligned} \min_{\mathbf{q}} \quad & g_1(\mathbf{q}) = \mathbf{q}^T \mathcal{A} \mathbf{q} \\ \text{s.t.} \quad & 1 - \mathbf{q} \cdot \mathbf{q} = 0 \end{aligned} \tag{38}$$

whose closed-form solution, as shown in the previous sections, is the unit eigenvector associated to the smallest eigenvalue of $\mathcal{A}$. Once $\mathbf{q}$ is determined, we can use formula (72) in the Appendix to find the corresponding rotation matrix R, which is the solution to the original matrix-variable minimization problem (35).

Minimize f_2 The error function f_2 in (36) contains multiple least-squares terms. To find the solution, we can consolidate all the terms into a single least squares problem by stacking the matrices and vectors.

Let $\mathcal{B} \in \mathbb{R}^{3n\times 3}$ be:

$$\mathcal{B} = \begin{bmatrix} K_1 - I \\ K_2 - I \\ \vdots \\ K_n - I \end{bmatrix} \tag{39}$$

and $\mathbf{c} \in \mathbb{R}^{3n}$ be:

$$\mathbf{c} = \begin{bmatrix} R\mathbf{p}_1 - \mathbf{p}'_1 \\ R\mathbf{p}_2 - \mathbf{p}'_2 \\ \vdots \\ R\mathbf{p}_n - \mathbf{p}'_n \end{bmatrix} \tag{40}$$

then the original problem (36) can be written as a single least-squares problem:

$$\min_{\mathbf{t}} \quad f_2(\mathbf{t}) = ||\mathcal{B}\mathbf{t} - \mathbf{c}||^2 \tag{41}$$

Assuming $\mathcal{B}^T\mathcal{B}$ is invertible, then we have the closed-form solution:

$$\mathbf{t} = \left(\mathcal{B}^T\mathcal{B}\right)^{-1} \mathcal{B}^T \mathbf{c} \tag{42}$$

If $\mathcal{B}^T\mathcal{B}$ is not invertible, then using the pseudoinverse, we have:

$$\mathbf{t} = \mathcal{B}^{+}\mathbf{c} \tag{43}$$

Remarks To avoid propagating the error in solving the rotation to solving the translation, we can skip the translation until the rotation converges.

END OF THE MAIN TEXT

Appendix – Quaternion and 3D Rotation

In the main text, we show that using unit quaternions to represent rotations can transform the non-convex matrix-variable minimization problem (14) into a convex vector-variable minimization problem (19). In this appendix, we list some properties of quaternions for readers' reference.

Quaternion Definition

A *quaternion* $q \in \mathbb{H}^4$ is defined as a special complex number with one real part and three imaginary parts:

$$q = q_w + \hat{\mathbf{i}}q_x + \hat{\mathbf{j}}q_y + \hat{\mathbf{k}}q_z \tag{44}$$

with:

$$\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1 \tag{45}$$

where the coefficients q_w, q_x, q_y, and q_z are all real numbers. One may also write quaternion in the form of a $\mathbb{R}^4$ vector:

$$\mathbf{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T \tag{46}$$

which has a basis $\{1, \hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}\}$. q and $\mathbf{q}$ will appear interchangeably in the following text. They represent the same thing. When it comes to vector and matrix operations, we use $\mathbf{q}$ to comply with the vector notation convention in this documentation.

The *conjugate* of quaternion q, denoted by $\bar{q}$, is defined by:

$$\bar{q} = q_w - \hat{\mathbf{i}}q_x - \hat{\mathbf{j}}q_y - \hat{\mathbf{k}}q_z \tag{47}$$

and in vector representation:

$$\bar{\mathbf{q}} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \end{bmatrix}^T \tag{48}$$

According to equation (45), one can verify:

$$\hat{\mathbf{i}}\hat{\mathbf{j}} = -\hat{\mathbf{j}}\hat{\mathbf{i}} = \hat{\mathbf{k}} \tag{49a}$$
$$\hat{\mathbf{j}}\hat{\mathbf{k}} = -\hat{\mathbf{k}}\hat{\mathbf{j}} = \hat{\mathbf{i}} \tag{49b}$$
$$\hat{\mathbf{k}}\hat{\mathbf{i}} = -\hat{\mathbf{i}}\hat{\mathbf{k}} = \hat{\mathbf{j}} \tag{49c}$$

which is helpful for deriving quaternion multiplication and implies that quaternion multiplication is non-commutative.

