SLAM Geometry Notes

Sami Osborn

1 Coordinate Frames

A coordinate frame is the combination of a defined origin point and a set of three orthogonal unit axes directions.

World / Global Frame (W)

The world frame (or global frame) is fixed and is generally where the object starts.

Camera / Body Frame (C/B)

The camera frame (or body frame) is where the camera is in space. It moves with the robot.

Landmark Frame (L)

This is used for local feature points.

2 Transformations

We define T_{AB} as the transformation from frame B to A. A pose is a pair (R, t) describing the orientation and position of one frame relative to another.

Definition of a Lie Group

Formally, a Lie group G is a set that satisfies:

1. **Group:** There exists a binary operation (group multiplication) $\circ : G \times G \to G$ such that:
   - **Closure:** $a \circ b \in G$
   - **Identity element:** $e \in G$ such that $e \circ g = g \circ e = g$
   - **Inverse element:** for all $g \in G$, there exists $g^{-1} \in G$
   - **Associativity:** for all $a, b, c \in G$, $(a \circ b) \circ c = a \circ (b \circ c)$
2. **Smooth manifold structure:** The set G is also a differentiable manifold.
3. **Smooth group operations:** The group operations are smooth (infinitely differentiable) maps:

$$\begin{aligned} \text{Multiplication:} \quad & G \times G \to G, \quad (g_1, g_2) \mapsto g_1 \circ g_2 \\ \text{Inverse:} \quad & G \to G, \quad g \mapsto g^{-1} \end{aligned}$$

Switching Coordinate Frames

To move between frames, we first translate the origin, then rotate the frame. Consider moving from world to camera frame:

$$p_C = R_{CW}(p_W - t_{CW})$$

$$p_W = R_{WC}p_C + t_{CW}$$

Where:

- $R_{WC} = R_{CW}^T$

Rotation Groups – SO(3)

The Special Orthogonal Group of degree 3, $SO(3)$, is the group of rotations in 3D.

Rotation Matrix, R

A rotation matrix $R \in \mathbb{R}^{3\times3}$ satisfies:

$$RR^T = I, \quad \det(R) = 1$$

Rotation Axis–Angle Representation

Any 3D rotation can be expressed as a rotation by an angle θ about a unit vector (axis) $\hat{u} \in \mathbb{R}^3$:

$$(\hat{u}, \theta) \quad \text{with } \|\hat{u}\| = 1$$

Axis–Angle to a Rotation Matrix

The corresponding rotation matrix is obtained using the Rodrigues formula:

$$R(\hat{u}, \theta) = I + \sin\theta\, \hat{u}^\wedge + (1 - \cos\theta)(\hat{u}^\wedge)^2$$

Where the hat operator $(\cdot)^\wedge$ maps a 3D vector to its skew-symmetric matrix:

$$\hat{u}^\wedge = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

Applying to a vector: Let $\mathbf{v} \in \mathbb{R}^3$

$$\mathbf{v}' = R(\hat{u}, \theta)\mathbf{v}$$

Expanded as Rodrigues' formula:

$$\mathbf{v}' = \mathbf{v}\cos(\theta) + (\hat{u} \times \mathbf{v})\sin(\theta) + \hat{u}(\hat{u} \cdot \mathbf{v})(1 - \cos(\theta))$$

Extracting Axis–Angle from a Rotation Matrix

Given a rotation matrix $R \in SO(3)$:

$$\theta = \cos^{-1}\left(\frac{\text{trace}(R) - 1}{2}\right)$$

If $\theta \neq 0$, the rotation axis $\hat{u}$ can be recovered as:

$$\hat{u} = \frac{1}{2\sin\theta}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$$

Quaternion Vector

A quaternion vector is:

$$q = q_w + q_x i + q_y j + q_z k$$

Where:

- q_w is the scalar component.
- $\mathbf{q_v}$ is the vector component.

