SLAM Geometry Notes

Sami Osborn

1 Coordinate Frames

A coordinate frame consists of an origin and a set of three orthogonal unit axes.

World / Global Frame (W)

The world frame (or global frame) is fixed and generally serves as the reference frame.

Camera / Body Frame (C/B)

The camera or body frame moves with the robot and expresses points relative to the robot's position.

Landmark Frame (L)

A local frame used for describing static scene landmarks.

2 Transformations and Lie Groups

Rigid Transformations

We denote T_{AB} as the transformation from frame B to A. A pose is a pair (R, t), where R is the rotation and t is the translation.

Switching between world and camera coordinates:

$$p_C = R_{CW}(p_W - t_{CW}), \qquad p_W = R_{WC}p_C + t_{CW}, \qquad R_{WC} = R_{CW}^T$$

Definition of a Lie Group

A Lie group G is both a group and a smooth manifold with smooth group operations:

$$(g_1, g_2) \mapsto g_1 g_2, \qquad g \mapsto g^{-1}$$

Definition of a Lie Algebra

A Lie algebra is a vector space equipped with a bilinear, antisymmetric bracket $[a, b] = ab - ba$ satisfying the Jacobi identity:

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

Examples:

$$SO(3) \leftrightarrow \mathfrak{so}(3), \qquad SE(3) \leftrightarrow \mathfrak{se}(3)$$

The lowercase symbols denote the Lie algebras (tangent spaces at the identity), while uppercase symbols denote the Lie groups (manifolds of transformations).

3 The Special Orthogonal Group $SO(3)$

Definition

$$SO(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I,\ \det(R) = 1\}$$

It represents all 3D rotations. Each $R \in SO(3)$ preserves lengths and angles.

Associated Lie Algebra $\mathfrak{so}(3)$

The Lie algebra of $SO(3)$ is

$$\mathfrak{so}(3) = \{A \in \mathbb{R}^{3\times 3} \mid A^T = -A\}$$

Every element of $\mathfrak{so}(3)$ corresponds to a vector $\phi = (\phi_x, \phi_y, \phi_z)^T \in \mathbb{R}^3$ via the **hat operator**:

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}, \qquad (\phi^\wedge)^\vee = \phi$$

The hat operator maps a vector to its corresponding skew-symmetric matrix such that $\phi^\wedge v = \phi \times v$.

Exponential Map on $\mathfrak{so}(3)$

The exponential map links infinitesimal rotations in the Lie algebra $\mathfrak{so}(3)$ to finite rotations in the Lie group $SO(3)$. For a rotation vector $\phi \in \mathbb{R}^3$ with skew-symmetric form $\phi^\wedge \in \mathfrak{so}(3)$,

$$\exp : \mathfrak{so}(3) \to SO(3), \qquad R(\phi) = \exp\big(\phi^\wedge\big)$$

This expression arises by integrating the differential equation:

$$\dot{R}(t) = \phi^\wedge R(t), \qquad R(0) = I$$

whose unique solution is $R(t) = \exp(t\,\phi^\wedge)$. The trajectory $R(t)$ forms a smooth one-parameter subgroup of $SO(3)$, representing continuous rotation with constant angular velocity ϕ. Evaluating at $t = 1$ gives the finite rotation $R(\phi)$, corresponding to a rotation by angle $\|\phi\|$ about the unit axis $\hat{u} = \phi/\|\phi\|$.

Rodrigues' Formula

Let $\theta = \|\phi\|$ and $\hat{u} = \phi/\theta$. The exponential of a skew-symmetric matrix admits a closed form:

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

which is known as Rodrigues' rotation formula. Expanding $\exp(\phi^\wedge)$ via the Taylor series and using $(\hat{u}^\wedge)^3 = -\hat{u}^\wedge$ gives the above closed form.

