

计算机视觉中的矩阵李群

王京

jp08@foxmail.com

UAV514, 自动化系, 清华大学

说明：这只是一个用于交流的大致的总结笔记，如有谬误请联系作者。

2014-04-18, Version 0

2014-04-28, Version 1, 增加 4.4

目录

1 3D 刚体运动的表示	3
1.1 刚体（坐标系）3D 旋转	3
1.2 旋转作用下的固定点坐标变换	3
1.3 坐标系旋转与固定点坐标变换关系	3
1.4 刚体运动（平移+旋转）	3
1.5 位姿更新、左乘与右乘	4
2 3D 运动李群表示	4
2.1 李群基本数学定义	4
2.2 反对称矩阵(skew-symmetric)	5
2.3 李代数	5
2.4 $SO(3)$ 与 $so(3)$ 的关系	6
2.5 $SE(3)$ 与 $se(3)$ 的关系	7
2.6 相似变换、 $Sim(3)$ 李群与其李代数的关系	8
2.7 李群与李代数关系总结	9
3 李群的增量和导数表示	10
3.1 李群中的增量及相对于增量的导数	10
3.2 点坐标变换关于李代数增量的导数	11
4 李群在视觉 SLAM 中的应用	12
4.1 重投影误差（reprojection error）的导数	12
4.2 光束平差法（BA, Bundle Adjustment）	13
4.3 利用最小化重投影误差来计算相似变换	14
4.4 光测量误差（photometric error）最小化	18
5 其他	19
5.1 欧拉角与 $SO(3)$ 的关系	19

1.3D 刚体运动的表示

1.1 刚体（坐标系）3D 旋转

考虑对某个物体进行旋转，假设此物体上固连了一个坐标系 \mathcal{F} ，其单位正交基为 e_1, e_2, e_3 （3个互相正交的单位向量），那么对此物体的旋转等同于对坐标系进行的旋转。

经过旋转，固连的坐标系从 $\mathcal{F}^w = [e_1^w, e_2^w, e_3^w]$ （基组成的矩阵）旋转到了 $\mathcal{F}^c = [e_1^c, e_2^c, e_3^c]$ 。旋转后的基向量满足

$$\begin{aligned} e_1^c &= r_{11}e_1^w + r_{21}e_2^w + r_{31}e_3^w \\ e_2^c &= r_{12}e_1^w + r_{22}e_2^w + r_{32}e_3^w \\ e_3^c &= r_{13}e_1^w + r_{23}e_2^w + r_{33}e_3^w \end{aligned}$$

即

$$[e_1^c, e_2^c, e_3^c] = [e_1^w, e_2^w, e_3^w] \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \Leftrightarrow \mathcal{F}^c = \mathcal{F}^w \cdot R_{wc} \Rightarrow (\mathcal{F}^w)^{-1} \cdot \mathcal{F}^c = R_{wc}$$

可以说，将 \mathcal{F}^w 经过旋转 R_{wc} 得到了 \mathcal{F}^c 。

根据线性代数知识， R_{wc} 是一个单位正交阵， $R_{wc}R_{wc}^T = I$ 并且 $\det(R_{wc}) = 1$ 。

1.2 旋转作用下的固定点坐标变换

考虑 3D 空间中的固定点 P ，其在 \mathcal{F}^w 中坐标 $p_w = [x_1, y_1, z_1]$ ，在 \mathcal{F}^c 中坐标 $p_c = [x_2, y_2, z_2]$ ，即

$$P = \mathcal{F}^w \cdot p_w = \mathcal{F}^c \cdot p_c$$

那么，显然有

$$p_w = (\mathcal{F}^w)^{-1} \cdot \mathcal{F}^c \cdot p_c = R_{wc} \cdot p_c$$

以及

$$p_c = R_{wc}^{-1} \cdot p_w = R_{wc}^T \cdot p_w \stackrel{\text{def}}{=} R_{cw} \cdot p_w$$

1.3 坐标系旋转与固定点坐标变换关系

其实上面的 R_{cw} 和 R_{wc} 都可以表示物体的旋转，直观来说 R_{wc} 更直白一些，而 R_{cw} 表示的是固定点的坐标变换关系。从上面的推导可以看出，这两者之间就是互为逆矩阵。

（这是线性代数基表示的基本数学公式，这里算是回顾一下。）

如果把物体具体化成相机，要知道这个相机的欧拉角姿态，那么根据一般的偏航 ψ -俯仰 θ -滚转 ϕ 顺序，三次坐标系变换依次为

$$\mathcal{F}^\psi = \mathcal{F}^w \cdot R_\psi, \quad \mathcal{F}^\theta = \mathcal{F}^\psi \cdot R_\theta, \quad \mathcal{F}^c = \mathcal{F}^\theta \cdot R_\phi, \quad \Rightarrow \mathcal{F}^c = \mathcal{F}^w \cdot R_\psi R_\theta R_\phi$$

因此 $R_{wc} = R_\psi R_\theta R_\phi$ ，利用这个等式计算偏航 ψ -俯仰 θ -滚转 ϕ 的角度。

（注意 $R_\psi R_\theta R_\phi$ 的顺序，是按照先后顺序从左至右，新的变换乘在右边。后文会说明，因为这是坐标系旋转，是右乘的方式）

1.4 刚体运动（平移+旋转）

假设物体在旋转以外，还进行了平移。这个平移无法通过基向量进行表示，但可以另外引入一个三维向量 $t \in \mathbb{R}^3$ ，来代表坐标系原点 O 的位移

$$O_c = O_w + t_{wc}$$

即 \mathcal{F}^w 经 t_{wc} 平移到 \mathcal{F}^c 了，而且在 \mathcal{F}^w 下 O_c 坐标即为 t_{wc} 。

那么，坐标系的这个运动也就可以建模成旋转 R_{wc} 和平移 t_{wc} 的形式。

仍然考虑某个固定点的坐标变换，运动后 \mathcal{F}^c 的原点 O_c 在 \mathcal{F}^w 中的坐标即为 t_{wc} ，从 O_w 到 P 的向量 $V_{O_w \rightarrow P}$ 是两个原点之间向量与 O_c 到点 P 向量之和，即

$$V_{O_w \rightarrow P} = V_{O_w \rightarrow O_c} + V_{O_c \rightarrow P}$$

写成基向量和坐标的形式有

$$\mathcal{F}^w \cdot p_w = \mathcal{F}^w \cdot t_{wc} + \mathcal{F}^c \cdot p_c \quad (1)$$

两边同时左乘 $(\mathcal{F}^w)^{-1}$ 可以得到

$$p_w = t_{wc} + (\mathcal{F}^w)^{-1} \cdot \mathcal{F}^c \cdot p_c = t_{wc} + R_{wc} \cdot p_c$$

上市两边再同时左乘 $R_{wc}^T = R_{cw}$ 可以得到

$$p_c = R_{cw}p_w - R_{cw}t_{wc} \stackrel{\text{def}}{=} R_{cw}p_w + t_{cw}$$

其中 $t_{cw} = -R_{cw}t_{wc} = -R_{wc}^T t_{wc}$

上面的坐标如果写成齐次坐标如 $\dot{p}_c = [p_c; 1]$ 的形式，则这个点在两个坐标系之间的坐标变换关系为

$$\dot{p}_c = \begin{bmatrix} R_{cw} & t_{cw} \\ 0 & 1 \end{bmatrix} \cdot \dot{p}_w \stackrel{\text{def}}{=} T_{cw}\dot{p}_w, \quad \dot{p}_w = \begin{bmatrix} R_{wc} & t_{wc} \\ 0 & 1 \end{bmatrix} \cdot \dot{p}_c \stackrel{\text{def}}{=} T_{wc}\dot{p}_c$$

这里的 $T_{cw}, T_{wc} \in \mathbb{R}^4$ 也就是常说的位姿（pose），这两者都可以表示相机位姿，只不过一个是坐标系的运动表示 T_{wc} ，另一个是固定点的坐标变换表示 T_{cw} 。

同样的，如果要直观的获得相机当前的位置和角度，应当通过 T_{wc} 来计算 R_{wc}, t_{wc} ，而如果考虑固定点的坐标，则应当通过 T_{cw} 来进行变换计算。

1.5 位姿更新、左乘与右乘

假设世界坐标系 \mathcal{F}^w 是固定的参考坐标系， \mathcal{F}^c 是一直在运动的当前相机坐标系，那么相机当前时刻一个微小的运动（位姿运动更新），固连的坐标系运动到 $\mathcal{F}^{c'}$ ，与上面(1)式一样有

$$\mathcal{F}^c \cdot p_c = \mathcal{F}^c \cdot t_{cc'} + \mathcal{F}^{c'} \cdot p_{c'}$$

即

$$p_c = t_{cc'} + (\mathcal{F}^c)^{-1} \cdot \mathcal{F}^{c'} \cdot p_{c'} = R_{cc'}p_{c'} + t_{cc'}$$