Quaternion Algebra

Addition Adding quaternions gives a quaternion:

$$q + r = \left(q_w + \hat{\mathbf{i}}q_x + \hat{\mathbf{j}}q_y + \hat{\mathbf{k}}q_z\right) + \left(r_w + \hat{\mathbf{i}}r_x + \hat{\mathbf{j}}r_y + \hat{\mathbf{k}}r_z\right) \tag{50a}$$
$$= (q_w + r_w) + (q_x + r_x)\hat{\mathbf{i}} + (q_y + r_y)\hat{\mathbf{j}} + (q_z + r_z)\hat{\mathbf{k}} \tag{50b}$$

[4] The use of $\mathbb{H}$ to represent the set of quaternions is to commemorate Sir William Rowan Hamilton, who made his discovery of the algebra of quaternions in 1843.

Dot product Dot product of quaternions gives a scalar:

$$q \cdot r = \left(q_w + \hat{\mathbf{i}}q_x + \hat{\mathbf{j}}q_y + \hat{\mathbf{k}}q_z\right) \cdot \left(r_w + \hat{\mathbf{i}}r_x + \hat{\mathbf{j}}r_y + \hat{\mathbf{k}}r_z\right) \tag{51a}$$

$$= q_w r_w + q_x r_x + q_y r_y + q_z r_z \tag{51b}$$

$$= \mathbf{q} \cdot \mathbf{r} \tag{51c}$$

Multiplication Multiplication (also known as the Hamilton multiplication) of quaternions gives a quaternion:

$$q * r = \left(q_w + \hat{\mathbf{i}}q_x + \hat{\mathbf{j}}q_y + \hat{\mathbf{k}}q_z\right) * \left(r_w + \hat{\mathbf{i}}r_x + \hat{\mathbf{j}}r_y + \hat{\mathbf{k}}r_z\right) \tag{52a}$$

$$\begin{aligned}
= & q_w r_w + \hat{\mathbf{i}}q_w r_x + \hat{\mathbf{j}}q_w r_y + \hat{\mathbf{k}}q_w r_z + \\
& \hat{\mathbf{i}}q_x r_w - q_x r_x + \hat{\mathbf{k}}q_x r_y - \hat{\mathbf{j}}q_x r_z + \\
& \hat{\mathbf{j}}q_y r_w - \hat{\mathbf{k}}q_y r_x - q_y r_y + \hat{\mathbf{i}}q_y r_z + \\
& \hat{\mathbf{k}}q_z r_w + \hat{\mathbf{j}}q_z r_x - \hat{\mathbf{i}}q_z r_y - q_z r_z
\end{aligned} \tag{52b}$$

$$\begin{aligned}
= & \left(q_w r_w - q_x r_x - q_y r_y - q_z r_z\right) + \\
& \hat{\mathbf{i}}\left(q_w r_x + q_x r_w + q_y r_z - q_z r_y\right) + \\
& \hat{\mathbf{j}}\left(q_w r_y - q_x r_z + q_y r_w + q_z r_x\right) + \\
& \hat{\mathbf{k}}\left(q_w r_z + q_x r_y - q_y r_x + q_z r_w\right)
\end{aligned} \tag{52c}$$

which can be written using a matrix notation:

$$\mathbf{q} * \mathbf{r} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} * \begin{bmatrix} r_w \\ r_x \\ r_y \\ r_z \end{bmatrix} \tag{53a}$$

