The Derivation of Jacobian Matrix for the Point-Line-Plane Reprojection Factor

Shi-Sheng Huang

Email: shishenghuang.net@gmail.com

Abstract

In this material, we give a detailed derivation of the Jacobian matrix for the point-, line-, and plane- reprojection factor respectively. The point reprojection factor is straight forward, while the line reprojection factor and plane reprojection factor is somehow complicated due to the minimum parameters of lines and planes respectively. We adopt the line reprojection error following [1], [2], and the plane reprojection error following [3]. Using point-line-plane structure primitives as landmark during the motion tracking, we can lead to a more accurate and robust SLAM system in both the frame-to-frame tracking (front-end) and factor graph based bundle adjustment (back-end).

I. PRELIMINARIES

Special Orthogonal Group, SO(3) An rotation in $3D$ space can be represented as an orthogonal matrix $R \in R^{3\times3}$. Those orthogonal matrixes coincide with the general matrix multiplication and transposition form the Special Orthogonal Group, i.e. $SO(3) \doteq \{R \in R^{3\times3} : RR^T = I, det(R) = 1\}$. Given a rotation matrix $R \in SO(3)$, there is a skew matrix $S \in so(3)$ corresponding to R using the *exponential map* and *log map*:

$$\begin{aligned} exp(S) = R & \qquad \in SO(3) \\ log(R) = S & \qquad \in so(3) \end{aligned}$$

where $so(3)$ is the **Lie Algebra** of manifold $SO(3)$ [4]. We can identify every skew matrix with a vector $\omega \in R^3$ using the *hat* operator:

$$\omega^\wedge = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \in R^3$$

and on the vice verse the *vee* operator $\omega = S^\vee$ if $S = \omega^\wedge \in so(3)$. Therefore we directly build the *exponential map* and *log map* between a vector $\omega \in R^3$ and rotation matrix $R \in SO(3)$ in the following:

$$Exp(\omega) = exp(\omega^\wedge) = R \qquad \in SO(3)$$

$$Log(R) = log(R)^\vee = \omega \qquad \in R^3$$

and use a vector $\omega \in R^3$ as the minimal representation of a rotation matrix $R \in SO(3)$. Actually, such vector ω is the rotation vector of the rotation matrix R following the Rodrigues Fomula:

$$Exp(\omega) = exp(\omega^\wedge) = I + sin\theta(\hat{\omega})^\wedge + (1 - cos\theta)(\hat{\omega}\hat{\omega}^T - I)$$

with $\theta = |\omega|$ and $\hat{\omega} = \frac{\omega}{|\omega|}$.

According the the **BCH formula** [4], we have

$$Log(Exp(\omega_1)Exp(\omega_2)) \approx \begin{cases} J_l(\omega_2)^{-1}\omega_1 + \omega_2 & |\omega_1| < \epsilon \\ J_r(\omega_1)^{-1}\omega_2 + \omega_1 & |\omega_2| < \epsilon \end{cases} \tag{1}$$

with $J_l(\omega)$ and $J_r(\omega)$ is the Left Jacobian and Right Jacobian of rotation matrix separately, with $J_l(\omega) = Exp(\omega)J_r(\omega)$ [4].

Quaternion A rotation matrix $R \in SO(3)$ can also be represented as a unit quaternion $q \in R^4$ with $|q| = 1$. Since the DoF of a unit quaternion q is 3, we denote a quaternion set $S^3 \doteq \{q \in R^4 : |q| = 1\}$ to represent all the unit quaternions. A multiplication operator $\oplus$ [4] is defined as

$$q \oplus q' = \begin{pmatrix} q'_4 I_3 - \mathbf{q'}^{\wedge} & \mathbf{q'} \\ -\mathbf{q'}^T & q'_4 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ q_4 \end{pmatrix}$$

$$q = \begin{pmatrix} \mathbf{q} \\ q_4 \end{pmatrix} \qquad q' = \begin{pmatrix} \mathbf{q'} \\ q'_4 \end{pmatrix}$$

Therefore iversion q^{-1} of a quaternion q is just

$$q^{-1} = \begin{pmatrix} -\mathbf{q} \\ q_4 \end{pmatrix} \qquad q = \begin{pmatrix} \mathbf{q} \\ q_4 \end{pmatrix}$$

Similarly the *exponential map* and *log map* between a vector $\omega \in R^3$ and a quaternion $q \in S^3$ [4] is defined as

$$exp(\omega) = \begin{pmatrix} \frac{1}{2} sinc(\frac{\theta}{2})\hat{\omega} \\ cos(\frac{\theta}{2}) \end{pmatrix} \in S^3 \quad \theta = |\omega| \quad \hat{\omega} = \frac{\omega}{|\omega|}$$

$$log(q) = \frac{2 * cos^{-1}(q_4)}{|\mathbf{q}|}\mathbf{q} \in R^3$$

$$q = (\mathbf{q} \qquad q_4)^T \in S^3$$

Given a vector $\omega \in R^3$, there is a rotation matrix $R_\omega \in SO(3)$ such that $Exp(\omega) = R_\omega$ and there is also a quaternion $q_\omega \in S^3$ such that $exp(\omega) = q_\omega$. Therefore we denote $R(q_\omega) = R_\omega$ to define the mapping between quaternion q_ω and rotation matrix R_ω which correponding to the same vector ω, and obviously get

$$log(q_\omega) = Log(R(q_\omega))$$

which means the minimal parameterization of a quaterion $q \in S^3$ is a vector $\omega \in R^3$ as manifold $SO(3)$ does.

