DLT 算法整理

DLT 算法作为 PnP 算法的一种，相比于 P3P、EPnP 等其他常见算法，并不计算参考点深度，而是直接以参考点坐标变换关系列写方程作为约束，求解相机所在位姿。作为一种很早就已经提出的算法，它的计算量较低但是也有着误差较大的缺陷。

参考点从世界坐标系到像素坐标系的转换关系为

$$s_i\boldsymbol{u}_i = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{p}_i^w$$

其中s_i为尺度参数，$\boldsymbol{u}_i = [u_i \quad v_i \quad 1]^T$为参考点像素坐标系下坐标，$\boldsymbol{K}$为相机内参，$[\boldsymbol{R}|\boldsymbol{t}]$为所求的位姿（也是相机外参），$\boldsymbol{p}_i^w = [x_i^w \quad y_i^w \quad z_i^w \quad 1]^T$为参考点世界坐标系下坐标。

将$\boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$记为$3\times 4$矩阵$\boldsymbol{T}$，又称投影矩阵，并且将$\boldsymbol{T}$的各行记作

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \boldsymbol{t}_2^T \\ \boldsymbol{t}_3^T \end{bmatrix}$$

其中

$$\boldsymbol{t}_1 = [t_1 \quad t_2 \quad t_3 \quad t_4]^T$$
$$\boldsymbol{t}_2 = [t_5 \quad t_6 \quad t_7 \quad t_8]^T$$
$$\boldsymbol{t}_3 = [t_9 \quad t_{10} \quad t_{11} \quad t_{12}]^T$$

这样$\boldsymbol{u}_i$和$\boldsymbol{T}\boldsymbol{p}_i^w$间只差 1 个尺度参数，相互间叉积为零。

$$\boldsymbol{u}_i \times \boldsymbol{T}\boldsymbol{p}_i^w = \boldsymbol{0}$$

展开为

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{t}_1^T\boldsymbol{p}_i^w \\ \boldsymbol{t}_2^T\boldsymbol{p}_i^w \\ \boldsymbol{t}_3^T\boldsymbol{p}_i^w \end{bmatrix} = \begin{bmatrix} v_i\boldsymbol{t}_3^T\boldsymbol{p}_i^w - \boldsymbol{t}_2^T\boldsymbol{p}_i^w \\ \boldsymbol{t}_1^T\boldsymbol{p}_i^w - u_i\boldsymbol{t}_3^T\boldsymbol{p}_i^w \\ u_i\boldsymbol{t}_2^T\boldsymbol{p}_i^w - v_i\boldsymbol{t}_1^T\boldsymbol{p}_i^w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

易知$\boldsymbol{t}_j^T\boldsymbol{p}_i^w = \boldsymbol{p}_i^{w^T}\boldsymbol{t}_j$成立，可化为反对称矩阵形式

$$\begin{bmatrix} \boldsymbol{0}^T & -\boldsymbol{p}_i^{w^T} & v_i\boldsymbol{p}_i^{w^T} \\ \boldsymbol{p}_i^{w^T} & \boldsymbol{0}^T & -u_i\boldsymbol{p}_i^{w^T} \\ -v_i\boldsymbol{p}_i^{w^T} & u_i\boldsymbol{p}_i^{w^T} & \boldsymbol{0}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

其中第一项为3×9矩阵；第二项为9×1矩阵，记作$\boldsymbol{t}_s$。

易知等式第三行与前两行线性相关，因此可将前两行作为约束

$$\begin{bmatrix} \boldsymbol{0}^T & -\boldsymbol{p}_i^{w^T} & v_i\boldsymbol{p}_i^{w^T} \\ \boldsymbol{p}_i^{w^T} & \boldsymbol{0}^T & -u_i\boldsymbol{p}_i^{w^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