$$= \begin{bmatrix} q_w r_w - q_x r_x - q_y r_y - q_z r_z \\ q_w r_x + q_x r_w + q_y r_z - q_z r_y \\ q_w r_y - q_x r_z + q_y r_w + q_z r_x \\ q_w r_z + q_x r_y - q_y r_x + q_z r_w \end{bmatrix} \tag{53b}$$

$$= \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} \begin{bmatrix} r_w \\ r_x \\ r_y \\ r_z \end{bmatrix} \tag{53c}$$

$$= \begin{bmatrix} r_w & -r_x & -r_y & -r_z \\ r_x & r_w & r_z & -r_y \\ r_y & -r_z & r_w & r_x \\ r_z & r_y & -r_x & r_w \end{bmatrix} \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{53d}$$

By defining the *pre-multiplication* matrix:

$$Q(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} \tag{54}$$

and the *post-multiplication* matrix:

$$W(\mathbf{q}) = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \tag{55}$$

we have:

$$\mathbf{q} * \mathbf{r} = Q(\mathbf{q})\mathbf{r} = W(\mathbf{r})\mathbf{q} \tag{56}$$

Note that quaternion multiplication is generally non-commutative as $Q(\mathbf{q}) \neq W(\mathbf{q})$. However, this does not mean that for two quaternions q and r, $q * r$ is never equal to $r * q$. A counterexample is the multiplication of a quaternion q and its conjugate quaternion $\bar{q}$. According to equation (52), we have:

$$q * \bar{q} = \bar{q} * q = q_w^2 + q_x^2 + q_y^2 + q_z^2 \tag{57}$$

and in vector representation:

$$\mathbf{q} * \bar{\mathbf{q}} = \bar{\mathbf{q}} * \mathbf{q} = \begin{bmatrix} q_w^2 + q_x^2 + q_y^2 + q_z^2 \\ 0_{3\times 1} \end{bmatrix} \tag{58}$$

We can understand this counterexample as follows. Recall (49) that the non-commutativity of quaternion multiplication is introduced by imaginary parts of the quaternion (e.g. $\hat{\mathbf{i}}\hat{\mathbf{j}} = \hat{\mathbf{k}} \neq \hat{\mathbf{j}}\hat{\mathbf{i}} = -\hat{\mathbf{k}}$). While in (57) and (58), the coefficients of imaginary parts in the multiplication result is all zero. Since there is no imaginary part in the multiplication result, the non-commutativity does not apply.

One can verify the following properties of quaternion multiplication:

$$Q(\bar{\mathbf{q}}) = Q(\mathbf{q})^T \tag{59}$$

$$W(\bar{\mathbf{q}}) = W(\mathbf{q})^T \tag{60}$$

$$Q(\mathbf{q})^T\mathbf{q} = W(\mathbf{q})^T\mathbf{q} = (q * \bar{q})\,\mathbf{e} \tag{61}$$

$$Q(\mathbf{q})^T Q(\mathbf{q}) = W(\mathbf{q})^T W(\mathbf{q}) = (q * \bar{q})\, I \tag{62}$$

$$Q(\mathbf{q})Q(\mathbf{r}) = Q(Q(\mathbf{q})\mathbf{r}) = Q(\mathbf{q} * \mathbf{r}) \tag{63}$$

$$W(\mathbf{r})W(\mathbf{q}) = W(W(\mathbf{r})\mathbf{q}) = W(\mathbf{q} * \mathbf{r}) \tag{64}$$

where $\mathbf{e} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ is the vector representation of the identity quaternion, $I \in \mathbb{R}^{4\times 4}$ is the identity matrix.

Norm The norm of a quaternion is a scalar defined as the square root of its product with its conjugate:

$$||q|| = \sqrt{q * \bar{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \tag{65}$$

which gives the same result of the Euclidean norm of the quaternion's vector representation:

$$||\mathbf{q}|| = \sqrt{\mathbf{q} \cdot \mathbf{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \tag{66}$$

One can verify

$$q * \bar{q} = q \cdot q = ||q||^2 \tag{67}$$

$$||q * r||^2 = ||q||^2\, ||r||^2 = ||\mathbf{q} * \mathbf{r}||^2 = ||\mathbf{q}||^2\, ||\mathbf{r}||^2 \tag{68}$$

Inverse The inverse (also known as the reciprocal) of a quaternion q (assuming its norm is not zero) is a quaternion defined as:

$$q^{-1} = \frac{\bar{q}}{||q||^2} \tag{69}$$

Note that for a unit quaternion with norm 1, its inverse is equal to its conjugate.