Special Euclidian Group, SE(3) Similarly, a transformation T in $3D$ space consists with a rotation matrix $R \in SO(3)$ and a translation vector $t \in R^3$, i.e. $T = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right)_{4\times 4}$. Those transformations coincide with general matrix multiplication and inversion form the Special Euclidian Group $SE(3) \doteq \{T = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right)_{4\times 4} : R \in$

$SO(3), t \in R^3\}$. Since the DoF of a transformation T is 6, we can also define the *hat* operator for a vector $\xi = \{\omega, \rho\}^T \in R^6$ as

$$\xi^\wedge = \left(\begin{array}{c|c} \omega^\wedge & \rho \\ \hline 0 & 0 \end{array}\right) \in se(3)$$

with $se(3)$ is the **Lie Algebra** of $SE(3)$. On the vice verse, we can define the *vee* operator mapping a matrix $S \in se(3)$ to a vector $\xi \in R^6$ as $S^\vee = \xi$ if $\xi^\wedge = S$. Given the *exponential map* and *log map* [4] :

$$\begin{aligned} exp(S) = T \qquad & \in SE(3) \\ log(T) = S \qquad & \in se(3) \end{aligned}$$

We can also form the *exponential map* and *log map* between a vector $\xi \in R^6$ and a transformation matrix $T \in SE(3)$ in the following:

$$\begin{aligned} Exp(\xi) = exp(\xi^\wedge) = T \qquad & \in SE(3) \\ Log(T) = log(T)^\vee = \xi \qquad & \in R^6 \end{aligned}$$

For a small vector $\delta\xi \in R^6$ we have

$$Exp(\delta\xi) \approx I + (\delta\xi)^\wedge$$

II. Point Reprojection Factor

According to the point re-projection error, we have:

$$\rho_{ij} = (p_i - {p_i}^{'}) \in \mathbb{R}^2$$

with $p_i = (p_i^u, p_i^v)$ is a pixel coordinates in the image plane, ${p_i}^{'}$ is the back-projected pixel coordinates by re-projecting a 3D point ${}^wP_i = (x, y, z)$ to a frame with camera pose $T(\xi_j) = \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix}$ as:

$${p_i}^{'} = \begin{bmatrix} u^{'} \\ v^{'} \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} \frac{P_x}{P_z} \\ \frac{P_y}{P_z} \\ 1 \end{bmatrix}, \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Rightarrow \frac{\partial {p_i}^{'}}{\partial (P_x, P_y, P_z)} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} \frac{1}{P_z} & 0 & -\frac{P_x}{{P_z}^2} \\ 0 & \frac{1}{P_z} & -\frac{P_y}{{P_z}^2} \\ 0 & 0 & 0 \end{bmatrix} \tag{2}$$

$$\Rightarrow \frac{\partial (P_x, P_y, P_z)}{\partial (x, y, z)} = R_j \tag{3}$$

$$\Rightarrow \frac{\partial (P_x, P_y, P_z, 1)}{\partial \xi_j} = \lim_{\delta\xi_j \to 0} \frac{Exp(\delta\xi_j) Exp(\xi_j) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} - Exp(\xi_j) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}{\delta\xi_j} \approx \lim_{\delta\xi_j \to 0} \frac{(I + {\delta\xi_j}^\wedge) Exp(\xi_j) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} - Exp(\xi_j) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}{\delta\xi_j}$$

$$= \lim_{\delta\xi_j \to 0} \frac{\delta\xi_j{}^{\wedge} \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}{\delta\xi_j} = \lim_{\delta\xi_j \to 0} \frac{\begin{bmatrix} \delta\omega_j{}^{\wedge} & \delta\rho_j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}{\delta\xi_j = (\delta\omega_j \ \delta\rho_j)} = \begin{bmatrix} I & -\lfloor R_j \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t_j \rfloor_\times \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I & -\lfloor R_j{}^w P_i + t_j \rfloor_\times \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow \frac{\partial(P_x, P_y, P_z)}{\xi_j} = \begin{bmatrix} I & -\lfloor R_j{}^w P_i + t_j \rfloor_\times \end{bmatrix}_{3\times 6} \tag{4}$$