齐次坐标

$$\dot{p}_c = T_{cc'}\dot{p}_{c'}$$

则

$$\dot{p}_w = T_{wc}\dot{p}_c = T_{wc} \cdot T_{cc'}\dot{p}_{c'} = T_{wc'}\dot{p}_{c'}$$

那么

$$T_{wc'} = T_{wc} \cdot T_{cc'}$$

也就是说，坐标系运动的位姿更新，是在原有位姿 T_{wc} 基础上，右乘一个运动变化量 $T_{cc'}$ 。

而点的坐标变换矩阵 $T_{c'w}$ 满足

$$T_{c'w} = T_{wc'}^{-1} = (T_{wc} \cdot T_{cc'})^{-1} = T_{cc'}^{-1} \cdot T_{wc}^{-1} = T_{c'c} \cdot T_{cw}$$

也就是说，点坐标变换的位姿表示，是在原有位姿 T_{cw} 基础上，左乘变化量 $T_{c'c}$ 。

如果不考虑几何运动关系，纯数学表示上 T_{wc} 和 T_{cw} 都是同样的形式。但是这两种不同的几何含义下，更新方式却分为了左乘和右乘，数学处理手段也会不太一样。

由于计算机视觉中，对场景中固定点在不同坐标系下的转换用的比较多，所以很多时候默认是 T_{cw} 的形式，位姿的更新方式也默认是左乘的方式。

后面推导李群增量的优化形式也会按左乘方式进行，具体见下文。

2 3D 运动李群表示

2.1 李群基本数学定义

李群就是一个群，只是在此基础上要求是光滑（无限次可微）的，所以像整数群 \mathbb{Z} 就不是一个李群。

群是一个集合 G ，以及在这个集合上定义的操作符 $\circ: G \circ G \rightarrow G$ ，并且满足 4 个要求：

- 封闭性： $g_1 \circ g_2 \in G, \forall g_1, g_2 \in G$

- 结合律: $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3), \forall g_1, g_2, g_3 \in G$
- 单位元: $\exists e \in G: e \circ g = g \circ e = g, \forall g \in G$
- 可逆性: $\exists g^{-1} \in G: g \circ g^{-1} = g^{-1} \circ g = e, \forall g \in G$

典型的 (矩阵) 群有

- 一般线性群: $GL(n)$

由所有 $n \times n$ 的可逆矩阵所构成, 操作符为矩阵乘法, 可以验证其满足 4 条群公理。显然, 其单位元是单位矩阵 $I_{n \times n}$, 逆元素就是矩阵的逆。

- 正交群: $O(n) \subset GL(n)$

$$O(n) = \{R \in GL(n) | R^T R = I\}$$

这里面包括了旋转和镜像作用 (分别满足 $\det(R) = +1$ or -1)

- 特殊正交群: $SO(n) \subset O(n) \subset GL(n)$

$$SO(n) = \{R \in GL(n) | R^T R = I, \det(R) = +1\}$$

代表了旋转作用 ($O(n)$ 中镜像作用, 即 $\det(R) = -1$ 的部分, 对矩阵乘法不封闭, 不构成群)

- 欧几里得群 Euclidean Group: $E(n) \subset GL(n+1)$

$$E(n) = \left\{ \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \mid R \in O(n), t \in \mathbb{R}^n \right\}$$

- 特殊欧几里得群 Special Euclidean Group: $SE(n) \subset E(n) \subset GL(n+1)$

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

而在三维空间 \mathbb{R}^3 中相关的常用李群, 即 $SO(3)$ 和 $SE(3)$ 。

2.2 反对称矩阵(skew-symmetric)

对两个 3D 向量 $a = [a_1, a_2, a_3]^T \in \mathbb{R}^3$ 和 $b = [b_1, b_2, b_3]^T \in \mathbb{R}^3$ 进行叉乘, 结果为

$$a \times b = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \cdot b \stackrel{\text{def}}{=} [a]_{\times} b$$

也就是说, 叉乘可以表示为一个矩阵与向量相乘的形式, 其中矩阵 $[a]_{\times}$ 是一个反对称矩阵

$$[a]_{\times}^T = -[a]_{\times}$$

并且, 任何反对称矩阵 (对角线元素为零, 对称位置元素互为相反数) 也都可以表示为某个向量对应的的叉乘矩阵

$$W = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} = [w]_{\times}, \quad w = [w_1, w_2, w_3]^T$$

一般在李群相关的表示中, 反对称矩阵在这里表示为一个 **hat** 操作符 (后面会用到), 即

$$[w]_{\times} \stackrel{\text{def}}{=} \hat{w}$$

假设 $w = \phi \cdot u$, 其中 u 为单位向量, $\phi = |w|$, 那么反对称矩阵满足如下性质:

$$\hat{w}^2 = \hat{w} \hat{w} = ww^T - \phi^2 I = \phi^2 \hat{u} \hat{u}, \quad \hat{w}^3 = \hat{w} \cdot \hat{w} \hat{w} = -\phi^2 \hat{w}$$

2.3 李代数

李代数也是一个数学空间 (一个数域上的代数, algebra over a field. 参考 wikipedia)。

Algebra over a field 是一个向量空间, 以及在这个向量空间中定义的一个双线性乘法。

只看和矩阵李群对应的李代数, 这个双线性乘法称为李括号 (Lie bracket), 满足 4 条性质。

(性质略: 封闭性、双线性、alternating、雅克比等式, 参考[4])

(1) 与 $SO(3)$ 对应的李代数定义为

向量空间: $\mathfrak{so}(3) = \{W = \hat{w} \in \mathbb{R}^{3 \times 3}, w \in \mathbb{R}^3\}$

数域: 实数域 \mathbb{R}

李括号: $[W_1, W_2] = W_1 W_2 - W_2 W_1$

(习惯上有些会直接把向量空间 $\mathfrak{so}(3)$ 称为 $SO(3)$ 对应的李代数)

(2) 与 $SE(3)$ 对应的李代数定义为

向量空间: $\mathfrak{se}(3) = \{\Xi = \hat{\xi} \in \mathbb{R}^{4 \times 4}, \xi = [w, v]^T \in \mathbb{R}^6, w \in \mathbb{R}^3, v \in \mathbb{R}^3\}$

数域: 实数域 \mathbb{R}

李括号: $[\Xi_1, \Xi_2] = \Xi_1 \Xi_2 - \Xi_2 \Xi_1$

其中 $\mathfrak{se}(3)$ 中的 $\hat{\cdot}$ 操作符定义为 (定义的来源见下文)

$$\hat{\xi} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}, \quad \xi = [w, v]^T$$

2.4 $SO(3)$ 与 $\mathfrak{so}(3)$ 的关系

(以下 2.4 的过程参考[1])

行列式为 1 的三维单位正交矩阵 R 代表 3D 旋转运动, R 是三维旋转 (矩阵) 李群 $SO(3)$ 中的元素, 正交矩阵意味着满足 $R^T R = I$ 。

假设这个旋转运动随时间 τ 变化表示为 $R(\tau)$, 那么, 对 $R(\tau)R(\tau)^T = I$ 关于 τ 求导, 有

$$(R(\tau)\dot{R}(\tau)^T) = 0$$

展开括号有

$$R(\tau)\dot{R}(\tau)^T + \dot{R}(\tau)R(\tau)^T = 0$$

$$\Rightarrow \dot{R}(\tau)R(\tau)^T = -R(\tau)\dot{R}(\tau)^T = -[\dot{R}(\tau)R(\tau)^T]^T$$

也就是说, $\dot{R}(\tau)R(\tau)^T$ 是一个反对称矩阵 Ω , 满足 $\Omega^T = -\Omega$ 。

从 2.2 可以看出, 存在一个 3D 向量 $w \in \mathbb{R}^3$ 的反对称矩阵 \hat{w} 和 $\dot{R}(\tau)R(\tau)^T$ 对应, 即

$$\hat{w} = \dot{R}(\tau)R(\tau)^T$$

上式左右两边乘以 $R(\tau)$, 有

$$\dot{R}(\tau) = \hat{w} \cdot R(\tau)$$

这个形式, 和一般实数域 \mathbb{R} 上的指数函数 $f(x) = \exp(ax)$ 的导数形式类似, 即

$$\dot{f}(x) = a \cdot f(x)$$

扩展到矩阵上, 也满足指数映射的关系, 即

$$R(\tau) = \exp(\hat{w}\tau)$$

令 $w = \omega\tau$, 则

$$R(\tau) = \exp(\hat{w}\tau) = \exp(\hat{w})$$

也就是说, $SO(3)$ 中的元素 $R(\tau)$ 和 $\mathfrak{so}(3)$ 中的元素 \hat{w} 之间, 满足指数映射的关系。

(除矩阵李群外, 其他的李群与李代数之间的映射关系也称为指数映射, 但只是名字这样称呼, 本质上不一定是这种指数函数的形式。)