One can verify (by dividing $||q||^2$ on both sides of equation (57))

$$q * q^{-1} = q^{-1} * q = 1 \tag{70}$$

Unit Quaternion and 3D Rotation

Let $q \in \mathbb{H}_1 = \{q \in \mathbb{H} : ||q|| = 1\}$ be a *unit quaternion*, we can find that q encodes a 3D rotation by calculating:

$$\begin{aligned}
W(\mathbf{q})^T Q(\mathbf{q}) &= \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}^T \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} && \text{(71a)} \\
&= \begin{bmatrix} q_w & q_x & q_y & q_z \\ -q_x & q_w & -q_z & q_y \\ -q_y & q_z & q_w & -q_x \\ -q_z & -q_y & q_x & q_w \end{bmatrix} \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} && \text{(71b)} \\
&= \begin{bmatrix} ||q||^2 & 0 & 0 & 0 \\ 0 & q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 0 & 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 0 & 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} && \text{(71c)} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 0 & 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 0 & 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} && \text{(71d)}
\end{aligned}$$

Let

$$R(\mathbf{q}) = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{72}$$

One can verify $R(\mathbf{q})$ is a rotation matrix as $R(\mathbf{q})^T R(\mathbf{q}) = I$ and $det\,(R(\mathbf{q})) = 1$.

Let $\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \in \mathbb{R}^3$ be a 3D vector. Using rotation matrix representation, the image of $\mathbf{v}$ by a 3D rotation transform is

$$\mathbf{v}' = R\mathbf{v} \tag{73}$$

where

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad R^T R = I \text{ and } det(R) = 1 \tag{74}$$

is a 3D rotation matrix.

According to equations (56), (71), and (72), the rotation transform (73) is equivalent to multiplication of quaternions:

$$\mathbf{v}_q' = \mathbf{q} * \mathbf{v}_q * \bar{\mathbf{q}} \tag{75}$$

where

$$\mathbf{v}_q = \begin{bmatrix} 0 & \frac{v_x}{||\mathbf{v}||^2} & \frac{v_y}{||\mathbf{v}||^2} & \frac{v_z}{||\mathbf{v}||^2} \end{bmatrix}^T \tag{76}$$

is the vector representation of a pure imaginary unit quaternion constructed from $\mathbf{v}$ and $\mathbf{q}$ is the vector representation of the unit quaternion representing the same rotation of R, by reversing (72), $\mathbf{q}$ can be obtained by the following formula:

$$\mathbf{q}(R) = \pm \begin{bmatrix} \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \\ sign(r_{23} - r_{32})\frac{1}{2}\sqrt{1 + r_{11} - r_{22} - r_{33}} \\ sign(r_{31} - r_{13})\frac{1}{2}\sqrt{1 - r_{11} + r_{22} - r_{33}} \\ sign(r_{12} - r_{21})\frac{1}{2}\sqrt{1 - r_{11} - r_{22} + r_{33}} \end{bmatrix} \tag{77}$$

Double coverage Equation (77) implies that $\mathbf{q}$ and $-\mathbf{q}$ encodes the same rotation, i.e., $R(\mathbf{q}) = R(-\mathbf{q})$. Some literature refers to this phenomenon as unit quaternion *double covering* three-dimensional rotations. In practice, we can use some conventions to determine the sign of a unit quaternion. For example, the following section will show that by taking q_w greater than 0, we assume that the rotation angle range of the rotation represented by the quaternion is $(0, \pi)$.