将第一项记作$\boldsymbol{A}_i$，有

$$\boldsymbol{A}_i\boldsymbol{t} = \boldsymbol{0}$$

这样 1 个参考点可提供 2 个约束，而$\boldsymbol{t}$中共 12 个元素，因此需要至少 6 个参考点来求解。全部n个参考点组成完整的方程组

$$\begin{bmatrix} \boldsymbol{0}^T & -\boldsymbol{p}_1^{w^T} & v_1\boldsymbol{p}_1^{w^T} \\ \boldsymbol{p}_1^{w^T} & \boldsymbol{0}^T & -u_1\boldsymbol{p}_1^{w^T} \\ \vdots & \vdots & \vdots \\ \boldsymbol{0}^T & -\boldsymbol{p}_n^{w^T} & v_n\boldsymbol{p}_n^{w^T} \\ \boldsymbol{p}_n^{w^T} & \boldsymbol{0}^T & -u_n\boldsymbol{p}_n^{w^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

简写作

$$\boldsymbol{At} = \boldsymbol{0}$$

可通过方程组求解得到$\boldsymbol{t}$中的每个元素的值。但是在实际的解算中，由于噪声的存在，且一般情况下n会远大于 6，因此需要求其近似解。近似解要求在满足$\|\boldsymbol{t}\| = 1$的前提下，最小化$\|\boldsymbol{At}\|$，因此解为$\boldsymbol{A}^T\boldsymbol{A}$最小特征值对应的单位特征向量。或者用矩阵 SVD 分解的方式，解为最小奇异值对应的单位奇异向量。将求解得到的值重新组成矩阵，记作$\boldsymbol{T}_L$。

实际问题中，为了抑制噪声，需要对参考点坐标进行归一化处理，通过欧氏变换，使得点集中心位于原点，且参考点到原点的平均距离分别为$\sqrt{2}$、$\sqrt{3}$。将变换后的坐标记作$\widehat{\boldsymbol{u}}_i$和$\widehat{\boldsymbol{p}}_i^w$，有

$$\widehat{\boldsymbol{u}}_i = \boldsymbol{T}_{2D}\boldsymbol{u}_i$$
$$\widehat{\boldsymbol{p}}_i^w = \boldsymbol{T}_{3D}\boldsymbol{p}_i^w$$

其中$\boldsymbol{T}_{2D}$、$\boldsymbol{T}_{3D}$分别为 2D、3D 下的变换矩阵。将归一化坐标求解得到的投影矩阵$\widehat{\boldsymbol{T}}_L$。去归一化，得到投影矩阵

$$\boldsymbol{T}_L = \boldsymbol{T}_{2D}^{-1}\widehat{\boldsymbol{T}}_L\boldsymbol{T}_{3D}$$

考虑到$\boldsymbol{T}_L$中还有内参矩阵$\boldsymbol{K}$，令

$$\boldsymbol{J} = \boldsymbol{K}^{-1}\boldsymbol{T}_L$$

那么

$$\boldsymbol{J} = s[\boldsymbol{R}|\boldsymbol{t}]$$

其中s为尺度参数。为抵消尺度参数的影响，取

$$\boldsymbol{L} = \mathrm{sign}(J_{34}) \cdot \boldsymbol{J}$$

其中J_{34}为矩阵$\boldsymbol{J}$的第 3 行第 4 列元素，而$\mathrm{sign}()$函数则为符号函数

$$\mathrm{sign}(J_{34}) = \begin{cases} 1,\ if\ \ J_{34} > 0 \\ -1,\ if\ \ J_{34} < 0 \end{cases}$$

令

$$\boldsymbol{L} = [\boldsymbol{L}_1 \quad \boldsymbol{L}_2 \quad \boldsymbol{L}_3 \quad \boldsymbol{L}_4]$$

那么

$$[\boldsymbol{R}|\boldsymbol{t}] = \frac{\boldsymbol{L}}{\|\boldsymbol{L}_3\|}$$

即为所求的位姿矩阵。

参考

[1] Hartley R, Zisserman A. Multiple View Geometry in Computer Vision[M]. Cambridge: Cambridge university press, 2003.

[2] 张培科,武元新.基于改进的 DLT 算法的 PnP 问题求解[J].信息技术,2018(01):36-41.

[3] 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.