Pure quaternion: $\mathbf{v} = [0, \mathbf{v}]$

Multiplication: Consider two quaternions, $[s_1, \mathbf{a}], [s_2, \mathbf{b}]$:

$$[s_1, \mathbf{a}][s_2, \mathbf{b}] = [s_1 s_2 - a \cdot b, \quad s_1\mathbf{b} + s_2\mathbf{a} + \mathbf{a} \times \mathbf{b}]$$

Rotation Quaternion

A unit quaternion $q = [q_w, q_x, q_y, q_z]^T \in \mathbb{R}^4$ represents the same rotation, where:

$$q_w = \cos\left(\frac{\theta}{2}\right), \quad \mathbf{q}_v = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \hat{u}\sin\left(\frac{\theta}{2}\right)$$

Thus:

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \hat{u}_x \sin\left(\frac{\theta}{2}\right) \\ \hat{u}_y \sin\left(\frac{\theta}{2}\right) \\ \hat{u}_z \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Inversely:

$$\mathbf{q^{-1}} = \mathbf{q}* = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ -\hat{u}_x \sin\left(\frac{\theta}{2}\right) \\ -\hat{u}_y \sin\left(\frac{\theta}{2}\right) \\ -\hat{u}_z \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Where u is the axis of rotation, and θ is the rotation angle. It's more computationally stable and efficient than Euler angles or rotation matrices.

Applying a Rotation Quaternion

Let $\mathbf{v}$ be a pure quaternion - i.e. a point in 3D.

To rotate v by the rotation quaternion q we perform:

$$v' = qvq^{-1}$$

Quaternion to Axis–Angle Conversion

Given a unit quaternion $q = [q_w, q_x, q_y, q_z]^T$:

$$\theta = 2\cos^{-1}(q_w), \quad \hat{u} = \frac{1}{\sin(\theta/2)} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \quad \text{for } \sin\left(\tfrac{\theta}{2}\right) \neq 0$$

Quaternion to Rotation Matrix

The rotation matrix corresponding to q is:

$$R(q) = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_z q_w) & 2(q_x q_z + q_y q_w) \\ 2(q_x q_y + q_z q_w) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_x q_w) \\ 2(q_x q_z - q_y q_w) & 2(q_y q_z + q_x q_w) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix}$$

Rotation Matrix to Quaternion

Given a rotation matrix,

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \in SO(3)$$

The equivalent quaternion $q = [q_w, q_x, q_y, q_z]^T$ can be extracted from the matrix elements.

Trace-based formula: For trace ¿ 0 set: $t = R_{11} + R_{22} + R_{33}$

Then:

$$\begin{cases} q_w = \frac{1}{2}\sqrt{1+t} \\ q_x = \frac{1}{4q_w}(R_{32} - R_{23}) \\ q_y = \frac{1}{4q_w}(R_{13} - R_{31}) \\ q_z = \frac{1}{4q_w}(R_{21} - R_{12}) \end{cases} \quad \text{if } t > 0$$

This formula works well when the trace of R is positive (small rotation angles).

Numerically stable version: For rotations near 180°, the trace t may become small or negative.

To avoid numerical instability, choose the largest diagonal element of R and compute accordingly:

$$\text{If } R_{11} \text{ is largest:} \quad \begin{cases} q_x = \frac{1}{2}\sqrt{1 + R_{11} - R_{22} - R_{33}} \\ q_y = \frac{1}{4q_x}(R_{12} + R_{21}) \\ q_z = \frac{1}{4q_x}(R_{13} + R_{31}) \\ q_w = \frac{1}{4q_x}(R_{32} - R_{23}) \end{cases}$$

$$\text{If } R_{22} \text{ is largest:} \quad \begin{cases} q_y = \frac{1}{2}\sqrt{1 - R_{11} + R_{22} - R_{33}} \\ q_x = \frac{1}{4q_y}(R_{12} + R_{21}) \\ q_z = \frac{1}{4q_y}(R_{23} + R_{32}) \\ q_w = \frac{1}{4q_y}(R_{13} - R_{31}) \end{cases}$$

$$\text{If } R_{33} \text{ is largest:} \quad \begin{cases} q_z = \frac{1}{2}\sqrt{1 - R_{11} - R_{22} + R_{33}} \\ q_x = \frac{1}{4q_z}(R_{13} + R_{31}) \\ q_y = \frac{1}{4q_z}(R_{23} + R_{32}) \\ q_w = \frac{1}{4q_z}(R_{21} - R_{12}) \end{cases}$$