Geometric Interpretation

Let unit vector $\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T \in \mathbb{R}^3$ and $\theta \in (0, \pi)$ be the rotation axis and rotation angle for a rotation transform. According to the Rodrigues' rotation formula, the image of $\mathbf{v} \in \mathbb{R}^3$ by the 3D rotation transform is:

$$
\begin{aligned}
\mathbf{v}' &= \mathbf{v}\cos\theta + (\mathbf{u}\times\mathbf{v})\sin\theta + \mathbf{u}(\mathbf{v}\cdot\mathbf{u})(1-\cos\theta) && (78a)\\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\cos\theta + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\sin\theta + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}^T\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}(1-\cos\theta) && (78b)\\
&= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\cos\theta + \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}\sin\theta + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}(u_x v_x + u_y v_y + u_z v_z)(1-\cos\theta) && (78c)\\
&= \begin{bmatrix} v_x\cos\theta + (u_y v_z - u_z v_y)\sin\theta + u_x(u_x v_x + u_y v_y + u_z v_z)(1-\cos\theta) \\ v_y\cos\theta + (u_z v_x - u_x v_z)\sin\theta + u_y(u_x v_x + u_y v_y + u_z v_z)(1-\cos\theta) \\ v_z\cos\theta + (u_x v_y - u_y v_x)\sin\theta + u_z(u_x v_x + u_y v_y + u_z v_z)(1-\cos\theta) \end{bmatrix} && (78d)\\
&= \begin{bmatrix} u_x^2(1-\cos\theta)+\cos\theta & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_x u_y(1-\cos\theta) + u_z\sin\theta & u_y^2(1-\cos\theta)+\cos\theta & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_x u_z(1-\cos\theta) - u_y\sin\theta & u_y u_z(1-\cos\theta) + u_x\sin\theta & u_z^2(1-\cos\theta)+\cos\theta \end{bmatrix}\mathbf{v} && (78e)
\end{aligned}
$$

from which we can extract the rotation matrix in terms of rotation axis and rotation angle:

$$
R(\mathbf{u},\theta) = \begin{bmatrix} u_x^2(1-\cos\theta)+\cos\theta & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_x u_y(1-\cos\theta) + u_z\sin\theta & u_y^2(1-\cos\theta)+\cos\theta & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_x u_z(1-\cos\theta) - u_y\sin\theta & u_y u_z(1-\cos\theta) + u_x\sin\theta & u_z^2(1-\cos\theta)+\cos\theta \end{bmatrix} \tag{79}
$$

By comparing (74) and (79), we can reveal the the axis and angle of rotation corresponding to the rotation matrix R:

$$
\theta(R) = \arccos\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right) \tag{80}
$$

and:

$$
\mathbf{u}(R) = \frac{1}{2\sin\theta(R)}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix} \tag{81}
$$

where $\theta \in (0,\pi)$.

Using the same technique, we can reveal the axis and angle of rotation corresponding to the unit quaternion q. Plugging (72) into (80) yields:

$$
\begin{aligned}
\theta(\mathbf{q}) &= \arccos\left(\frac{3q_w^2 - q_x^2 - q_y^2 - q_z^2 - 1}{2}\right) && (82a)\\
&= \arccos\left(\frac{4q_w^2 - 2}{2}\right) && (82b)\\
&= \arccos(2q_w^2 - 1) && (82c)\\
&= 2\arccos(q_w) && (82d)
\end{aligned}
$$

Hint: from (82c) to (82d) we used the trigonometric half-angle identity for cosine:

$$\cos\left(\frac{\theta}{2}\right) = \sqrt{\frac{1+\cos\theta}{2}}, \quad \theta \in (0, \pi) \tag{83}$$

