

# ICP 算法整理

## 一. 算法简介

对于没有描述子信息只有三维坐标信息的点云而言，无论是求取被测物体/场景几何信息的三维重建，还是求取两组点云之间的旋转平移的导航定位，抑或是同时追求定位和建图的SLAM，将不同视角不同坐标系下的点云统一到同一坐标系下都是一个十分重要的问题。解决这一问题的主要思路就是进行点云配准，而点云配准中，最常用的算法就是ICP (Iterative Closest Point，迭代最近点)。

ICP 算法不需要提取特征的描述子，没有曲线或曲面导数，无需预处理的三维数据，只需要一个计算最近点的过程；通过迭代，使最小均方距离单调收敛，以达到配准的目的。该算法由 Besl 的一篇论文提出。在论文中提到，除点集外，ICP 算法还可以应用于线段集、隐式曲线、参数曲线、三角面、隐式曲面、参数曲面等。

ICP 算法的匹配误差最终收敛于一个预设的公差。在适当的初值下，ICP 算法的收敛速度在开头的几次迭代过程中非常快；但是若初值赋予不当，则有可能影响最终的收敛结果。

## 二. 迭代过程

已知：数据形状点集  $\mathbf{P} = \{\mathbf{p}_i\}$ ，其中  $i = 1, 2, \dots, N_p$ ；模板形状  $\mathbf{X} = \{\mathbf{x}_i\}$ ，其中  $i = 1, 2, \dots, N_x$ 。根据实际应用中的不同情况， $\mathbf{x}_i$  可能是组成形状的点、线段三角平面等（点云匹配的时候  $\mathbf{x}_i$  为点）。

以一个七元向量  $\mathbf{q}$  表示从  $\mathbf{P}$  到  $\mathbf{X}$  的位姿变换

$$\begin{aligned}\mathbf{q} &= [\mathbf{q}_R | \mathbf{q}_T]^T \\ \mathbf{q}_R &= [q_0 \quad q_1 \quad q_2 \quad q_3]^T \\ \mathbf{q}_T &= [q_4 \quad q_5 \quad q_6]^T\end{aligned}$$

其中  $q_0 \geq 0$ ， $\mathbf{q}_R$  为表示旋转的单位 Hamilton 四元数， $\mathbf{q}_T$  为位移向量。

以  $\mathbf{Y}$  表示（在  $\mathbf{X}$  中）与  $\mathbf{P}$  中的点  $\mathbf{p}_i$  所对应（即最近点）的点  $\mathbf{y}_i$  的集合， $k$  表示迭代次数， $d_k$  表示第  $k$  次迭代的均方误差，公差  $\tau$  表示预先指定的迭代精度。

参数初始化为

$$\begin{aligned}\mathbf{P}_0 &= \mathbf{P} \\ \mathbf{q}_0 &= [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \\ k &= 0\end{aligned}$$

那么标准 ICP 算法的迭代过程为：

- 1) 计算最近点集  $\mathbf{Y}_k$ ，即取模板形状  $\mathbf{X}$  中到  $\mathbf{P}_k$  中的点  $\mathbf{p}_{ki}$  的最近点  $\mathbf{y}_{ki}$  的集合（即， $\mathbf{y}_{ki}$  都是  $\mathbf{X}$  中选取的点），其中相同下角标的  $\mathbf{p}_{ki}$  与  $\mathbf{y}_{ki}$  是一一对应的关系。

$$\mathbf{Y}_k = \mathcal{C}(\mathbf{P}_k, \mathbf{X})$$

其平均复杂度为  $O(N_p \log N_x)$ ；最坏情况下复杂度为  $O(N_p N_x)$ 。为加快速度，可在此基础上进行 KD-tree 快速查找。

- 2) 计算位姿变换及匹配误差。由于  $\mathbf{q}_k$  的定义为  $\mathbf{P}_0$  到模板形状  $\mathbf{X}$  中最近点集  $\mathbf{Y}_k$  的位姿变换，因而最终迭代结束时的匹配能够表示完整的位姿变换。

$$(\mathbf{q}_k, d_k) = \mathcal{Q}(\mathbf{P}_0, \mathbf{Y}_k)$$

复杂度为  $O(N_p)$ 。

- 3) 更新位姿变换结果

$$\mathbf{P}_{k+1} = \mathbf{q}_k(\mathbf{P}_0)$$

4) 迭代结束条件判断。均方误差的变化量小于一个预设的指定值时，迭代结束。

$$d_k - d_{k+1} < \tau$$

Besl 的论文中提到，若要得到无量纲的 $\tau$ ，可预设 $\tau = \sqrt{\text{tr}(\Sigma_X)}$ ，即模板形状 $X$ 的协方差矩阵的迹，可表明模板形状的粗糙程度。