实数域中指数函数的 Taylor 展开式为

$$\exp(x) = \sum_{n=0}^{+\infty} \frac{x^n}{n!}$$

对矩阵的指数函数, 同样有

$$\exp(M) = \sum_{n=0}^{+\infty} \frac{M^n}{n!}$$

那么

$$\exp(\hat{w}) = \sum_{n=0}^{+\infty} \frac{\hat{w}^n}{n!} = I + \hat{w} + \frac{\hat{w}^2}{2!} + \frac{\hat{w}^3}{3!} + \frac{\hat{w}^4}{4!} + \dots$$

根据 2.2 中反对称矩阵的性质，将向量 w 表示为模长乘以单位向量形式 $w = |w| \cdot u = \phi u$ ，有

$$\hat{w}^2 = \phi^2 \hat{u} \hat{u}, \quad \hat{w}^3 = -\phi^3 \hat{u}$$

上式 $\exp(\hat{w})$ 的 Taylor 展开可以整理为

$$\begin{aligned} \exp(\hat{w}) &= I + \left(\phi - \frac{\phi^3}{3!} + \frac{\phi^5}{5!} - \frac{\phi^7}{7!} + \dots \right) \hat{u} + \left(\frac{\phi^2}{2!} - \frac{\phi^4}{4!} + \frac{\phi^6}{6!} - \frac{\phi^8}{8!} + \dots \right) \hat{u} \hat{u} \\ &= I + \sin(\phi) \cdot \hat{u} + (1 - \cos(\phi)) \cdot \hat{u} \hat{u} \end{aligned}$$

即 Rodrigues 公式

$$R = \exp(\hat{w}) = I + \sin(|w|) \cdot \frac{\hat{w}}{|w|} + (1 - \cos(|w|)) \cdot \frac{\hat{w} \hat{w}}{|w|^2}$$

$SO(3)$ 与 $\mathfrak{so}(3)$ 元素之间的指数映射关系可以按这个公式进行计算。

需要说明的是，由于三角函数的周期性， $SO(3)$ 与 $\mathfrak{so}(3)$ 之间的映射并不是一一对应的，一个 $SO(3)$ 元素与无穷多个 $\mathfrak{so}(3)$ 元素相对应（角度相差 $n \times 2\pi$ ）。但是，如果限制在 $[-\pi, \pi)$ ，甚至是一个微小增量满足 $w \approx 0$ 的话，则可以是一一对应的。

2.5 $SE(3)$ 与 $\mathfrak{se}(3)$ 的关系

对于 $SE(3)$ 中的位姿元素 $T \in SE(3)$ ，其形式

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \quad R \in SO(3)$$

逆矩阵为

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

同样参考 $SO(3)$ 的推导过程，考虑位姿关于时间 τ 变化表示为 $T(\tau)$ ，关于时间的导数与逆矩阵相乘为

$$\dot{T}(\tau) T(\tau)^{-1} = \begin{bmatrix} \dot{R} & \dot{t} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{R} R^T & \dot{t} - \dot{R} R^T t \\ 0 & 0 \end{bmatrix}$$

由于 $R \in SO(3)$ ，所以存在 $w \in \mathbb{R}^3$ 满足 $\dot{R} R^T = [w]_{\times} = \hat{w}$ ，因此

$$\dot{T}(\tau) T(\tau)^{-1} = \begin{bmatrix} \hat{w} & \dot{t} - \hat{w} t \\ 0 & 0 \end{bmatrix}$$

令 $v = \dot{t} - \hat{w} t \in \mathbb{R}^3$ ，则

$$\dot{T}(\tau) T(\tau)^{-1} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}$$

两边同时右乘 $T(\tau)$ 有

$$\dot{T}(\tau) = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \cdot T(\tau)$$

也是指数函数导数的形式，其中 6 自由度李代数 $\mathfrak{se}(3)$ 的元素和 hat 运算符表示为

$$\hat{\xi} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \in \mathfrak{se}(3), \quad \xi = \begin{bmatrix} w \\ v \end{bmatrix} \in \mathbb{R}^6, w \in \mathbb{R}^3, v \in \mathbb{R}^3$$

指数映射关系为

$$T = \exp(\hat{\xi}) = \sum_{n=0}^{+\infty} \frac{\hat{\xi}^n}{n!} = \sum_{n=0}^{+\infty} \frac{1}{n!} \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}^n = \begin{bmatrix} \sum_{n=0}^{+\infty} \frac{1}{n!} \hat{w}^n & \sum_{n=0}^{+\infty} \frac{1}{(n+1)!} \hat{w}^n v \\ 0 & 1 \end{bmatrix}$$

同样有闭式解

$$T = \exp(\hat{\xi}) = \begin{bmatrix} \exp(\hat{w}) & J \cdot v \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & J \cdot v \\ 0 & 1 \end{bmatrix}$$

其中

$$J = \sum_{n=0}^{+\infty} \frac{1}{(n+1)!} \hat{w}^n = \frac{(I - \exp(\hat{w}))\hat{w}v + ww^T v}{|w|} = \frac{(I - R)\hat{w}v + ww^T v}{|w|}$$

即 $[w, v]$ 与 R, t 的关系为

$$R = \hat{w}, \quad t = Jv$$

2.6 相似变换、**Sim(3)**李群与其李代数的关系

$SE(3)$ 是刚体旋转和平移的变换，如果再考虑尺度缩放，即相似变换（similarity transform）的话，需要另外引入一个与尺度相关的自由度，构成**Sim(3)**群

相似变换的直观作用可以理解为，把一个刚体先移动 t 、再旋转 R 、再各向同性地压缩或者膨胀为 s 倍

Sim(3)中元素是

$$S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}, \quad R \in SO(3), s \in \mathbb{R}, t \in \mathbb{R}^3$$

相应的点坐标变换与 $SE(3)$ 类似，为

$$\dot{p}_c = S_{cw}\dot{p}_w, \quad p_c = sRp_w + t$$

同样参考 $SO(3)$ 和 $SE(3)$ 推导过程，假设 S 关于时间的表达形式 $S(\tau)$ ，其导数和逆矩阵相乘为

$$\begin{aligned} \dot{S} \cdot S^{-1} &= \begin{bmatrix} s\dot{R} & \dot{t} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{s}R^T & -\frac{1}{s}R^T t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{s}R + s\dot{R} & \dot{t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{s}R^T & -\frac{1}{s}R^T t \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{\dot{s}}{s}RR^T + \dot{R}R^T & \dot{t} - \frac{\dot{s}}{s}RR^T t - \dot{R}R^T t \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\dot{s}}{s}I + \hat{w} & \dot{t} - \frac{\dot{s}}{s}t - \hat{w}t \\ 0 & 0 \end{bmatrix} \end{aligned}$$

其中 $\hat{w} = \dot{R}R^T \in \mathfrak{so}(3)$ 。

令

$$\sigma = \frac{\dot{s}}{s} \in \mathbb{R}, \quad v = \dot{t} - \sigma t - \hat{w}t \in \mathbb{R}^3, \quad \zeta = \begin{bmatrix} w \\ v \\ \sigma \end{bmatrix} \in \mathbb{R}^7$$

则上式可以表示为

$$\dot{S} \cdot S^{-1} = \begin{bmatrix} \sigma I + \hat{w} & v \\ 0 & 0 \end{bmatrix} \stackrel{\text{def}}{=} \hat{\zeta}$$

这里的 **hat** 运算符定义为

$$\hat{\cdot}_{Sim(3)} : \zeta = [w, v, \sigma] \in \mathbb{R}^7 \rightarrow \hat{\zeta} = \begin{bmatrix} \sigma I + \hat{w} & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

同样，

$$\dot{S}(\tau) = \hat{\zeta} \cdot S(\tau)$$

因此指数映射为

$$S = \exp(\hat{\zeta}\tau)$$

把时间 τ 乘进 ζ 看做一个向量 ζ ，即

$$S = \exp(\hat{\zeta})$$

同样根据 Taylor 展开的形式

$$S = \exp(\hat{\zeta}) = \sum_{n=0}^{+\infty} \frac{\hat{\zeta}^n}{n!} = \sum_{n=0}^{+\infty} \frac{1}{n!} \begin{bmatrix} \sigma I + \hat{w} & v \\ 0 & 0 \end{bmatrix}^n$$

$$= \begin{bmatrix} \sum_{n=0}^{+\infty} \frac{1}{n!} (\sigma I + \hat{w})^n & \sum_{n=0}^{+\infty} \frac{1}{(n+1)!} (\sigma I + \hat{w})^n v \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \exp(\sigma I + \hat{w}) & J \cdot v \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^\sigma \cdot R & J \cdot v \\ 0 & 1 \end{bmatrix}$$

其中 J 为（参考[5]）

$$J = CI + \frac{A\sigma + (1-B)\theta}{\sigma^2 + \theta^2} \left(\frac{[w]_\times}{\theta} \right) + \left(C - \frac{(B-1)\sigma + A\theta}{\sigma^2 + \theta^2} \right) \left(\frac{[w]_\times}{\theta} \right)^2$$

$$A = e^\sigma \sin(\theta), B = e^\sigma \cos(\theta), C = \frac{e^\sigma - 1}{\sigma}, \theta = |w|$$