Combining (2), (3) and (4) we have:

$$\begin{cases} \dfrac{\partial \rho_{ij}}{\partial^w P_i} = \dfrac{\partial \rho_{ij}}{\partial p_i{}'} \dfrac{\partial p_i{}'}{\partial(P_x, P_y, P_z)} \dfrac{\partial(P_x, P_y, P_z)}{\partial(x, y, z)} = -\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} \frac{1}{P_z} & 0 & -\frac{P_x}{P_z{}^2} \\ 0 & \frac{1}{P_z} & -\frac{P_y}{P_z{}^2} \\ 0 & 0 & 0 \end{bmatrix} R_j \\ \dfrac{\partial \rho_{ij}}{\partial \xi_j} = \dfrac{\partial \rho_{ij}}{\partial p_i{}'} \dfrac{\partial p_i{}'}{\partial(P_x, P_y, P_z)} \dfrac{\partial(P_x, P_y, P_z)}{\partial \xi_j} = -\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} \frac{1}{P_z} & 0 & -\frac{P_x}{P_z{}^2} \\ 0 & \frac{1}{P_z} & -\frac{P_y}{P_z{}^2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I & -\lfloor R_j{}^w P_i + t_j \rfloor_\times \end{bmatrix} \end{cases} \tag{5}$$

III. Line Reprojection Factor

3D Plücker Line Representation. For a detected line segment, we use the Plücker coordinates [5] to represent its 3D Line landmark. Suppose a 3D line consists of two spatial endpoints $P^T \sim (\bar{P}^T \mid p)$ and $Q^T \sim (\bar{Q}^T \mid q)$, the Plücker coordinates can be formulated as:

$$L = \begin{bmatrix} n \\ v \end{bmatrix} = \begin{bmatrix} \bar{P} \times \bar{Q} \\ p\bar{Q} - q\bar{P} \end{bmatrix} \in \mathbb{R}^6$$

which is a $6D$ vector consisting of $n \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$. n is the normal vector of the plane determined by the line and the origin. v is the direction vector of the line. The distance from the origin to the line is $d = \frac{|n|}{|v|}$.

Since the degree of freedom of a 3D line is only four, so the Plücker coordinates $L \in \mathbb{R}^6$ is overparametered. To make the Bundle Adjustment numerical accurate, we adopt the orthonormal representation [5] as the minimun parameter for a 3D line. The orthonormal representation $(U \ W) \in SO(3) \times SO(2)$ can be obtained from the Plücker coordinates:

$$\mathcal{L} = [n \mid v] = [\frac{n}{|n|} \ \frac{v}{|v|} \ \frac{n \times v}{|n \times v|}] \begin{bmatrix} |n| & 0 \\ 0 & |v| \\ 0 & 0 \end{bmatrix} \sim U(\Theta) W(\rho)$$

$$U(\Theta) = [\frac{n}{|n|} \ \frac{v}{|v|} \ \frac{n \times v}{|n \times v|}]$$

$$W(\rho) = \begin{bmatrix} \omega_1 & -\omega_2 \\ \omega_2 & \omega_1 \end{bmatrix}, \sigma = [|n| \ |v|], [\omega_1 \ \omega_2] = \frac{\sigma}{|\sigma|}$$

Following the orthonormal representation, we can use a four dimension vector $\phi = [\Theta \ \rho] \in \mathbb{R}^4$ as the minimum parameters during the Bundle Adjustment for each 3D line. Given a small perturbation $\delta_\phi = [\delta_\Theta \ \delta_\rho]$, the four