### 三. 计算方法

在迭代过程中，需要对最近点集、位姿变换、匹配误差等进行计算。计算最近点集时，一般是先计算点到模板形状的距离，之后取模板形状上相应的最近点。而在确定了最近点集后，可以用四元数、SVD 分解、BA 优化等方式计算位姿变换；以均方误差来衡量匹配误差。

#### 3.1. 点到形状距离

➤ 计算点到点之间的距离一般采用欧氏距离，即

$$d(\mathbf{p}_i, \mathbf{y}_i) = \sqrt{(x_{pi} - x_{yi})^2 + (y_{pi} - y_{yi})^2 + (z_{pi} - z_{yi})^2} = \|\mathbf{p}_i - \mathbf{y}_i\|$$

➤ 点 $\mathbf{p}$ 到点集 $\mathbf{A} = \{\mathbf{a}_i\}$ 的距离（其中 $i = 1, 2, \dots, N_a$ ）为

$$d(\mathbf{p}, \mathbf{A}) = \min_{i=1,2,\dots,N_a} d(\mathbf{p}, \mathbf{a}_i)$$

➤ 设 $\mathbf{l}$ 为连接 $\mathbf{r}_1$ 、 $\mathbf{r}_2$ 两点的线段，则点 $\mathbf{p}$ 到线段 $\mathbf{l}$ 的距离为

$$d(\mathbf{p}, \mathbf{l}) = \min_{u+v=1} \|u\mathbf{r}_1 + v\mathbf{r}_2 - \mathbf{p}\|$$

其中 $u, v \in [0, 1]$ 。

➤ 点 $\mathbf{p}$ 到线段集 $\mathbf{L} = \{\mathbf{l}_i\}$ 的距离（其中 $i = 1, 2, \dots, N_l$ ）为

$$d(\mathbf{p}, \mathbf{L}) = \min_{i=1,2,\dots,N_l} d(\mathbf{p}, \mathbf{l}_i)$$

即 $\mathbf{L}$ 中到 $\mathbf{p}$ 的最近点 $\mathbf{y}_j$ 满足

$$d(\mathbf{p}, \mathbf{y}_j) = d(\mathbf{p}, \mathbf{L})$$

➤ 设 $\mathbf{t}$ 为 $\mathbf{r}_1$ 、 $\mathbf{r}_2$ 、 $\mathbf{r}_3$ 三点所定义的三角面，则点 $\mathbf{p}$ 到平面 $\mathbf{t}$ 的距离为

$$d(\mathbf{p}, \mathbf{t}) = \min_{u+v+w=1} \|u\mathbf{r}_1 + v\mathbf{r}_2 + w\mathbf{r}_3 - \mathbf{p}\|$$

其中 $u, v, w \in [0, 1]$ 。

➤ 点 $\mathbf{p}$ 到平面集 $\mathbf{T} = \{\mathbf{t}_i\}$ 的距离（其中 $i = 1, 2, \dots, N_t$ ）为

$$d(\mathbf{p}, \mathbf{T}) = \min_{i=1,2,\dots,N_t} d(\mathbf{p}, \mathbf{t}_i)$$

即 $\mathbf{T}$ 中到 $\mathbf{p}$ 的最近点 $\mathbf{y}_j$ 满足

$$d(\mathbf{p}, \mathbf{y}_j) = d(\mathbf{p}, \mathbf{T})$$

#### 3.2. 点到参数实体的距离

便于分析，将参数曲线、参数曲面统一处理为单参数实体 $\mathbf{r}(\mathbf{u})$ 。在参数曲线中 $\mathbf{u} = u \in \mathbb{R}$ ，参数曲面中 $\mathbf{u} = (u, v) \in \mathbb{R}^2$ 。

那么点 $\mathbf{p}$ 到参数实体 $\mathbf{E}$ 的距离为

$$d(\mathbf{p}, \mathbf{E}) = \min_{\mathbf{r}(\mathbf{u}) \in \mathbf{E}} d(\mathbf{p}, \mathbf{r}(\mathbf{u}))$$