即 $[w, v, \sigma]$ 与 s, R, t 的关系为

$$s = e^\sigma, R = \exp(\hat{w}), t = J_{w,\sigma} v$$

2.7 李群与李代数关系总结

以 $SE(3)$ 为例，上面的例子都有如下形式

$$\dot{T}(\tau) = \hat{\xi} \cdot T(\tau)$$

对于 $T(\tau_0)$ 处，如果经过一个很小的时刻 $\delta\tau \rightarrow 0$ ，此时的位姿为

$$T(\tau_0 + \delta\tau) \approx T(\tau_0) + \dot{T}(\tau_0) \cdot \delta\tau = T(\tau_0) + \hat{\xi} \cdot T(\tau_0) \cdot \delta\tau$$

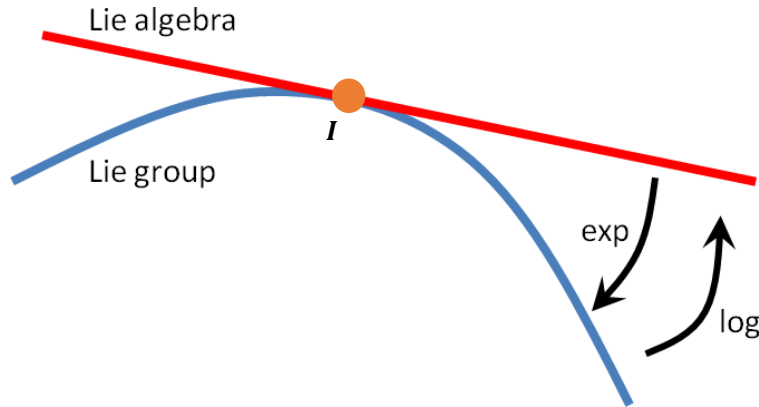
假设 0 时刻位姿为单位矩阵， $T(0) = I$ ，也就是说对于 $\tau_0 = 0$ 时，上式可以写成

$$T(\delta\tau) \approx T(0) + \hat{\xi} \cdot T(0) \cdot \delta\tau = I + \hat{\xi} \cdot \delta\tau$$

这个式子的右边，可以看成是在 $T(0) = I$ 处的切空间（tangent space）元素。也就是说，李代数 $\hat{\xi}$ 实际上是单位元素 I 处的切空间。

（ $\hat{\xi} \cdot \delta\tau$ 仍然是 $\mathfrak{se}(3)$ 中的元素。而在非单位元素处，切空间中元素是 $\hat{\xi} \cdot T(\tau_0) \delta\tau$ ，不是李代数的元素。）

整体李群与李代数之间的关系如下图，李代数是李群在单位元素处的切空间，李群元素与李代数对应元素之间满足指数映射的关系（逆映射-对数映射这里忽略，参考[1][4]）



3 李群的增量和导数表示

3.1 李群中的增量及相对于增量的导数

对于线性空间中的变量 x ，利用 Gauss-Newton 等迭代方法进行优化时，通常是计算出一个增量 Δx 并且叠加到原有状态上进行更新： $x \leftarrow x + \Delta x$

但是李群并不是线性空间，如果进行简单的矩阵加法，得到的元素不是在李群中。而矩阵李群对于乘法是封闭的，以 $T \in SE(3)$ 为例，也就是说对于 $\Delta T \in SE(3)$ ，则有 $T \cdot \Delta T \in SE(3)$ 以及 $\Delta T \cdot T \in SE(3)$ 。（分别对应右乘和左乘更新）

ΔT 中的元素数大于自由度(6DOF)，如果直接将其用于计算，计算 ΔT 时需要加入下面的约束

$$\Delta T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, R \in SO(3), t \in \mathbb{R}^3$$

显然这不太方便。

而李代数是一个向量空间，并且可以用最少的变量表示李群元素，那么如果在李代数中考虑，

$$T_0 = \exp(\xi_0), \quad T = \exp(\xi_0 + \Delta\xi)$$

这样也就是通常的向量空间中处理的形式。在左乘的定义下，对于小增量 $\Delta\xi \rightarrow 0$

$$T = \exp(\xi_0 + \Delta\xi) \approx \exp(\Delta\xi) \exp(\xi_0) = \exp(\Delta\xi) \cdot T_0$$

这里和实数的指数函数不同，是约等号，因为对于矩阵元素，满足 BCH 公式 (Baker-Campbell-Hausdorff formula, 略。参考[4]或 wikipedia)

$$\exp(X) \exp(Y) = \exp\left(X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}([X, [X, Y]] + [Y, [Y, X]]) \dots\right)$$

由于 $\Delta\xi$ 通常很小，因此后面的李括号的项 $[\cdot, \cdot]$ 可以忽略（未证明，按个人理解，从结果反推理由），可以得到上面的约等号。

因此，一般可以用李代数中的一个微小增量的指数映射来表示增量 ΔT ，即左乘定义下的状态更新表示为（也可以不考虑上面 BCH 相关的近似，直接按下面的方式定义状态更新）

$$T \leftarrow \exp(\hat{\xi}) \cdot T, \quad \xi \in \mathbb{R}^6$$

在这种定义下， T 关于 ξ 的导数为（以下推导参考[5]）

$$\left. \frac{\partial [\exp(\hat{\xi}) \cdot T]}{\partial \xi} \right|_{\xi=0}$$

对 $\xi = [\xi_1, \xi_2, \dots, \xi_6]$ 的每一维分别考虑，即

$$\left. \frac{\partial [\exp(\hat{\xi}) \cdot T]}{\partial \xi_k} \right|_{\xi=0} = \left. \frac{\partial [\exp(t\hat{e}_k) \cdot T]}{\partial t} \right|_{t=0}$$

其中 e_k 是第 k 维单位向量，如 $e_2 = [0, 1, 0, 0, 0, 0]$

类似实数指数函数

$$\frac{\partial \exp(ax)}{\partial x} = a \cdot \exp(ax)$$

对 $\exp(t\hat{e}_k)$ 也类似，有

$$\left. \frac{\partial [\exp(t\hat{e}_k) \cdot T]}{\partial t} \right|_{t=0} = [\hat{e}_k \cdot \exp(t\hat{e}_k) \cdot T] \big|_{t=0} = \hat{e}_k \cdot T \stackrel{\text{def}}{=} G_k T$$

其中 $G_k = \hat{e}_k$ 是 $\mathfrak{se}(3)$ 的 generators。

显然根据定义

$$G_k = \hat{e}_k \in \mathbb{R}^{4 \times 4}, \quad \begin{bmatrix} \hat{w} \\ v \end{bmatrix} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix}$$

（这里 $\mathfrak{se}(3)$ 中顺序是先 3DOF 旋转、后 3DOF 平移。若两者顺序交换，推导也调整即可。）

有

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

对某一维的导数 $G_k T$ 是矩阵，把所有维的组合（堆叠）起来是一个张量

$$\frac{\partial[\exp(\xi) \cdot T]}{\partial \xi} \Big|_{\xi=0} = \left[\frac{\partial[\exp(\xi) \cdot T]}{\partial \xi_1} \Big|_{\xi=0}, \frac{\partial[\exp(\xi) \cdot T]}{\partial \xi_2} \Big|_{\xi=0}, \dots \right]$$

$$= [G_1, G_2, G_3, G_4, G_5, G_6]T$$

结果实际上是由 6 个矩阵 $G_k T$ “堆”起来的 $4 \times 4 \times 6$ 的张量（这里表示方式可能不太正规，用于理解即可）

3.2 点坐标变换关于李代数增量的导数

(1) 考虑点坐标变换情况，令 $\dot{x} = T\dot{y} = [x; 1]$ ，则

$$\frac{\partial[\exp(\xi) \cdot T\dot{y}]}{\partial \xi} \Big|_{\xi=0} = [G_1, G_2, G_3, G_4, G_5, G_6]T\dot{y} = [G_1\dot{x}, G_2\dot{x}, G_3\dot{x}, G_4\dot{x}, G_5\dot{x}, G_6\dot{x}]$$

$$= \begin{bmatrix} -\hat{x} & I_{3 \times 3} \\ 0 & 0 \end{bmatrix}$$

（张量与 4×1 维齐次点坐标向量 \dot{y} 相乘，从 $4 \times 4 \times 6$ 张量变为了 4×6 矩阵的形式。）

(2) 对于点坐标变换的情况，有另一种简单的方式推导。

令 $T\dot{y} = \dot{x}$ ，对于很小的增量 $\xi \rightarrow 0$ ，有

$$\exp(\xi) \approx I + \hat{\xi}$$

则

$$\frac{\partial[\exp(\xi) \cdot T\dot{y}]}{\partial \xi} \Big|_{\xi=0} = \frac{\partial[\exp(\xi) \cdot \dot{x}]}{\partial \xi} \Big|_{\xi=0}$$

$$\approx \frac{\partial[(I + \hat{\xi}) \cdot \dot{x}]}{\partial \xi} \Big|_{\xi=0} = \frac{\partial[\hat{\xi}\dot{x}]}{\partial \xi} \Big|_{\xi=0}$$