For rotation axis, plugging (72) into (81) yields:

$$\mathbf{u}(\mathbf{q}) = \frac{1}{2\sin\theta(\mathbf{q})}\begin{bmatrix}4q_wq_x\\4q_wq_y\\4q_wq_z\end{bmatrix} \tag{84a}$$

$$= \frac{2q_w}{\sin\theta(\mathbf{q})}\begin{bmatrix}q_x\\q_y\\q_z\end{bmatrix} \tag{84b}$$

$$= \frac{2\cos\left(\frac{\theta(\mathbf{q})}{2}\right)}{2\sin\left(\frac{\theta(\mathbf{q})}{2}\right)\cos\left(\frac{\theta(\mathbf{q})}{2}\right)}\begin{bmatrix}q_x\\q_y\\q_z\end{bmatrix} \tag{84c}$$

$$= \frac{1}{\sin\left(\frac{\theta(\mathbf{q})}{2}\right)}\begin{bmatrix}q_x\\q_y\\q_z\end{bmatrix} \tag{84d}$$

$$= \frac{1}{\sqrt{1-q_w^2}}\begin{bmatrix}q_x\\q_y\\q_z\end{bmatrix} \tag{84e}$$

$$= \frac{1}{\sqrt{q_x^2+q_y^2+q_z^2}}\begin{bmatrix}q_x\\q_y\\q_z\end{bmatrix} \tag{84f}$$

Similar to a rotation matrix, by reserving (82d) and (84d), we can represent a unit quaternion using its corresponding rotation axis and angle as:

$$q(\mathbf{u}, \theta) = \cos\left(\frac{\theta}{2}\right) + \hat{\mathbf{i}}u_x\sin\left(\frac{\theta}{2}\right) + \hat{\mathbf{j}}u_y\sin\left(\frac{\theta}{2}\right) + \hat{\mathbf{k}}u_z\sin\left(\frac{\theta}{2}\right) \tag{85}$$

or in vector representation:

$$\mathbf{q}(\mathbf{u}, \theta) = \begin{bmatrix}\cos\left(\frac{\theta}{2}\right)\\ \sin\left(\frac{\theta}{2}\right)\mathbf{u}\end{bmatrix} \tag{86}$$

With this axis-angle representation, we can interpret algebra of unit quaternions from a geometric perspective:

Inverse According to (69) and (86), the inverse of a unit quaternion is a unit quaternion that represents a rotation of the same angle and opposite direction as the original quaternion (i.e., an inverse rotation):

$$\mathbf{q}^{-1}(\mathbf{u}, \theta) = \begin{bmatrix}\cos\left(\frac{\theta}{2}\right)\\ \sin\left(\frac{\theta}{2}\right)(-\mathbf{u})\end{bmatrix} \tag{87}$$

Multiplication According to (53b) and (86), the product of two unit quaternions is a quaternion that represents the superposition of the rotations represented by the two original quaternions:

$$\mathbf{q}_2(\mathbf{u}_2,\theta_2) * \mathbf{q}_1(\mathbf{u}_1,\theta_1)$$

$$= \begin{bmatrix} \cos\left(\frac{\theta_2}{2}\right) \\ \sin\left(\frac{\theta_2}{2}\right)\mathbf{u}_2 \end{bmatrix} * \begin{bmatrix} \cos\left(\frac{\theta_1}{2}\right) \\ \sin\left(\frac{\theta_1}{2}\right)\mathbf{u}_1 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta_2}{2}\right) \\ u_{2x}\sin\left(\frac{\theta_2}{2}\right) \\ u_{2y}\sin\left(\frac{\theta_2}{2}\right) \\ u_{2z}\sin\left(\frac{\theta_2}{2}\right) \end{bmatrix} * \begin{bmatrix} \cos\left(\frac{\theta_1}{2}\right) \\ u_{1x}\sin\left(\frac{\theta_1}{2}\right) \\ u_{1y}\sin\left(\frac{\theta_1}{2}\right) \\ u_{1z}\sin\left(\frac{\theta_1}{2}\right) \end{bmatrix} \tag{88a}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1}{2}\right) - (u_{2x}u_{1x} + u_{2y}u_{1y} + u_{2y}u_{1y})\sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) \\ u_{1x}\cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) + u_{2x}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) + (u_{2y}u_{1x} - u_{2z}u_{1y})\sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) \\ u_{1y}\cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) + u_{2y}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) + (u_{2z}u_{1x} - u_{2z}u_{1z})\sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) \\ u_{1z}\cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) + u_{2z}\cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right) + (u_{2x}u_{1y} - u_{2y}u_{1x})\sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) \end{bmatrix} \tag{88b}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1}{2}\right) - (\mathbf{u}_2\cdot\mathbf{u}_1)\sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right) \\ \cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)\mathbf{u}_1 + \cos\left(\frac{\theta_1}{2}\right)\sin\left(\frac{\theta_2}{2}\right)\mathbf{u}_2 + \sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1}{2}\right)(\mathbf{u}_2\times\mathbf{u}_1) \end{bmatrix} \tag{88c}$$