Axis–Angle Representation

A rotation can equivalently be described by an angle θ about a unit axis $\hat{u}$:

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

and inversely,

$$\theta = \cos^{-1}\left(\frac{\mathrm{trace}(R)-1}{2}\right), \quad \hat{u} = \frac{1}{2\sin\theta}\begin{bmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{bmatrix}$$

4 Quaternion Representation of Rotation

A unit quaternion $q = [q_w, q_x, q_y, q_z]^T$ encodes rotation as:

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

Quaternion conjugate and inverse:

$$q^* = [q_w, -q_x, -q_y, -q_z]^T, \qquad q^{-1} = q^*$$

Applying a Quaternion Rotation

Represent a vector v as a pure quaternion $[0, \mathbf{v}]$. The rotated vector is given by:

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

Quaternion to Matrix and Back

Rotation matrix from quaternion:

$$R(q) = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y - q_zq_w) & 2(q_xq_z + q_yq_w) \\ 2(q_xq_y + q_zq_w) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_xq_w) \\ 2(q_xq_z - q_yq_w) & 2(q_yq_z + q_xq_w) & 1-2(q_x^2+q_y^2) \end{bmatrix}$$

To recover a quaternion from R:

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

5 The Special Euclidean Group $SE(3)$

Definition

The group of rigid-body motions in 3D:

$$SE(3) = \left\{ \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \middle| R \in SO(3),\ t \in \mathbb{R}^3 \right\}$$

Homogeneous Coordinates

A point p_W in homogeneous form:

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

Inverse transform:

$$T_{WC} = T_{CW}^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$$

6 The Lie Algebra $\mathfrak{se}(3)$ and Exponential Map

Elements and Hat Operator

An element of $\mathfrak{se}(3)$ is a 6D twist:

$$\xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix}, \quad \rho, \phi \in \mathbb{R}^3$$

with hat representation:

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

Exponential Map $\exp : \mathfrak{se}(3) \to SE(3)$

$$T(\xi) = \exp\big(\xi^\wedge\big) = \begin{bmatrix} R(\phi) & J(\phi)\rho \\ 0 & 1 \end{bmatrix}, \qquad R(\phi) = \exp\big(\phi^\wedge\big)$$

where $J(\phi)$ is the **left Jacobian** of $SO(3)$.

Jacobian Definition

The left Jacobian of $SO(3)$, denoted $J(\phi)$, relates small perturbations in the rotation vector space $\mathbb{R}^3$ to displacements on the manifold. It is defined by the infinite series expansion

$$J(\phi) = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\phi^\wedge)^n$$

which has a closed-form expression that depends on the rotation angle $\theta = \|\phi\|$:

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

This exact form is valid for any finite rotation and follows from the Taylor expansion of the exponential map on $SO(3)$.

When the rotation vector ϕ is small ($\|\phi\| \to 0$), higher-order terms in the series can be neglected. Using the small-angle approximations $\sin\theta \approx \theta$ and $\cos\theta \approx 1 - \frac{\theta^2}{2}$, the Jacobian reduces to a first-order form

$$J(\phi) \approx I + \tfrac{1}{2}\phi^\wedge$$

which is often used in linearised pose updates such as those in EKF-SLAM and bundle adjustment.

Pose Updates on $SE(3)$

Small motion increments in 3D space are represented in the Lie algebra $\mathfrak{se}(3)$ and applied to a current pose $T \in SE(3)$ via the exponential map.

Given a perturbation $\xi = [\rho^T,\, \phi^T]^T \in \mathbb{R}^6$, the pose update is written as:

$$T \leftarrow \exp\big(\xi^\wedge\big)\, T$$

This update composes a small transformation on the manifold while preserving the $SE(3)$ group structure.

The rotational and translational components evolve as:

$$R \leftarrow \exp\big(\phi^\wedge\big)R, \qquad t \leftarrow \exp\big(\phi^\wedge\big)t + J(\phi)\rho$$

For small angles ($\theta \to 0$), the first-order approximations hold:

$$R \approx (I + \phi^\wedge)R, \qquad t \approx t + \rho + \tfrac{1}{2}\phi^\wedge\rho$$

or equivalently,

$$\exp\big(\xi^\wedge\big) \approx \begin{bmatrix} I + \phi^\wedge & \rho + \frac{1}{2}\phi^\wedge\rho \\ 0 & 1 \end{bmatrix}$$

7 Pinhole Camera Projection Model

The pinhole camera model describes how a 3D point in the camera frame projects onto the 2D image plane. It idealises the imaging process as light rays passing through a single optical centre before intersecting the image plane, so that the geometry of projection is fully determined by the camera intrinsics.