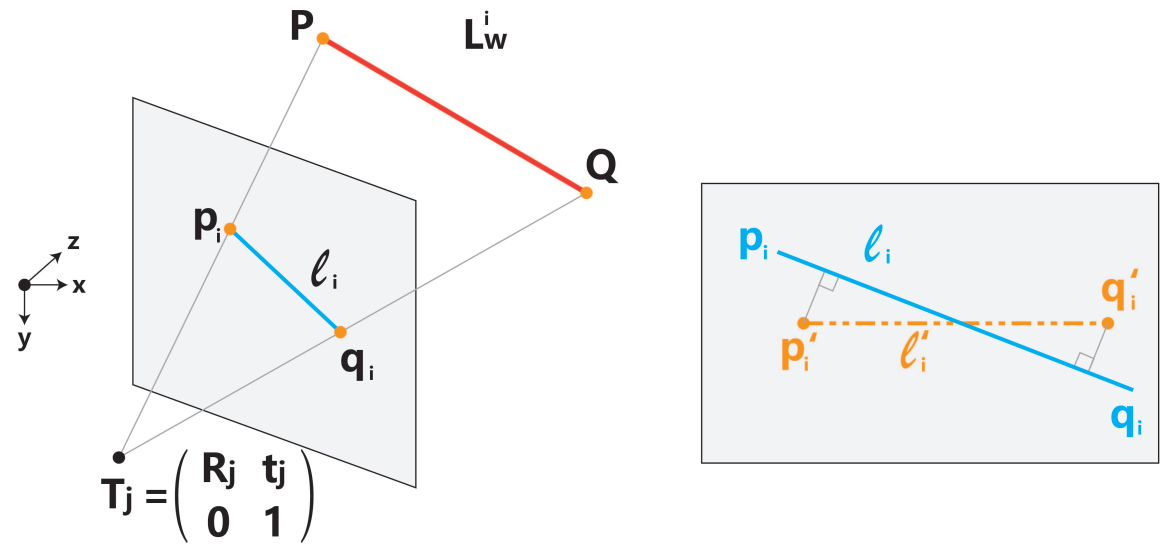

Fig. 1. A 3D line landmark L_w^i is re-projected onto the image plane yielding a 2D line l_i (left), the error between the re-projected line l_i and the matched line segment $l_i^{'}$ consists of two distances from the endpoints to the line l_i (right).

parameters can be updated as, $\hat{\phi} = [\hat{\Theta} \ \hat{\rho}] = \delta_\phi \boxplus \phi$ with $U(\hat{\Theta}) = U(\delta_\Theta)U(\phi)$ and $W(\hat{\rho}) = W(\delta_\rho)W(\rho)$. Note that the orthonormal representation of each 3D line is only used when performing Bundle Adjustment, the Plücker coordinate is maintained in other steps due to its convenience in camera projection, endpoints trimming, and line triangulation [1].

3D Line Projection. Consider a 3D line L_w in the world coordinate, we can project L_w to a image plane with camera pose $T_w^c \in SE(3)$ and obtain a 2D line l as illustrated in Fig 1. Suppose the intrinsic parameters $K = [f_u, f_v, c_u, c_v]$ is known and the camera pose T_w^c denote the rigid transformation from the world coordinate to the camera coordinate, which consist of a rotation matrix $R_w^c \in SO(3)$ and translation vector $t_w^c \in \mathbb{R}^3$, we can first transform the 3D line L_w from the world coordinate to the camera coordinate yeilding L_c:

$$T_w^c = \begin{bmatrix} R_w^c & t_w^c \\ 0 & 1 \end{bmatrix} \in SE(3)$$

$$L_c = \begin{bmatrix} n_c \\ v_c \end{bmatrix} = H_w^c L_w = \begin{bmatrix} R_w^c & \lfloor t_w^c \rfloor_\times R_w^c \\ 0 & R_w^c \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix}$$

which $\lfloor . \rfloor_\times$ denotes the skew-symmetric matrix of a vector. Then the projected line l can be obtained using:

$$l = \mathcal{K} n_c = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & -f_u c_v & f_u f_v \end{bmatrix} n_c = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \in \mathbb{R}^3$$

According to the definition of $\gamma_{ij} = [\frac{{l_i}^T {p_i}^{'}}{\sqrt{{l_i^1}^2 + {l_i^2}^2}} \ \ \frac{{l_i}^T {q_i}^{'}}{\sqrt{{l_i^1}^2 + {l_i^2}^2}}]^T \in \mathbb{R}^2$ with $l_i = [l_i^1 \ l_i^2 \ l_i^3]^T \in \mathbb{R}^3$, ${p_i}^{'} = [p_i^{1'} \ p_i^{2'} \ p_i^{3'}]^T \in \mathbb{R}^3$ and ${q_i}^{'} = [q_i^{1'} \ q_i^{2'} \ q_i^{3'}] \in \mathbb{R}^3$, we have:

$$\frac{\partial \gamma_{ij}}{\partial l_i} = \frac{1}{\sqrt{{l_i^1}^2 + {l_i^2}^2}} \begin{bmatrix} {p_i^1}' - \frac{l_i^1 {l_i}^T {p_i}'}{{l_i^1}^2 + {l_i^2}^2} & {p_i^2}' - \frac{l_i^2 {l_i}^T {p_i}'}{{l_i^1}^2 + {l_i^2}^2} & 1 \\ {q_i^1}' - \frac{l_i^1 {l_i}^T {q_i}'}{{l_i^1}^2 + {l_i^2}^2} & {q_i^2}' - \frac{l_i^2 {l_i}^T {q_i}'}{{l_i^1}^2 + {l_i^2}^2} & 1 \end{bmatrix}_{2\times 3} \tag{6}$$

Since $l_i = \mathcal{K} n_c$, i.e.

$$l_i = [\mathcal{K}\ 0]_{3\times 6} \begin{bmatrix} n_c \\ v_c \end{bmatrix} = [\mathcal{K}\ 0]_{3\times 6} L_c, \qquad \mathcal{K} = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & -f_u c_v & f_u f_v \end{bmatrix}$$

$$\Rightarrow \frac{\partial l_i}{\partial L_c} = [\mathcal{K}\ 0]_{3\times 6} \tag{7}$$