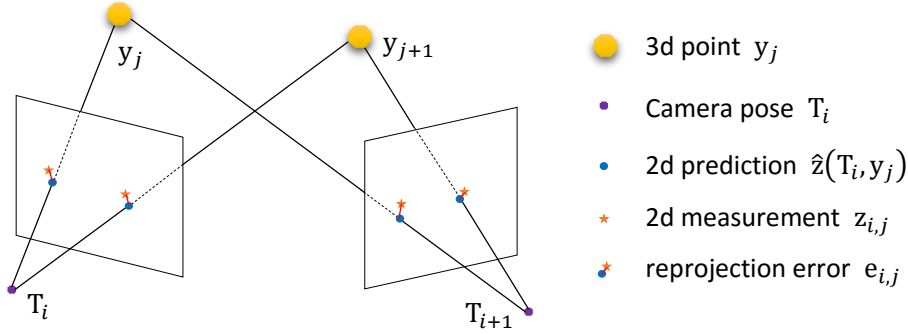
$$= \frac{\partial \left[\begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ 1 \end{bmatrix} \right]}{\partial \xi} \Big|_{\xi=[w;v]=0} = \frac{\partial \begin{bmatrix} \hat{w}x + v \\ 0 \end{bmatrix}}{\partial [w, v]} \Big|_{\xi=[w;v]=0}$$

$$= \frac{\partial \begin{bmatrix} -\hat{x}w + v \\ 0 \end{bmatrix}}{\partial [w, v]} = \begin{bmatrix} -\hat{x} & I_{3 \times 3} \\ 0 & 0 \end{bmatrix}$$

倒数第二个等号利用了 $\hat{w}x = -\hat{x}w$ 的性质。得到的结果和上面(1)一样。

4 李群在视觉 SLAM 中的应用

4.1 重投影误差（reprojection error）的导数



假设有相机位姿 $T_{i=1,\dots,m}$ 和 3D 特征点 $y_{j=1,\dots,n}$ ，相机 T_i 的图像中看到 y_j 所在位置为 $z_{i,j}$ ，而根据相机模型将 y_j 投影在 T_i 中的预测位置为 $\hat{z}(T_i, y_j)$ ，这两个位置间残差（重投影误差）为 $e_{i,j}$

$$e_{i,j} := z_{i,j} - \hat{z}(T_i, y_j)$$

相机投影模型为

$$\hat{z}(x) = \text{proj}(x) = K \cdot \frac{1}{x_3} x, \quad x = [x_1, x_2, x_3]^T \in \mathbb{R}^3, \quad K = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix}$$

对于 y_j 在 T_i 中的坐标 $x_{i,j}$ 有

$$x_{i,j} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 1} \end{bmatrix} \cdot T_i \dot{y}_j, \quad \dot{y}_j = \begin{bmatrix} y_j \\ 1 \end{bmatrix}$$

因此 y_j 在 T_i 图像中的观测位置为

$$\hat{z}(T_i, y_j) = \text{proj}(\begin{bmatrix} I_{3 \times 3} & 0_{3 \times 1} \end{bmatrix} \cdot T_i \dot{y}_j)$$

这个估计位置与图像中测量点位置的残差为

$$e_{i,j}(T_i, y_j) = z_{i,j} - \hat{z}(T_i, y_j)$$

重投影误差关于相机位姿的 Jacobian 为

$$\frac{\partial e_{i,j}(\exp(\xi_i) T_i, y_j)}{\partial \xi_i} \Big|_{\xi_i=0} = - \frac{\partial \hat{z}(x)}{\partial x} \Big|_{x=\begin{bmatrix} I_{3 \times 3} & 0_{3 \times 1} \end{bmatrix} \cdot T_i \dot{y}_j} \cdot \frac{\partial x(\exp(\xi_i) T_i, y_j)}{\partial \xi_i} \Big|_{\xi_i=0}$$

其中

$$\frac{\partial \hat{z}(x)}{\partial x} = \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix}$$

$$\frac{\partial x(\exp(\xi_i) T_i, y_j)}{\partial \xi_i} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 1} \end{bmatrix} \cdot \frac{\partial \exp(\xi_i) T_i \cdot \dot{y}_j}{\partial \xi_i} \Big|_{\xi_i=0}$$

根据上面 3.2 的推导，令 $\dot{x}_{i,j} = T_i \cdot \dot{y}_j$ ，有

$$\frac{\partial \exp(\xi_i) T_i \cdot \dot{y}_j}{\partial \xi_i} \Big|_{\xi_i=0} = \begin{bmatrix} -\hat{x}_{i,j} & I_{3 \times 3} \\ 0 & 0 \end{bmatrix} \Rightarrow \frac{\partial x(\exp(\xi_i) T_i, y_j)}{\partial \xi_i} = \begin{bmatrix} -\hat{x}_{i,j} & I_{3 \times 3} \end{bmatrix}$$

因此重投影误差关于位姿的 Jacobian 为（实际上，就是投影位置 \hat{z} 的 Jacobian 的相反数）

$$\frac{\partial e_{i,j}(\exp(\xi_i) T_i, y_j)}{\partial \xi_i} \Big|_{\xi_i=0} = - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot \begin{bmatrix} -\hat{x} & I_{3 \times 3} \end{bmatrix}$$

$$x = [x_1, x_2, x_3]^T = R_i y_j + t_i$$

而重投影误差关于 3D 点位置 y_j 的 Jacobian 很容易算出

$$\frac{\partial e_{i,j}(T_i, y_j)}{\partial y_j} = -\frac{\partial \hat{z}(x)}{\partial x} \cdot \frac{\partial x(T_i, y_j)}{\partial y_j} = -\frac{\partial \hat{z}(x)}{\partial x} \cdot \frac{\partial (R_i y_j + t_i)}{\partial y_j}$$

即

$$\frac{\partial e_{i,j}(T_i, y_j)}{\partial y_j} = -\begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot R_i, \quad x = [x_1, x_2, x_3]^T = R_i y_j + t_i$$

其中 R_i 是位姿 T_i 中的旋转矩阵

$$T_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix}$$

4.2 光束平差法 (BA, Bundle Adjustment)

Bundle Adjustment 是求解一个优化问题：调整相机位姿 $T_{i=1,\dots,m}$ 和 3D 特征点 $y_{j=1,\dots,n}$ ，使重投影误差之和最小。（按个人理解，因为调整相机位姿和 3D 点，相当于在调整所有连接相机和 3D 点之间的光线束 **bundle**，所以名字叫 **Bundle Adjustment**）

这个优化问题即

$$\begin{aligned} \min_{T_{i=1,\dots,m}, y_{j=1,\dots,n}} C(T_{i=1,\dots,m}, y_{j=1,\dots,n}) \\ C(T_{i=1,\dots,m}, y_{j=1,\dots,n}) &= \sum_{i,j} e_{i,j}^T W_{i,j} e_{i,j} = e^T W e \\ &= [e_{1,1}^T \quad \dots \quad e_{i,j}^T \quad \dots] W \begin{bmatrix} e_{1,1} \\ \vdots \\ e_{i,j} \\ \vdots \end{bmatrix} \end{aligned}$$

为了用 Gauss-Newton 类的方法求解这个问题，需要计算 e 的 Jacobian（以下参考[5]）

$$J = [J_T \quad J_y]$$

其中

$$\begin{aligned} J_T &:= \left[\frac{\partial e(\exp(\hat{\xi}_1) \cdot T_1, \dots, T_m, y_1, \dots, y_n)}{\partial \xi_1} \Big|_{\xi_1=0} \quad \dots \quad \frac{\partial e(T_1, \dots, \exp(\hat{\xi}_m) \cdot T_m, y_1, \dots, y_n)}{\partial \xi_m} \Big|_{\xi_m=0} \right] \\ J_y &:= \left[\frac{\partial e(T_1, \dots, T_m, y_1, \dots, y_n)}{\partial y_1} \quad \dots \quad \frac{\partial e(T_1, \dots, T_m, y_1, \dots, y_n)}{\partial y_n} \right] \end{aligned}$$

是误差向量 e 关于位姿和 3D 点的 Jacobian。

用 Gauss-Newton 法，计算增量 $\Delta x = [\xi_1, \dots, \xi_m, \Delta y_1, \dots, \Delta y_n]$ 的方式为

$$J^T W J \cdot \Delta x = -J^T W e$$

算出 Δx 后，状态更新的方式为

$$T_i \leftarrow \exp(\hat{\xi}_i) \cdot T_i, \quad y_j \leftarrow y_j + \Delta y_j$$