点 $\mathbf{p}$ 到参数实体集 $\mathbf{F} = \{\mathbf{E}_i\}$ 的距离（其中 $i = 1, 2, \dots, N_e$ ）为

$$d(\mathbf{p}, \mathbf{F}) = \min_{i=1,2,\dots,N_e} d(\mathbf{p}, \mathbf{E}_i)$$

即 $F$ 中到 $p$ 的最近点 $y_j$ 满足

$$d(p, y_j) = d(p, F)$$

在实际的计算中，一般很难得到对距离函数的解析解。因此在计算点到参数实体的距离时，可以先通过构造单形体近似（将参数曲线/曲面近似为线段集或三角面集）的方式计算出一定误差范围内的估计值，再通过非线性优化的方式（Besl 论文中为牛顿法迭代）最小化误差，得到一定精度下距离函数的数值解。

在构造单形体近似时，对于空间参数曲线 $C = \{r(u)\}$ ，可在不偏离预设误差 $\delta$ 的情况下，将其分段线性近似为线段集 $L(C, \delta)$ ；之后通过标记该线段集 $L(C, \delta)$ 上每一点与曲线上相对应的自变量 $u$ ，可以得到线段集中的最近点的自变量 $u_a$ 的估计值。

相似地，对于参数曲面 $S = \{r(u, v)\}$ ，也可以在预设误差 $\delta$ 范围内计算出其分段三角近似的三角集 $T(S, \delta)$ （triangle set, Besl 的论文中所提及）。通过标记每个三角的顶点与曲面上相对应的自变量 $(u, v)$ ，可以得到三角集中的最近点的自变量 $(u_a, v_a)$ 的估计值。

这样可以假定 $u_a$ 可获得，且 $r(u_a)$ 与参数实体上的最近点足够接近，并将之作为牛顿法优化的初值，对误差函数进行最小化，即可得到点 $p$ 到参数实体距离的数值解。最小化的目标函数为

$$f(u) = \|r(u) - p\|^2$$

以 $\nabla = [\partial/\partial u]^T$ 表示微分（梯度）操作，那么牛顿迭代法中所用到的 $g$  (Gradient, 梯度向量) 和  $H$  (Hessian, 海森矩阵, 即二阶梯度矩阵) 分别为 $\nabla f$ 和 $\nabla \nabla^T(f)$ 。

若为参数曲面， $u = (u, v)$ ，那么 2D 梯度向量为

$$\nabla f = [f_u \quad f_v]^T$$

2D 海森矩阵为

$$\nabla \nabla^T(f) = \begin{bmatrix} f_{uu} & f_{uv} \\ f_{uv} & f_{vv} \end{bmatrix}$$

其中所提及的各偏导数为

$$\begin{aligned} f_u(u) &= 2r_u^T(u)(r(u) - p) \\ f_v(u) &= 2r_v^T(u)(r(u) - p) \\ f_{uu}(u) &= 2r_{uu}^T(u)(r(u) - p) - 2r_u^T(u)r_u(u) \\ f_{vv}(u) &= 2r_{vv}^T(u)(r(u) - p) - 2r_v^T(u)r_v(u) \\ f_{uv}(u) &= 2r_{uv}^T(u)(r(u) - p) - 2r_u^T(u)r_v(u) \end{aligned}$$

若为参数曲线， $u = u$ ，那么只需取一阶导数 $f_u$ 及二阶导数 $f_{uu}$ 。

牛顿迭代公式为

$$u_{k+1} = u_k - [\nabla \nabla^T(f)(u_k)]^{-1} \nabla f(u_k)$$

### 3.3. 相对位姿

在确定了最近点集后，可用四元数、SVD 分解、BA 优化等方式计算位姿变换。在 Besl 的论文中使用的是基于四元数的算法，而最常用的则是基于 SVD 分解的算法。

#### 3.3.1. 基于四元数的位姿计算方式

前文已经阐明，以一个七元向量 $q$ 表示从 $P$ 到 $Y$ 的位姿变换

$$\begin{aligned} q &= [q_R | q_T]^T \\ q_R &= [q_0 \quad q_1 \quad q_2 \quad q_3]^T \\ q_T &= [q_4 \quad q_5 \quad q_6]^T \end{aligned}$$