Since $L_c = H_w^c L_w$, i.e.

$$L_c = H_w^c L_w = \begin{bmatrix} R_w^c & \lfloor t_w^c \rfloor_\times R_w^c \\ 0 & R_w^c \end{bmatrix} L_w$$

$$\Rightarrow \frac{\partial L_c}{\partial L_w} = \begin{bmatrix} R_w^c & \lfloor t_w^c \rfloor_\times R_w^c \\ 0 & R_w^c \end{bmatrix}_{6\times 6} \tag{8}$$

According to the orthonormal representation a 3D Line L_w has a minimum parameters $\phi = [\Theta\ \rho] \in \mathbb{R}^4$ with

$$L_w(\phi) = \begin{bmatrix} n_w(\phi) \\ v_w(\phi) \end{bmatrix} = \begin{bmatrix} \omega_1 u_1 \\ \omega_2 u_2 \end{bmatrix}$$

$$U(\Theta) = [u_1\ u_2\ u_3] \in SO(3), W(\rho) = \begin{bmatrix} \omega_1 & -\omega_2 \\ \omega_2 & \omega_1 \end{bmatrix}_{2\times 2}$$

If ϕ has a small perturbation $\delta\phi = [\delta\Theta\ \delta\rho]$ then $L_w(\phi + \delta\phi)$ can be computed as :

$$L_w(\phi + \delta\phi) = \begin{bmatrix} n_w(\phi + \delta\phi) \\ v_w(\phi + \delta\phi) \end{bmatrix} = \begin{bmatrix} {\omega_1}' {u_1}' \\ {\omega_2}' {u_2}' \end{bmatrix}$$

$$U(\Theta + \delta\Theta) = Exp(\delta\Theta) U(\Theta) = [{u_1}'\ {u_2}'\ {u_3}'], W(\rho + \delta\rho) = \begin{bmatrix} {\omega_1}' & -{\omega_2}' \\ {\omega_2}' & {\omega_1}' \end{bmatrix}_{2\times 2} = \begin{bmatrix} cos(\delta\rho) & -sin(\delta\rho) \\ sin(\delta\rho) & cos(\delta\rho) \end{bmatrix} \begin{bmatrix} \omega_1 & -\omega_2 \\ \omega_2 & \omega_1 \end{bmatrix}_{2\times 2}$$

$$\Rightarrow \frac{\partial L_w}{\partial \phi} = \lim_{\delta\phi \to 0} \frac{L_w(\phi + \delta\phi) - L_w(\phi)}{\delta\phi} = \begin{bmatrix} -\lfloor \omega_1 u_1 \rfloor_\times & -\omega_2 u_1 \\ -\lfloor \omega_2 u_2 \rfloor_\times & \omega_1 u_2 \end{bmatrix}_{6\times 4} \tag{9}$$

Similarly if the minimum parameter ξ_j of $T_j(\xi_j)$ has a small perturbation $\delta\xi_j = [\delta\omega_j\ \delta\rho_j]$, then $T_j(\xi_j)$ is updated as:

$$T_j(\xi_j + \delta\xi_j) = Exp(\delta\xi_j) Exp(\xi_j) \approx \begin{bmatrix} Exp(\delta\omega_j) & \delta\rho_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Exp(\delta\omega_j) R_j & \delta\rho_j + Exp(\delta\omega_j) t_j \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow H_w^c(\xi_j + \delta\xi_j) \approx \begin{bmatrix} Exp(\delta\omega_j) R_j & \lfloor \delta\rho_j + Exp(\delta\omega_j) t_j \rfloor_\times Exp(\delta\omega_j) R_j \\ 0 & Exp(\delta\omega_j) R_j \end{bmatrix}$$

$$\Rightarrow L_c(\xi_j+\delta\xi_j)=H_w^c(\xi_j+\delta\xi_j)L_w \approx \begin{bmatrix} Exp(\delta\omega_j)R_j & \lfloor \delta\rho_j + Exp(\delta\omega_j)t_j \rfloor_\times Exp(\delta\omega_j)R_j \\ 0 & Exp(\delta\omega_j)R_j \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix}$$

$$\approx \begin{bmatrix} (I+\lfloor\delta\omega_j\rfloor_\times)R_j & \lfloor \delta\rho_j + (I+\lfloor\delta\omega_j\rfloor_\times)t_j \rfloor_\times (I+\lfloor\delta\omega_j\rfloor_\times)R_j \\ 0 & (I+\lfloor\delta\omega_j\rfloor_\times)R_j \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix}$$