Normalisation: After computing, normalise to ensure the quaternion lies on the unit 4-sphere:

$$q \leftarrow \frac{q}{\|q\|}$$

Homogeneous Coordinates and the Special Euclidean Group $SE(3)$

The Special Euclidean group $SE(3)$ represents all rigid-body transformations (rotations and translations) in 3D space:

$$T_{WC} = \begin{bmatrix} R_{WC} & t_{WC} \\ 0 & 1 \end{bmatrix}, \quad R_{WC} \in SO(3),\ t_{WC} \in \mathbb{R}^3$$

A 3D point expressed in homogeneous coordinates is written as:

$$\tilde{p}_W = \begin{bmatrix} p_W \\ 1 \end{bmatrix}, \qquad \tilde{p}_C = T_{CW}\, \tilde{p}_W$$

World to Camera Transformation

At time step i, the camera pose in the world frame is $T_{WC_i} = (R_{WC_i}, t_{WC_i})$. Let p_{W_j} denote the j-th landmark (fixed in the world frame). The coordinates of this landmark in the camera frame are:

$$p_{C_j} = R_{CW_i}(p_{W_j} - t_{CW_i})$$

The homogeneous transformation matrix from world to camera is therefore:

$$T_{CW_i} = \begin{bmatrix} R_{CW_i} & t_{WC_i} \\ 0 & 1 \end{bmatrix} \in SE(3)$$

Where: $t_{WC_i} = -R_{CW_i} t_{CW_i}$

Its inverse gives the camera-to-world transformation:

$$T_{WC_i} = T_{CW_i}^{-1} = \begin{bmatrix} R_{WC_i} & t_{CW_i} \\ 0 & 1 \end{bmatrix}$$

Where: $R_{WC_i} = R_{CW_i}^T$

Thus, a point in the camera frame can be mapped back to world coordinates as:

$$p_{W_j} = R_{CW_i}^T p_{C_j} + t_{CW_i}$$

Algebras and Lie Algebras

Definition of an Algebra

An algebra is a vector space equipped with a multiplication operation combining two elements of the space to produce another element of the same space.

Formally, an algebra $\mathcal{A}$ over a field satisfies:

1. $\mathcal{A}$ is a vector space over the field.
2. There exists a bilinear product $\cdot : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$.
3. For all $a, b, c \in \mathcal{A}$ and $\alpha, \beta \in \mathbb{R}$:

$$(\alpha a + \beta b) \cdot c = \alpha(a \cdot c) + \beta(b \cdot c)$$
$$a \cdot (\alpha b + \beta c) = \alpha(a \cdot b) + \beta(a \cdot c)$$

Definition of a Lie Algebra

A Lie algebra is an algebra with a Lie bracket operation:

$$[a, b] = ab - ba$$

That is: i) bilinear, ii) anti-symmetric, and iii) satisfies the Jacobi identity:

$$[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0$$

Examples:

- SO(3) $\leftrightarrow$ so(3): 3×3 skew-symmetric matrices.
- SE(3) $\leftrightarrow$ se(3): 4×4 matrices combining rotation and translation.

SE(3) Lie Algebra and the Hat Operator

A small pose update can be expressed as a 6D vector:

$$\xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6$$

Where: $\rho \in \mathbb{R}^3$ (translation), $\phi \in \mathbb{R}^3$ (rotation)

Hat Operator $(\cdot)^\wedge$

For rotations:

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}$$

For rigid-body motion:

$$\xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0 & 0 \end{bmatrix}$$

Exponential Map

The exponential map links the Lie algebra (tangent space) to the Lie group (manifold):

$$T = \exp\bigl(\xi^\wedge\bigr) = \begin{bmatrix} \exp(\phi^\wedge) & J\rho \\ 0 & 1 \end{bmatrix}$$

For small angles $\|\phi\| \approx 0$:

$$R \approx I + \phi^\wedge, \quad J \approx I + \frac{1}{2}\phi^\wedge$$

For finite angles:

$$R = I + \frac{\sin\theta}{\theta}\phi^\wedge + \frac{1-\cos\theta}{\theta^2}(\phi^\wedge)^2$$

$$J = I + \frac{1-\cos\theta}{\theta^2}\phi^\wedge + \frac{\theta-\sin\theta}{\theta^3}(\phi^\wedge)^2$$

Where: $\theta = \|\phi\|$