In particular, we have

$$\mathbf{q}^2(\mathbf{u},\theta) = \mathbf{q}(\mathbf{u},\theta) * \mathbf{q}(\mathbf{u},\theta) \tag{89a}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\mathbf{u} \end{bmatrix} * \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\mathbf{u} \end{bmatrix} \tag{89b}$$

$$= \begin{bmatrix} \cos(\theta) \\ \sin(\theta)\,\mathbf{u} \end{bmatrix} \tag{89c}$$

$$= \mathbf{q}(\mathbf{u}, 2\theta) \tag{89d}$$

We can extend (89d) to

$$\mathbf{q}^n(\mathbf{u},\theta) = \mathbf{q}(\mathbf{u}, n\theta), \quad n \in \mathbb{N} \tag{90}$$

By reversing (90) we can define

$$\mathbf{q}^{\frac{1}{n}}(\mathbf{u},\theta) = \mathbf{q}(\mathbf{u}, \frac{\theta}{n}) \quad n \in \mathbb{N} \tag{91}$$

Combing (90) and (91) yields

$$\mathbf{q}^t(\mathbf{u},\theta) = \mathbf{q}^{\frac{m}{n}}(\mathbf{u},\theta) \tag{92a}$$

$$= \mathbf{q}^m(\mathbf{u},\theta) * \mathbf{q}^{\frac{1}{n}}(\mathbf{u},\theta) \tag{92b}$$

$$= \mathbf{q}(\mathbf{u}, m\theta) * \mathbf{q}(\mathbf{u}, \frac{\theta}{n}) \tag{92c}$$

$$= \mathbf{q}(\mathbf{u}, \frac{m}{n}\theta) \tag{92d}$$

$$= \mathbf{q}(\mathbf{u}, t\theta) \tag{92e}$$

where $m, n \in \mathbb{N}$ and $t = \frac{m}{n} \in \mathbb{R}$. Equation (92e) implies that we can scale and interpolate the rotation angle without changing the rotation axis by raising the quaternion to a power. It also leads to the spherical linear interpolation (slerp) between two quaternions.

Slerp Let unit quaternions q_1 and q_2 represent the starting and ending points of the interpolation respectively. Let $t \in [0, 1]$ be the interpolation coefficient, we have:

$$q_2 = (\triangle q) * q_1 \tag{93a}$$
$$\Rightarrow \triangle q = q_2 * q_1^{-1} \tag{93b}$$

and:

$$slerp(q_1, q_2, t) = (\triangle q)^t * q_1 \tag{94a}$$
$$= \left(q_2 * q_1^{-1}\right)^t * q_1 \tag{94b}$$

Note that when $slerp(q_1, q_2, t) = q_1$ if $t = 0$, and $slerp(q_1, q_2, t) = q_2$ if $t = 1$. When $0 < t < 1$, for example, if $t = 0.5$, $slerp(q_1, q_2, t)$ is a quaternion representing a 50% rotation from q_1 to q_2.

Graßmann product Although beyond the scope of this documentation, equation (88c) implies the Graßmann product. Interested readers may consult the literature on exterior algebra for more information. It is recommended to compare the Graßmann product with the Rodrigue's rotation formula (78a), which helps to understand the relationship between quaternions and rotations.

END OF THIS DOCUMENTATION