$$\approx \begin{bmatrix} R_j & \lfloor t_j\rfloor_\times R_j \\ 0 & R_j \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix} + \begin{bmatrix} \lfloor\delta\omega_j\rfloor_\times R_j & \lfloor\delta\rho_j\rfloor_\times R_j + \lfloor\lfloor\delta\omega_j\rfloor_\times t_j\rfloor_\times R_j + \lfloor t_j\rfloor_\times\lfloor\delta\omega_j\rfloor_\times R_j \\ 0 & \lfloor\delta\omega_j\rfloor_\times R_j \end{bmatrix} \begin{bmatrix} n_w \\ v_w \end{bmatrix}$$

$$\approx L_c(\xi_j) + \begin{bmatrix} \lfloor\delta\omega_j\rfloor_\times R_j n_w + \lfloor\delta\rho_j\rfloor_\times R_j v_w + \lfloor\lfloor\delta\omega_j\rfloor_\times t_j\rfloor_\times R_j v_w + \lfloor t_j\rfloor_\times\lfloor\delta\omega_j\rfloor_\times R_j v_w \\ \lfloor\delta\omega_j\rfloor_\times R_j v_w \end{bmatrix}$$

$$\approx L_c(\xi_j) + \begin{bmatrix} -\lfloor R_j n_w\rfloor_\times \delta\omega_j - \lfloor R_j v_w\rfloor_\times \delta\rho_j - \lfloor\lfloor t_j\rfloor_\times R_j v_w\rfloor_\times \delta\omega_j \\ -\lfloor R_j v_w\rfloor_\times \delta\omega_j \end{bmatrix}$$

$$= L_c(\xi_j) + \begin{bmatrix} -\lfloor R_j n_w\rfloor_\times - \lfloor\lfloor t_j\rfloor_\times R_j v_w\rfloor_\times & -\lfloor R_j v_w\rfloor_\times \\ -\lfloor R_j n_w\rfloor_\times & \end{bmatrix} \begin{bmatrix} \delta\omega_j \\ \delta\rho_j \end{bmatrix}$$

$$\Rightarrow \frac{\partial L_c}{\partial \xi_j} = \lim_{\delta\xi_j\to 0} \frac{L_c(\xi_j+\delta\xi_j)-L_c(\xi_j)}{\delta\xi_j} = \begin{bmatrix} -\lfloor R_j n_w\rfloor_\times - \lfloor\lfloor t_j\rfloor_\times R_j v_w\rfloor_\times & -\lfloor R_j v_w\rfloor_\times \\ -\lfloor R_j v_w\rfloor_\times & 0 \end{bmatrix}_{6\times 6} \tag{10}$$

Combining (6),(7),(8) and (9) we get:

$$\frac{\partial \gamma_{ij}}{\partial \phi_i} = \frac{\partial \gamma_{ij}}{\partial l_i}\frac{\partial l_i}{\partial L_c}\frac{\partial L_c}{\partial L_w}\frac{\partial L_w}{\partial \phi_i}$$

$$= \frac{1}{\sqrt{{l_i^1}^2+{l_i^2}^2}} \begin{bmatrix} {p_i^1}' - \frac{l_i^1 {l_i}^T {p_i}'}{{l_i^1}^2+{l_i^2}^2} & {p_i^2}' - \frac{l_i^2 {l_i}^T {p_i}'}{{l_i^1}^2+{l_i^2}^2} & 1 \\ {q_i^1}' - \frac{l_i^1 {l_i}^T {q_i}'}{{l_i^1}^2+{l_i^2}^2} & {q_i^2}' - \frac{l_i^2 {l_i}^T {q_i}'}{{l_i^1}^2+{l_i^2}^2} & 1 \end{bmatrix} [\mathcal{K}\ 0] \begin{bmatrix} R_w^c & \lfloor t_w^c\rfloor_\times R_w^c \\ 0 & R_w^c \end{bmatrix} \begin{bmatrix} -\lfloor \omega_1 u_1\rfloor_\times & -\omega_2 u_1 \\ -\lfloor \omega_2 u_2\rfloor_\times & \omega_1 u_2 \end{bmatrix} \tag{11}$$

Combining (6),(7),(8) and (10) we get:

$$\frac{\partial \gamma_{ij}}{\partial \xi_j} = \frac{\partial \gamma_{ij}}{\partial l_i}\frac{\partial l_i}{\partial L_c}\frac{\partial L_c}{\partial \xi_j}$$

$$= \frac{1}{\sqrt{{l_i^1}^2+{l_i^2}^2}} \begin{bmatrix} {p_i^1}' - \frac{l_i^1 {l_i}^T {p_i}'}{{l_i^1}^2+{l_i^2}^2} & {p_i^2}' - \frac{l_i^2 {l_i}^T {p_i}'}{{l_i^1}^2+{l_i^2}^2} & 1 \\ {q_i^1}' - \frac{l_i^1 {l_i}^T {q_i}'}{{l_i^1}^2+{l_i^2}^2} & {q_i^2}' - \frac{l_i^2 {l_i}^T {q_i}'}{{l_i^1}^2+{l_i^2}^2} & 1 \end{bmatrix} [\mathcal{K}\ 0] \begin{bmatrix} -\lfloor R_j n_w\rfloor_\times - \lfloor\lfloor t_j\rfloor_\times R_j v_w\rfloor_\times & -\lfloor R_j v_w\rfloor_\times \\ -\lfloor R_j v_w\rfloor_\times & 0 \end{bmatrix} \tag{12}$$