其中 $q_0 \geq 0$ ， $q_R$ 为表示旋转的单位 Hamilton 四元数， $q_T$ 为位移向量。

点集 $P$ 的中心为

$$\mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i$$

最近点集 $\mathbf{Y}$ 的中心为(式中 $N_y = N_p$ )

$$\boldsymbol{\mu}_y = \frac{1}{N_y} \sum_{i=1}^{N_y} \mathbf{y}_i$$

$\mathbf{P}$ 和 $\mathbf{Y}$ 的协方差矩阵为

$$\begin{aligned} \boldsymbol{\Sigma}_{py} &= \frac{1}{N_p} \sum_{i=1}^{N_p} [(\mathbf{p}_i - \boldsymbol{\mu}_p)(\mathbf{y}_i - \boldsymbol{\mu}_y)^T] \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} [\mathbf{p}_i \mathbf{y}_i^T] - \boldsymbol{\mu}_p \boldsymbol{\mu}_y^T \end{aligned}$$

由协方差矩阵构造 $4 \times 4$ 对称矩阵

$$Q(\boldsymbol{\Sigma}_{py}) = \begin{bmatrix} \text{tr}(\boldsymbol{\Sigma}_{py}) & \Delta^T \\ \Delta & \boldsymbol{\Sigma}_{py} + \boldsymbol{\Sigma}_{py}^T - \text{tr}(\boldsymbol{\Sigma}_{py})\mathbf{I}_3 \end{bmatrix}$$

其中 $\mathbf{I}_3$ 为 $3 \times 3$ 单位矩阵,  $\text{tr}(\boldsymbol{\Sigma}_{py})$ 为协方差矩阵的迹,  $\Delta$ 为列向量,  $\Delta = [A_{23} \ A_{31} \ A_{12}]^T$ , 向量中的元素为

$$A_{ij} = (\boldsymbol{\Sigma}_{py} - \boldsymbol{\Sigma}_{py}^T)_{ij}$$

将 $Q(\boldsymbol{\Sigma}_{py})$ 的最大特征值所对应的特征向量(4维向量)单位化, 即为四元数 $\mathbf{q}_R$ 的最佳取值。对应的位移向量 $\mathbf{q}_T$ 为

$$\mathbf{q}_T = \boldsymbol{\mu}_y - \mathbf{R}(\mathbf{q}_R)\boldsymbol{\mu}_p$$

其中 $\mathbf{R}(\mathbf{q}_R)$ 为四元数 $\mathbf{q}_R$ 所对应的旋转矩阵(详见《三维空间刚体旋转运动描述》)。

### 3.3.2. 基于SVD分解的位姿计算方式

基于SVD分解的算法作为最常用的一种点集拟合算法, 于1987年在K. S. Arun的一篇论文中被提出。由于噪声的存在, 求解点集拟合的问题可描述为

$$\mathbf{y}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} + \mathbf{N}_i$$

其中 $\mathbf{R}$ 、 $\mathbf{t}$ 用来描述位姿,  $\mathbf{N}_i$ 为噪声向量。

点集 $\mathbf{P}$ 的中心为

$$\boldsymbol{\mu}_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{p}_i$$

最近点集 $\mathbf{Y}$ 的中心为(式中 $N_y = N_p$ )

$$\boldsymbol{\mu}_y = \frac{1}{N_y} \sum_{i=1}^{N_y} \mathbf{y}_i$$

假定问题的最小二乘解为 $\hat{\mathbf{R}}$ 、 $\hat{\mathbf{t}}$ , 那么有

$$\hat{\mathbf{y}}_i = \hat{\mathbf{R}}\mathbf{p}_i + \hat{\mathbf{t}}$$

此时 $\hat{\mathbf{Y}}$ 和 $\mathbf{Y}$ 有相同的中心, 且与 $\boldsymbol{\mu}_p$ 的相对位姿可由最小二乘解描述, 即

$$\begin{aligned} \boldsymbol{\mu}_y &= \boldsymbol{\mu}_{\hat{y}} \\ \boldsymbol{\mu}_{\hat{y}} &= \hat{\mathbf{R}}\boldsymbol{\mu}_p + \hat{\mathbf{t}} \end{aligned}$$

为方便求解旋转变换 $\mathbf{R}$ , 将对应的点集的中心移至坐标系原点, 消除位移向量 $\mathbf{t}$ 的影响, 即令各点坐标减去其点集中心坐标

$$\begin{aligned} \mathbf{p}'_i &\triangleq \mathbf{p}_i - \boldsymbol{\mu}_p \\ \mathbf{y}'_i &\triangleq \mathbf{y}_i - \boldsymbol{\mu}_y \end{aligned}$$