注意 J_T 中每一项的形式，例如

$$\frac{\partial e(\dots, \exp(\hat{\xi}_i) \cdot T_i, \dots, T_m, y_1, \dots, y_n)}{\partial \xi_i} \Big|_{\xi=0} = \begin{bmatrix} \frac{\partial e_{1,1}(\dots, \exp(\hat{\xi}_i) \cdot T_i, \dots, T_m, y_1, \dots, y_n)}{\partial \xi_i} \\ \vdots \\ \frac{\partial e_{i,j}(\dots, \exp(\hat{\xi}_i) \cdot T_i, \dots, T_m, y_1, \dots, y_n)}{\partial \xi_i} \\ \vdots \end{bmatrix}$$

其中，只有与 T_i 有关的误差项 $e_{i,x}$ 的 Jacobian 才非零，其他都为 0，因此计算 J_T 只需要计算 T_i 观测 $y_{j=1,\dots,n_i}$ 的所有重投影误差 $e_{i,x}$ 相关的项。

而 $e_{i,j}$ 只与 T_i 和 y_j 有关，上面已经计算出其结果为

$$\begin{aligned} \frac{\partial e_{i,j}(\cdots, \exp(\hat{\xi}_i) \cdot T_i, \dots, T_m, y_1, \dots, y_n)}{\partial \xi_i} \Big|_{\xi_i=0} &= \frac{\partial e_{i,j}(\exp(\hat{\xi}_i) T_i, y_j)}{\partial \xi_i} \Big|_{\xi_i=0} \\ &= - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot [-\hat{x}_{i,j} \quad I_{3 \times 3}] \end{aligned}$$

关于 J_y 的计算同理，将 J_y 中的非零项

$$\frac{\partial e_{i,j}(T_1, \dots, T_m, \dots, y_j, \dots, y_n)}{\partial y_j} = \frac{\partial e_{i,j}(T_i, y_j)}{\partial y_j} = - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot R_i$$

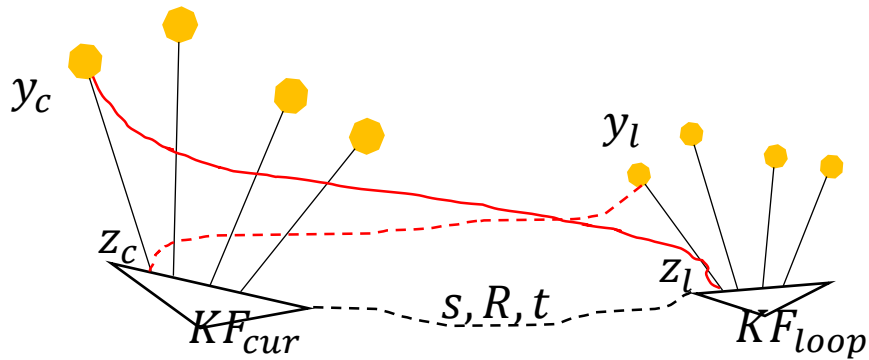
算出即可。

（这里看出，由于不是所有 T_i 都可以观测到所有 y_j ，Jacobian 中观测不到的那些 i,j 位置，对应项是 0，因此 Jacobian 矩阵和 Hessian 矩阵 $J^T W J$ 都是稀疏的，可以利用稀疏性进行高效计算。具体方式采用 Schur 补或者 Cholesky 分解，参见[4]或[6]）

4.3 利用最小化重投影误差来计算相似变换

在单目 SLAM 中，由于缺乏绝对尺度信息，误差的积累导致可能会发生尺度漂移^[7]。导致的结果是，单目 SLAM 相机经过一段时间运动，回到之前出现过的位置并检测到闭环时，之前地图中的特征点和当前帧所建地图中的特征点，尺度不一样了。

如下图，ORB_SLAM^[8]的闭环检测过程中，通过 BoW 方法，检测到和当前关键帧 KF_{cur} 场景相似的、之前出现过的关键帧 KF_{loop} 。当前所建局部地图中的 4 个点，和之前地图中已有的 4 个点通过 BoW 进行了匹配，即这四个点实际上是对应真实环境中同样的 4 个点。但是由于计算误差累积导致的尺度漂移， KF_{cur} 处的局部地图类似于 KF_{loop} 处地图的“膨胀版”，也就是两者尺度不一样。



在 ORB_SLAM 中，首先通过 closed-form 方式用线性方法 svd、在 RANSAC 框架下算出一组初始解 $s/R/t$ ，然后利用重投影误差最小化的方式来优化两个关键帧及地图点之间的这个 $s/R/t$ 相似变换。

假设两者之间，点坐标的 Sim(3)变换矩阵为

$$\dot{y}_c = S_{cl} \dot{y}_l, \quad S_{cl} = \begin{bmatrix} S_{cl} R_{cl} & t_{cl} \\ 0 & 1 \end{bmatrix}, \quad S_{cl} \in \mathbb{R}, R_{cl} \in SO(3), t_{cl} \in \mathbb{R}^3$$

非齐次坐标形式为

$$y_c = s_{cl} R_{cl} y_l + t_{cl}$$

反过来的点坐标变换及相似变换的逆矩阵为

$$\dot{y}_l = S_{lc} \dot{y}_c, \quad S_{lc} = S_{cl}^{-1} = \begin{bmatrix} \frac{1}{s_{cl}} R_{cl}^T & -\frac{1}{s_{cl}} R_{cl}^T t_{cl} \\ 0 & 1 \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} s_{lc} R_{lc} & t_{lc} \\ 0 & 1 \end{bmatrix}$$

其中

$$s_{lc} = \frac{1}{s_{cl}}, \quad R_{lc} = R_{cl}^T, \quad t_{lc} = -\frac{1}{s_{cl}} R_{cl}^T t_{cl}$$

ORB_SLAM 是这样考虑的 (OptimizeSim3 函数): 将 y_l 通过 S_{cl} 变换到 KFcur 的坐标并进行投影, 预测点 \hat{z}_c 与 y_c 的观测点 z_c 之间的残差为

$$e_c = z_c - \hat{z}_c = z_c - \text{proj}(s_{cl} R_{cl} y_l + t_{cl})$$

同样的, 将 y_c 投影到 KFcur 的预测点 \hat{z}_l , 也有残差项

$$e_l = z_l - \hat{z}_l = z_l - \text{proj}(s_{lc} R_{lc} y_c + t_{lc})$$

通过调整 $S_{cl} \in \text{Sim}(3)$, 使这些残差之和最小。

这和 Bundle Adjustment 一样, 只是每个实际的 3D 点对应两个不同尺度的地图点, 上面的两个残差是 g2o 优化的两条边, 而且这两条边对应同一个被优化量 S_{cl} , 只是其中一边是它的逆 $S_{lc} = S_{cl}^{-1}$ 。(两条边在 ORB_SLAM 中是 EdgeSim3ProjectXYZ、EdgeInverseSim3ProjectXYZ)

(0) 首先看 e_c 关于 y_l 和 e_l 关于 y_c 的 Jacobian

• e_c 关于 y_l 的 Jacobian

$$\begin{aligned} \frac{\partial e_c(S_{cl}, y_l)}{\partial y_l} &= - \frac{\partial \hat{z}_c(x)}{\partial x} \Big|_{x=s_{cl} R_{cl} y_l + t_{cl}} \cdot \frac{\partial x(S_{cl}, y_l)}{\partial y_l} \Big|_{x=s_{cl} R_{cl} y_l + t_{cl}} \\ &= - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot s_{cl} R_{cl} \end{aligned}$$

其中

$$x = [x_1, x_2, x_3]^T = s_{cl} R_{cl} y_l + t_{cl}, \quad S_{cl} = \begin{bmatrix} s_{cl} R_{cl} & t_{cl} \\ 0 & 1 \end{bmatrix}$$

• e_l 关于 y_c 的 Jacobian

$$\begin{aligned} \frac{\partial e_l(S_{lc}, y_c)}{\partial y_c} &= - \frac{\partial \hat{z}_l(x)}{\partial x} \Big|_{x=s_{lc} R_{lc} y_c + t_{lc}} \cdot \frac{\partial x(S_{lc}, y_c)}{\partial y_c} \Big|_{x=s_{lc} R_{lc} y_c + t_{lc}} \\ &= - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot s_{lc} R_{lc} \end{aligned}$$

其中 (注意下面的 x 与上面的 x 不同)

$$x = [x_1, x_2, x_3]^T = s_{lc} R_{lc} y_c + t_{lc}, \quad S_{lc} = \begin{bmatrix} s_{lc} R_{lc} & t_{lc} \\ 0 & 1 \end{bmatrix} = S_{cl}^{-1}$$

$$s_{lc} = \frac{1}{s_{cl}}, \quad R_{lc} = R_{cl}^T, \quad t_{lc} = -\frac{1}{s_{cl}} R_{cl}^T t_{cl}$$