IV. Plane Reprojection Factor

We adopt a quaternion $q \in S^3$ as the minimal parametrization of a plane π [3]. Given a camera pose T_w^c, we can transform a $3D$ Euclidian point p_w in the world frame to a point p_c in the camera frame, i.e. $p_c = T_w^c p_w$. The plane π_c in the camera frame can also be transformed into a plane π_w in the world frame using $q(\pi_w) = {T_w^c}^T q(\pi_c)$. This means we can reproject a plane $q(\pi_c)$ in the camera frame into the world frame as $3D$ point does. During the pose estimation, when a plane $q(\pi_c)$ is detected in the current frame with camera pose T_w^c which correspondes to a plane landmak plane $q(\pi_w)$ in the world frame, we can use the above *log map* [3] to measure the pose estimation error as

$$e = log(q((T_w^c)^T q(\pi_c)) \oplus q(\pi_w)^{-1})$$

this is called the plane reprojection error. Since the camera pose T_w^c and plane landmark $q(\pi_w)$ is to be estimated, and the minimal parameterizations for T_w^c and $q(\pi_w)$ are $\xi \in R^6$ and $\omega \in R^3$ separately, the plane reprojection error is actually the function of ξ and ω, i.e.

$$e = E(\xi, \omega) = log(q((T_w^c(\xi))^T q(\pi_c)) \oplus q(\omega)^{-1})$$

let denote $q(\pi_c)$ as $q(\hat{\omega})$ then

$$e = E(\xi, \omega) = log(q((T_w^c(\xi))^T q(q(\hat{\omega}))) \oplus q(\omega)^{-1})$$

In the following we given a detailed deviration of plane reprojection error fuction $E(\xi, \omega)$'s Jacobian matrix $J = \frac{\partial E(\xi,\omega)}{\partial(\xi,\omega)}$ by computing the $\frac{\partial E(\xi,\omega)}{\partial \xi}$ and $\frac{\partial E(\xi,\omega)}{\partial \omega}$ separately, which is a 3×9 dementional matrix.

A. *Derivation of* $\frac{\partial E(\xi,\omega)}{\partial \xi}$

Let denote:

$$y(\xi) = T_w^c(\xi)^T q(\hat{\omega}) \in R^4$$
$$\phi(\xi) = q(y(\xi)) = \frac{y(\xi)}{|y(\xi)|} \in S^3$$
$$u = log(\phi(\xi)) \qquad \in R^3$$

Then we have

$$E(\xi, \omega) = log(exp(u) \oplus exp(\omega)^{-1})$$

According the derivation chain rule we have:

$$\frac{\partial E(\xi, \omega)}{\partial \xi} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial \phi}\frac{\partial \phi}{\partial y}\frac{\partial y}{\partial \xi} \tag{13}$$

$$\begin{aligned}
\frac{\partial E}{\partial u} &= \frac{log(exp(\delta u) \oplus exp(u) \oplus exp(\omega)^{-1}) - log(exp(u) \oplus exp(\omega)^{-1})}{\delta u} \\
&\qquad (exp(\beta) \doteq exp(u) \oplus exp(\omega)^{-1}) \\
&= \frac{log(exp(\delta u) \oplus exp(\beta)) - log(exp(\beta))}{\delta u} \\
&= \frac{J_l(\beta)^{-1}\delta u + \beta - \beta}{\delta u} \qquad (\textbf{BCH formula}) \\
&= J_l(\beta)^{-1}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial u}{\partial \phi} &= \frac{\partial \frac{2*cos^{-1}(\phi_4)}{|(\phi_1,\phi_2,\phi_3)|}\begin{pmatrix}\phi_1\\ \phi_2\\ \phi_3\end{pmatrix}}{\partial(\phi_1, \phi_2, \phi_3, \phi_4)} \quad (\phi = \begin{pmatrix}\phi_1\\ \phi_2\\ \phi_3\\ \phi_4\end{pmatrix} = \begin{pmatrix}\phi_v \in R^3\\ \phi_4\end{pmatrix}) \\
&= \left(\begin{array}{c|c} \frac{2cos^{-1}(\phi_4)}{|\phi_v|} I_{3\times3} - \frac{2cos^{-1}(\phi_4)}{|q|^{\frac{3}{2}}}\phi_v{\phi_v}^T & \frac{2cos^{-1}(\phi_4)}{\sqrt{1-{\phi_4}^2}}\phi_v \end{array}\right)
\end{aligned}$$