那么，有均方误差阵

$$\Sigma^2 = \sum_{i=1}^{N_p} \|\mathbf{y}'_i - \mathbf{R}\mathbf{p}'_i\|^2$$

所求的 $\mathbf{R}$ 越精确，那么 $\Sigma^2$ 也相应地越小，越接近 0。

至此，求解相对位姿的最小二乘问题被分解、转化为两个步骤：

- 计算 $\hat{\mathbf{R}}$ ，以使 $\Sigma^2$ 达到最小
- 以所求的 $\hat{\mathbf{R}}$ ，通过 $\hat{\mathbf{t}} = \boldsymbol{\mu}_y - \hat{\mathbf{R}}\boldsymbol{\mu}_p$ ，计算 $\hat{\mathbf{t}}$

将 $\Sigma^2$ 展开，可得

$$\begin{aligned}\Sigma^2 &= \sum_{i=1}^{N_p} (\mathbf{y}'_i - \mathbf{R}\mathbf{p}'_i)^T (\mathbf{y}'_i - \mathbf{R}\mathbf{p}'_i) \\ &= \sum_{i=1}^{N_p} (\mathbf{y}'_i{}^T \mathbf{y}'_i + \mathbf{p}'_i{}^T \mathbf{R}^T \mathbf{R} \mathbf{p}'_i - \mathbf{y}'_i{}^T \mathbf{R} \mathbf{p}'_i - \mathbf{p}'_i{}^T \mathbf{R}^T \mathbf{y}'_i) \\ &= \sum_{i=1}^{N_p} (\mathbf{y}'_i{}^T \mathbf{y}'_i + \mathbf{p}'_i{}^T \mathbf{p}'_i - 2\mathbf{y}'_i{}^T \mathbf{R} \mathbf{p}'_i)\end{aligned}$$

前两项由点集所确定，为定值，因此最小化 $\Sigma^2$ 可等价于最大化

$$F = \sum_{i=1}^{N_p} \mathbf{y}'_i{}^T \mathbf{R} \mathbf{p}'_i$$

由矩阵的迹的性质

$$\mathbf{y}^H \mathbf{x} = \text{trace}(\mathbf{x} \mathbf{y}^H)$$

其中 $\mathbf{x}$ 、 $\mathbf{y}$ 为向量。可将 $F$ 化为

$$\begin{aligned}F &= \sum_{i=1}^{N_p} \mathbf{y}'_i{}^T \mathbf{R} \mathbf{p}'_i \\ &= \sum_{i=1}^{N_p} \text{trace}(\mathbf{R} \mathbf{p}'_i \mathbf{y}'_i{}^T) \\ &= \text{trace} \left( \sum_{i=1}^{N_p} \mathbf{R} \mathbf{p}'_i \mathbf{y}'_i{}^T \right) \\ &= \text{trace} \left( \mathbf{R} \sum_{i=1}^{N_p} \mathbf{p}'_i \mathbf{y}'_i{}^T \right) \\ &= \text{trace}(\mathbf{R} \mathbf{H})\end{aligned}$$

其中 $\mathbf{H} = \sum_{i=1}^{N_p} \mathbf{p}'_i \mathbf{y}'_i{}^T$ 。将 $\mathbf{H}$ 作 SVD 分解为

$$\mathbf{H} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{V}^T$$

其中 $\mathbf{U}$ 、 $\mathbf{V}$ 为 $3 \times 3$ 标准正交矩阵， $\boldsymbol{\Lambda}$ 为 $3 \times 3$ 非负定对角阵。令

$$\mathbf{X} = \mathbf{V} \mathbf{U}^T$$

那么

$$\mathbf{X}\mathbf{H} = \mathbf{V}\mathbf{U}^T \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$$

为对称正定矩阵。由矩阵的迹的性质（其中 $\mathbf{A}\mathbf{A}^T$ 为正定矩阵， $\mathbf{B}$ 为任意标准正交矩阵）

$$\text{trace}(\mathbf{A}\mathbf{A}^T) \geq \text{trace}(\mathbf{B}\mathbf{A}\mathbf{A}^T)$$

可知, 对 $\mathbf{X}\mathbf{H}$ 左乘任一标准正交阵(可视为左乘旋转变换), 其所得结果的迹只能小于等于 $\mathbf{X}\mathbf{H}$ , 即

$$\text{trace}(\mathbf{X}\mathbf{H}) \geq \text{trace}(\mathbf{B}\mathbf{X}\mathbf{H})$$

因此 $\mathbf{R} = \mathbf{X}$ 即为所求使 $F$ 达到最大值的 $\mathbf{R}$ 的取值。

综上, 基于 **SVD 分解** 的位姿计算算法步骤为:

- 1) 计算点集中心 $\mu_p$ 、 $\mu_y$ , 以及将对应的点集的中心移至坐标系原点, 得到 $\mathbf{p}'_i$ 、 $\mathbf{y}'_i$ 。
- 2) 计算矩阵 $\mathbf{H}$

$$\mathbf{H} = \sum_{i=1}^{N_p} \mathbf{p}'_i \mathbf{y}'_i{}^T$$

- 3) 将 $\mathbf{H}$ 进行 SVD 分解

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T$$

- 4) 计算

$$\mathbf{X} = \mathbf{V}\mathbf{U}^T$$

- 5) 判断, 若 $\det(\mathbf{X}) = 1$ , 则 $\mathbf{R} = \mathbf{X}$ ; 若 $\det(\mathbf{X}) = -1$ , 则算法失败, 无法解出 $\mathbf{R}$  (这种情况极少发生)。

- 6) 根据求出的 $\mathbf{R}$ , 计算

$$\mathbf{t} = \mu_y - \mathbf{R}\mu_p$$

所得的 $\mathbf{R}$ 、 $\mathbf{t}$ 即为所求相对位姿。

### 3.3.3. 基于 BA 优化的位姿计算方式

求解对应点集间的相对位姿的另一种方式是非线性优化, 以 BA 的方式, 利用高斯-牛顿、L-M 等迭代算法求解最优值。四元数有模长为 1 的约束, 旋转矩阵有标准正交阵的约束; 为了将有约束的优化问题转化为无约束的优化问题, 使用 6 维向量 $\xi = [\rho, \phi]^T$ 表示位姿 (前 3 维表示平移, 后 3 维为旋转向量。即很多资料中提及的将李群转化为李代数)。

误差函数为

$$\mathbf{e}_i = \mathbf{y}_i - \exp(\xi^\wedge) \mathbf{p}_i$$

设根据现有位姿 (即将要进行优化的位姿) 计算出的变换后的点坐标为 $\mathbf{q}_i(X'_i, Y'_i, Z'_i)$ , 即

$$\mathbf{q}_i = \exp(\xi^\wedge) \mathbf{p}_i$$

故有

$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{q}_i$$

用以最小化的代价函数为

$$\xi^* = \arg \min_{\xi} \frac{1}{2} \sum_{i=1}^{N_p} \|\mathbf{e}_i\|_2^2$$

$\mathbf{e}_i$ 为点的坐标误差 (3 维),  $\xi$ 为相对位姿 (6 维), 优化过程中所用到的 Jacobian 矩阵 (即一阶微分) 为 $3 \times 6$ 矩阵。由链式求导法则, 以及李代数的扰动模型, 推导 Jacobian 矩阵

$$\begin{aligned} J_i &= \frac{\partial \mathbf{e}_i}{\partial \delta \xi} \\ &= \frac{\partial \mathbf{e}_i}{\partial \mathbf{q}_i} \frac{\partial \mathbf{q}_i}{\partial \delta \xi} \end{aligned}$$

$$\begin{aligned}
&= -\frac{\partial \exp(\xi^\wedge) \mathbf{p}_i}{\partial \delta \xi} \\
&= -(\exp(\xi^\wedge) \mathbf{p}_i)^\odot \\
&= -[\mathbf{I}_3 \quad -[\mathbf{q}_i]_\times] \\
&= -\begin{bmatrix} 1 & 0 & 0 & 0 & Z'_i & -Y'_i \\ 0 & 1 & 0 & -Z'_i & 0 & X'_i \\ 0 & 0 & 1 & Y'_i & -X'_i & 0 \end{bmatrix}
\end{aligned}$$

### 3.4. 均方误差

在计算出相对最近点集的位姿后，可计算其均方误差

$$d_k = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{y}_{ik} - \mathbf{R}(\mathbf{q}_{kR})\mathbf{p}_{i0} - \mathbf{q}_{kT}\|^2$$

可证明，ICP 算法迭代的均方误差具有收敛性。

### 参考

- [1] ]Paul J., Besl, Neil D. Mckay. A Method for Registration of 3-D Shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992,14(2):239-256.
- [2] K. S. Arun, T. S. Huang, S. D. Blostein. Least-squares fitting of two 3-D point sets[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 1987, 9(5): 698-700
- [3] 张贤达.矩阵分析与应用[M].清华大学出版社:北京,2004:50.
- [4] 高翔,张涛等.视觉 SLAM 十四讲：从理论到实践[M].电子工业出版社:北京,2017:172-175.