(1) e_c 关于 S_{cl} 增量的 Jacobian, 和 $SE(3)$ 中一样, 这里也是用李代数中小增量来进行状态更新, 加上增量后的相似变换为

$$S_{cl}(\zeta) \stackrel{\text{def}}{=} \exp(\hat{\zeta}) S_{cl}$$

Jacobian 为

$$\left. \frac{\partial e_c(S_{cl}(\zeta), y_l)}{\partial \zeta} \right|_{\zeta=0} = - \left. \frac{\partial \hat{z}_c(x)}{\partial x} \right|_{x=S_{cl}R_{cl}y_l+t_{cl}} \cdot \left. \frac{\partial x(\exp(\hat{\zeta}) S_{cl}, y_l)}{\partial \zeta} \right|_{\zeta=0}$$

其中 $\zeta = [w; v; \sigma] \in \mathbb{R}^7, w, v \in \mathbb{R}^3, \sigma \in \mathbb{R}$

$$\left. \frac{\partial x(\exp(\hat{\zeta}) S_{cl}, y_l)}{\partial \zeta} \right|_{\zeta=0} = [I_{3 \times 3} \quad 0_{3 \times 1}] \cdot \left. \frac{\partial \exp(\hat{\zeta}) S_{cl} \cdot y_l}{\partial \zeta} \right|_{\zeta=0}$$

利用

$$\hat{\zeta} = \begin{bmatrix} \sigma I + \hat{w} & v \\ 0 & 0 \end{bmatrix}$$

可以算出, $\mathfrak{sim}(3)$ 的李代数的 generator 前六个与 $\mathfrak{se}(3)$ 的相同, 第7个是

$$G_7 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

那么令 $\dot{x} = S_{cl} \cdot \dot{y}_l$, 与 $SE(3)$ 同理有

$$\begin{aligned} \left. \frac{\partial [\exp(\hat{\zeta}) S_{cl} \cdot \dot{y}_l]}{\partial \zeta} \right|_{\zeta=0} &= [G_1, G_2, G_3, G_4, G_5, G_6, G_7] S_{cl} \dot{y}_l = [G_1 \dot{x}, G_2 \dot{x}, G_3 \dot{x}, G_4 \dot{x}, G_5 \dot{x}, G_6 \dot{x}, G_7 \dot{x}] \\ &= \begin{bmatrix} -\hat{x} & I_{3 \times 3} & x \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

代入得

$$\left. \frac{\partial e_c(S_{cl}(\zeta), y_l)}{\partial \zeta} \right|_{\zeta=0} = - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot \begin{bmatrix} -\hat{x} & I_{3 \times 3} & x \end{bmatrix}$$

其中 $x = [x_1, x_2, x_3]^T = S_{cl}R_{cl}y_l + t_{cl}$ 。

上式化简可得

$$\left. \frac{\partial e_c(S_{cl}(\zeta), y_c)}{\partial \zeta} \right|_{\zeta=0} = \begin{bmatrix} -\frac{f_u}{x_3} & 0 & \frac{x_1 f_u}{x_3^2} & \frac{x_1 x_2}{x_3^2} f_u & -\left(1 + \frac{x_1^2}{x_3^2}\right) f_u & \frac{x_2}{x_3} f_u & 0 \\ 0 & -\frac{f_v}{x_3} & \frac{x_2 f_v}{x_3^2} & \left(1 + \frac{x_2^2}{x_3^2}\right) f_v & -\frac{x_1 x_2}{x_3^2} f_v & -\frac{x_1}{x_3} f_v & 0 \end{bmatrix}$$

可以看到, 这部分的 Jacobian 和尺度无关。

(3) 再看 e_l 关于 S_{cl} 增量 (注意不是 S_{lc}) 的 Jacobian。 S_{cl} 加上增量之后, S_{lc} 是

$$S_{lc}: (\exp(\hat{\zeta}) S_{cl})^{-1} = S_{cl}(\zeta)^{-1}$$

e_l 关于 S_{cl} 的 Jacobian

$$\left. \frac{\partial e_l(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \right|_{\zeta=0} = - \left. \frac{\partial \hat{z}_l(x)}{\partial x} \right|_{x=S_{lc}R_{lc}y_c+t_{lc}} \cdot \left. \frac{\partial x(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \right|_{\zeta=0}$$

右边最后一项为

$$\left. \frac{\partial x(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \right|_{\zeta=0} = [I_{3 \times 3} \quad 0_{3 \times 1}] \cdot \left. \frac{\partial S_{cl}(\zeta)^{-1} \cdot y_c}{\partial \zeta} \right|_{\zeta=0}$$

根据矩阵的求导公式，有

$$\frac{\partial A^{-1}}{\partial a} = -A^{-1} \frac{\partial A}{\partial a} A^{-1}$$

所以上面右边最后一项可以化简为

$$\begin{aligned} \frac{\partial S_{cl}(\zeta)^{-1} \cdot \dot{y}_c}{\partial \zeta} \Big|_{\zeta=0} &= - \left[S_{cl}(\zeta)^{-1} \frac{\partial S_{cl}(\zeta)}{\partial \zeta} S_{cl}(\zeta)^{-1} \cdot \dot{y}_c \right] \Big|_{\zeta=0} \\ &= - \left[S_{lc} \cdot \frac{\partial S_{cl}(\zeta)}{\partial \zeta} \cdot S_{lc} \cdot \dot{y}_c \right] \Big|_{\zeta=0} \end{aligned}$$

根据上面(1)的推导

$$\frac{\partial [S_{cl}(\zeta) \cdot \dot{a}]}{\partial \zeta} \Big|_{\xi=0} = [G_1, G_2, G_3, G_4, G_5, G_6, G_7] S_{cl} \dot{a}$$

因此

$$\begin{aligned} \frac{\partial S_{cl}(\zeta) \cdot S_{lc} \dot{y}_c}{\partial \zeta} \Big|_{\zeta=0} &= [G_1, G_2, G_3, G_4, G_5, G_6, G_7] S_{cl} \cdot S_{lc} \dot{y}_c \\ &= [G_1, G_2, G_3, G_4, G_5, G_6, G_7] \dot{y}_c = \begin{bmatrix} -\hat{y}_c & I_{3 \times 3} & y_c \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

所以

$$\frac{\partial S_{cl}(\zeta)^{-1} \cdot \dot{y}_c}{\partial \zeta} \Big|_{\zeta=0} = -S_{lc} \cdot \begin{bmatrix} -\hat{y}_c & I_{3 \times 3} & y_c \\ 0 & 0 & 0 \end{bmatrix}$$

代入残差 e_l 整体的 Jacobian 中，有

$$\begin{aligned} \frac{\partial e_l(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \Big|_{\zeta=0} &= - \frac{\partial \hat{z}_c(x)}{\partial x} \Big|_{x=S_{lc}R_{lc}y_c+t_{lc}} \cdot [I_{3 \times 3} \quad 0_{3 \times 1}] \cdot \left(-S_{lc} \cdot \begin{bmatrix} -\hat{y}_c & I_{3 \times 3} & y_c \\ 0 & 0 & 0 \end{bmatrix} \right) \\ &= - \frac{\partial \hat{z}_c(x)}{\partial x} \Big|_{x=S_{lc}R_{lc}y_c+t_{lc}} \cdot \left([I_{3 \times 3} \quad 0_{3 \times 1}] \cdot \left(- \begin{bmatrix} S_{lc}R_{lc} & t_{lc} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\hat{y}_c & I_{3 \times 3} & y_c \\ 0 & 0 & 0 \end{bmatrix} \right) \right) \\ &= - \frac{\partial \hat{z}_c(x)}{\partial x} \Big|_{x=S_{lc}R_{lc}y_c+t_{lc}} \cdot S_{lc}R_{lc} \cdot [\hat{y}_c \quad -I_{3 \times 3} \quad -y_c] \end{aligned}$$

注意到(0)中求出 e_l 关于 y_c 的 Jacobian 的形式和前面几项一致

$$J_{y_c}^{e_l} \stackrel{\text{def}}{=} \frac{\partial e_l(S_{lc}, y_c)}{\partial y_c} = - \frac{\partial \hat{z}_l(x)}{\partial x} \Big|_{x=S_{lc}R_{lc}y_l+t_{lc}} \cdot S_{lc}R_{lc}$$

所以算出 $J_{y_c}^{e_l}$ 后可以直接代入上式，有

$$\frac{\partial e_l(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \Big|_{\zeta=0} = J_{y_c}^{e_l} \cdot [\hat{y}_c \quad -I_{3 \times 3} \quad -y_c]$$

(这里 ζ 定义的顺序是 $\zeta = [w; v; \sigma]$ ，也就是先 3DOF 旋转、再 3DOF 平移、再 1DOF 尺度。实际程序中根据定义顺序的不同进行矩阵块位置的调整。)

这部分 Jacobian 最后化简的 2x7 矩阵中，关于尺度的项不是 0。所以整个优化问题中，尺度 s 和旋转 R 、平移 t 一样，可以被优化。下面对此进行整理说明。