$$\frac{\partial\phi}{\partial y} = \frac{\partial \frac{y}{|y|}}{\partial y} = \frac{1}{|y|} I_{4\times 4} - \frac{1}{|y|^{\frac{3}{2}}} y y^T \qquad (y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \in R^4)$$

$$\begin{aligned}
\frac{\partial y}{\partial \xi} &= \frac{(Exp(\delta\xi)Exp(\xi))^T q(\hat{\omega}) - Exp(\xi)^T q(\hat{\omega})}{\delta\xi} \\
&= \frac{Exp(\xi)^T Exp(\delta\xi)^T q(\hat{\omega}) - Exp(\xi)^T q(\hat{\omega})}{\delta\xi} \\
&\approx \frac{Exp(\xi)^T (I + \delta\xi^\wedge)^T q(\hat{\omega}) - Exp(\xi)^T q(\hat{\omega})}{\delta\xi} \\
&= \frac{Exp(\xi)^T \delta{\xi^\wedge}^T q(\hat{\omega})}{\delta\xi} \\
&= \frac{\left(\begin{array}{c|c} R^T & 0 \\ \hline t^T & 1 \end{array}\right) \left(\begin{array}{c|c} \delta{\phi^\wedge}^T & 0 \\ \hline \delta\rho^T & 0 \end{array}\right) \begin{pmatrix} q_v^{\hat{\omega}} \in R^3 \\ q_4^{\hat{\omega}} \end{pmatrix}}{\delta\xi = (\delta\phi \quad \delta\rho)^T} \\
&= \frac{\begin{pmatrix} R^T \delta{\phi^\wedge}^T q_v^{\hat{\omega}} \\ t^T \delta{\phi^\wedge}^T q_v^{\hat{\omega}} + \delta\rho^T q_v^{\hat{\omega}} \end{pmatrix}}{\delta\xi = (\delta\phi \quad \delta\rho)^T} \\
&\qquad since \quad {\omega^\wedge}^T = -\omega^\wedge, \omega^\wedge a = -a^\wedge \omega \\
&= \frac{\begin{pmatrix} R^T {q_v^{\hat{\omega}}}^\wedge \delta\phi \\ t^T {q_v^{\hat{\omega}}}^\wedge \delta\phi + {q_v^{\hat{\omega}}}^T \delta\rho \end{pmatrix}}{\delta\xi = (\delta\phi \quad \delta\rho)^T} \\
&= \left(\begin{array}{c|c} R^T {q_v^{\hat{\omega}}}^\wedge & 0 \\ \hline t^T {q_v^{\hat{\omega}}}^\wedge & {q_v^{\hat{\omega}}}^T \end{array}\right)
\end{aligned}$$

B. Derivation of $\frac{\partial E(\xi,\omega)}{\partial\omega}$

$$\begin{aligned}
\frac{\partial E(\xi,\omega)}{\partial\omega} &= \frac{log(q({T_w^c}^T q(\hat{\omega})) \oplus (q(\delta\omega) \oplus q(\omega))^{-1}) - log(q({T_w^c}^T q(\hat{\omega})) \oplus q(\omega)^{-1})}{\delta\omega} \\
&= \frac{log(q({T_w^c}^T q(\hat{\omega})) \oplus q(\omega)^{-1} \oplus q(\delta\omega)^{-1}) - log(q({T_w^c}^T q(\hat{\omega})) \oplus q(\omega)^{-1})}{\delta\omega} \\
&\qquad let \qquad exp(u) = q({T_w^c}^T q(\hat{\omega})) \oplus q(\omega)^{-1} \\
&= \frac{log(exp(u) \oplus exp(\delta\omega)^{-1} - log(exp(u)))}{\delta\omega} \\
&= \frac{log(exp(u) \oplus exp(-\delta\omega) - log(exp(u)))}{\delta\omega} \\
&\approx \frac{J_r(u)^{-1}(-\delta\omega) + u - u}{\delta\omega} \qquad (\textbf{BCH formula}) \\
&= -J_r(u)^{-1}
\end{aligned}$$

REFERENCES

[1] Guoxuan Zhang, Jin Han Lee, Jongwoo Lim, and Il Hong Suh. Building a 3-d line-based map using stereo SLAM. *IEEE Trans. Robotics*, 31(6):1364–1377, 2015.

[2] Xingxing Zuo, Xiaojia Xie, Yong Liu, and Guoquan Huang. Robust visual SLAM with point and line features. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 1775–1782, 2017.

[3] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 4605–4611, 2015.

[4] G.S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2011.

[5] Adrien Bartoli and Peter F. Sturm. The 3d line motion matrix and alignment of line reconstructions. *International Journal of Computer Vision*, 57(3):159–178, 2004.