Image Frame Coordinate System

In computer vision, the image coordinate system is defined with the origin at the top-left corner of the image. The horizontal u-axis increases rightwards, and the v-axis vertical axis increases downwards. Similar to the convention in Numpy.

From 3D to 2D Pixels

A point in the camera coordinate frame is expressed as

$$p_C = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

In the pinhole model, this 3D point is projected onto the image plane through perspective division:

$$x = \frac{X_c}{Z_c}, \qquad y = \frac{Y_c}{Z_c}$$

where (x, y) are the normalised image coordinates in the plane $Z = 1$.

These coordinates represent where image plane intersects the line of sight from the camera centre through the point p_C.

The effect of division by Z_c captures the inverse-depth scaling of perspective: as the depth increases, the image location moves closer to the principal point on the image frame, and features appear smaller.

The mapping from the normalised image plane to the discrete pixel grid is defined by the camera intrinsic matrix K:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Here:

- f_x, f_y are the focal lengths in pixel units. They represent the distance between the optical plane and the image plane.
- (c_x, c_y) is the principal point, the pixel coordinate on the image plane, which intersects with the optical axis (the line perpendicular to the center of the optical plane).

Typically, the focal length f_{mm} is measured in millimetres, however we scale to pixel units:

$$f_x = \frac{f_{\text{mm}}}{s_x}, \qquad f_y = \frac{f_{\text{mm}}}{s_y}$$

where s_x, s_y are the pixel sizes along each sensor axis (mm/pixel). Thus, f_x, f_y describe how many pixels correspond to the optical focal distance. Larger focal lengths correspond to narrower fields of view (zoomed-in views), while smaller focal lengths capture a wider angular extent of the scene.

The full projection from camera coordinates to pixel coordinates is therefore:

$$\tilde{z} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x X_c + c_x Z_c \\ f_y Y_c + c_y Z_c \\ Z_c \end{bmatrix}$$

since,

$$\frac{u - c_x}{f_x} = \frac{X_c}{Z_c}, \qquad \frac{v - c_y}{f_y} = \frac{Y_c}{Z_c}$$

or equivalently,

$$u = f_x \frac{X_c}{Z_c} + c_x, \qquad v = f_y \frac{Y_c}{Z_c} + c_y$$

This mapping defines the measurement function:

$$h(x) = \begin{bmatrix} u \\ v \end{bmatrix}$$

which links a 3D landmark position in the camera frame, to its observed pixel coordinates on the image frame. Here x denotes the SLAM state vector, which contains the landmark(s) and camera pose parameters.

Field of View

The depth Z_c directly controls the strength of the perspective effect:

- As Z_c increases, the image motion becomes small, making distant landmarks more stable but less detailed.
- As $Z_c \to 0$, points approach the optical centre and the projection becomes singular ($1/Z_c$ diverges), so such close points cannot be imaged properly.

For numerical stability, a lower bound $Z_c > \epsilon$ (e.g. a few centimetres) is enforced during optimisation or filtering.

The focal lengths f_x, f_y not only scale the projection but also determine the camera's angular field of view (FOV):

$$\mathrm{FOV}_x = 2\tan^{-1}\!\left(\frac{w}{2f_x}\right), \qquad \mathrm{FOV}_y = 2\tan^{-1}\!\left(\frac{h}{2f_y}\right)$$

where w, h are the image width and height in pixels.

A short focal length yields a wide field of view (such as in wide-angle lenses), while a large focal length yields a narrow FOV (telephoto lens). Hence focal length and FOV are inversely related. Together with the principal point (c_x, c_y), these parameters fully define how the continuous 3D camera rays map to discrete image pixels.