Jacobian 的形式

$$\frac{\partial e_l(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \Big|_{\zeta=0} = - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot S_{lc}R_{lc} \cdot [\hat{y}_c \quad -I_{3 \times 3} \quad -y_c]$$

令 $x = [x_1, x_2, x_3]^T = S_{lc}R_{lc}y_c + t_{lc} \Rightarrow t_{lc} - x = -S_{lc}R_{lc}y_c$ ，因此

$$\begin{aligned}\frac{\partial e_l(S_{cl}(\zeta)^{-1}, y_c)}{\partial \zeta} \Big|_{\zeta=0} &= - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} [s_{lc} R_{lc} \cdot \hat{y}_c \quad -s_{lc} R_{lc} \quad -s_{lc} R_{lc} y_c] \\ &= - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} [s_{lc} R_{lc} \cdot \hat{y}_c \quad -s_{lc} R_{lc} \quad t_{lc} - x]\end{aligned}$$

其中最后一列关于尺度的项为

$$- \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot (t_{lc} - x) = - \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} t_{lc} = \begin{bmatrix} \frac{f_u t_1}{x_3} - \frac{f_u x_1 t_3}{x_3^2} \\ \frac{f_v t_2}{x_3} - \frac{f_v x_2 t_3}{x_3^2} \end{bmatrix}$$

其中

$$t_{lc} = [t_1, t_2, t_3]^T$$

4.4 光测量误差（photometric error）最小化

在 LSDSLAM 中，通过匹配参考关键帧的逆深度地图，来获得当前图像的位姿。具体的，是利用 Lucas-Canade 方法，最小化像素值的误差（photometric error，也翻译成光测误差）。优化目标如下（LSDSLAM 采用 forwards-additive 形式，参考[9]。也有其他不同形式）

$$\min \sum_{\mathbf{x}} \left(I(\mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T})) - \mathcal{T}(\mathbf{x}) \right)^2$$

其中 I 是当前图像， \mathcal{T} 是模板图像， $\mathbf{W}(\cdot)$ 是 Warping function。 \mathbf{T} 在 VSLAM 中是 $SE(3)$ 或 $Sim(3)$ 变换的参数——相机位姿， $D(\mathbf{x})$ 是 \mathcal{T} （不是 I ）相机坐标系中、像素位置 \mathbf{x} 处的逆深度。（计算所有像素梯度大于一定阈值的 \mathbf{x} 处的误差并累加）

上式中 $I(\mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T}))$ 的含义，是对于模板图像 \mathcal{T} 中的位置 \mathbf{x} ，经过变换对应到当前图像 I 的位置 z 处

$$z = \mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T})$$

根据相机的模型，这个 Warping 函数可以写成[10]

$$z = \mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T}) = \pi \left(\mathbf{T}, \pi^{-1} \left(\mathbf{x}, \frac{1}{D(\mathbf{x})} \right) \right) \stackrel{\text{def}}{=} \pi(\mathbf{T}, \mathbf{y})$$

其中 $\pi(\cdot)$ 是投影函数， $\mathbf{y} \in \mathbb{R}^3$ 是 \mathcal{T} 对应相机坐标系下的三维位置

$$\mathbf{y} \stackrel{\text{def}}{=} \pi^{-1} \left(\mathbf{x}, \frac{1}{D(\mathbf{x})} \right) = \mathbf{K}^{-1} \cdot \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \cdot \frac{1}{D(\mathbf{x})}$$

因此 \mathbf{x} 处的像素误差为

$$e(\mathbf{x}, \mathbf{T}) = I(\mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T})) - \mathcal{T}(\mathbf{x}) = I(z) - \mathcal{T}(\mathbf{x})$$

类似 Bundle Adjustment，优化计算 \mathbf{T} 时，需要计算出误差关于位姿 \mathbf{T} 的雅克比。明显只有 $I(z)$ 与位姿 \mathbf{T} 有关，计算雅克比时也只需要考虑这部分，根据链式法则，有如下形式（只是为了分成两个部分，形式上的表示，不是严格的数学表示）

$$J_T = \frac{\partial e}{\partial \mathbf{T}} = \frac{\partial I(z)}{\partial z} \cdot \frac{\partial z}{\partial \mathbf{T}}$$

- 第一部分就是图像 I 中位置 z 处的像素梯度

$$\frac{\partial I(z)}{\partial z} = \nabla I(z) \Big|_{z=\mathbf{W}(\mathbf{x}, D(\mathbf{x}), \mathbf{T})=\pi(\mathbf{T}, \mathbf{y})} = [\nabla I_u, \nabla I_v] \Big|_{z=[u,v]=\pi(\mathbf{T}, \mathbf{y})}$$

- 下面考虑第二部分——投影位置 z 关于位姿的 Jacobian:

$$\frac{\partial z}{\partial T}$$

根据上文 4.1, 投影位置 z 表示为

$$z = \pi(T, y) = \text{proj}([I_{3 \times 3} \quad 0_{3 \times 1}] \cdot T \cdot \dot{y})$$

其中 \dot{y} 是齐次坐标 $\dot{y} = [y; 1]$

LSDSLAM 中, 有 SE3Tracking 和 Sim3Tracking, 两者 Jacobian 的形式不同。根据上文 4.1 和 4.3 的推导, 对SE(3)和Sim(3)分别有:

- a) SE(3)——投影位置 z 关于位姿 $T \in SE(3)$ 的 Jacobian 为

$$\left. \frac{\partial z(\exp(\xi) \cdot T \cdot \dot{y})}{\partial \xi} \right|_{\xi=0} = \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot [-\hat{x} \quad I_{3 \times 3}]$$

其中 $\dot{x} = T_{SE(3)} \cdot \dot{y}$, 写成非齐次坐标形式

$$x = [x_1, x_2, x_3]^T = R y + t, \quad T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE(3), \quad y = K^{-1} \cdot \begin{bmatrix} x \\ 1 \end{bmatrix} \cdot \frac{1}{D(x)}$$

- b) Sim(3)——投影位置 z 关于位姿 $T \in Sim(3)$ 的 Jacobian 为

$$\left. \frac{\partial z(\exp(\xi) \cdot T \cdot \dot{y})}{\partial \xi} \right|_{\xi=0} = \begin{bmatrix} \frac{f_u}{x_3} & 0 & -\frac{f_u x_1}{x_3^2} \\ 0 & \frac{f_v}{x_3} & -\frac{f_v x_2}{x_3^2} \end{bmatrix} \cdot [-\hat{x} \quad I_{3 \times 3} \quad x]$$

其中 $\dot{x} = T_{Sim(3)} \cdot \dot{y}$, 写成非齐次坐标形式

$$x = [x_1, x_2, x_3]^T = s R y + t, \quad T = \begin{bmatrix} s R & t \\ 0 & 1 \end{bmatrix} \in Sim(3), \quad y = K^{-1} \cdot \begin{bmatrix} x \\ 1 \end{bmatrix} \cdot \frac{1}{D(x)}$$

5 其他

5.1 欧拉角与SO(3)的关系

假设偏航-俯仰-滚转顺序的欧拉角为 $[\psi, \theta, \phi]$, 则旋转矩阵可以表示为

$$R = R_\phi R_\theta R_\psi \in SO(3)$$

根据定义, 偏航是绕 $[0,0,1]^T$ 旋转, 即

$$R_\psi = \exp \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix}, \quad R_\theta = \exp \begin{bmatrix} 0 \\ \theta \\ 0 \end{bmatrix}, \quad R_\phi = \exp \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix}$$

即

$$R = \exp \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix} \cdot \exp \begin{bmatrix} 0 \\ \theta \\ 0 \end{bmatrix} \cdot \exp \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix}$$

参考:

- [1] TUM 的 Multiple View Geometry 公开课 https://www.youtube.com/playlist?list=PLTBdjV_4f-EJn6udZ34t9EVIW7lbeo4
- [2] 高翔（清华大学）博客 <http://www.cnblogs.com/gaoxiang12/p/5137454.html>
- [3] 贺一家（中科院自动化所）博客 <http://blog.csdn.net/heyijia0327/article/details/50446140>
- [4] Barfoot, T. D. (2016). *State Estimation for Robotics: A Matrix-Lie-Group Approach*
- [5] Strasdat, H. (2012). *Local accuracy and global consistency for efficient visual slam* (Doctoral dissertation, Department of Computing, Imperial College London).
- [6] Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.
- [7] Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2010, June). Scale Drift-Aware Large Scale Monocular SLAM. In *Robotics: Science and Systems*(Vol. 2, No. 3, p. 5).
- [8] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *Robotics, IEEE Transactions on*, 31(5), 1147-1163.
- [9] Baker, S., & Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3), 221-255.
- [10] T Schöps. (2014). Semi-dense visual SLAM on mobile devices (Master thesis). https://vision.in.tum.de/_media/spezial/bib/schoeps14msc.pdf