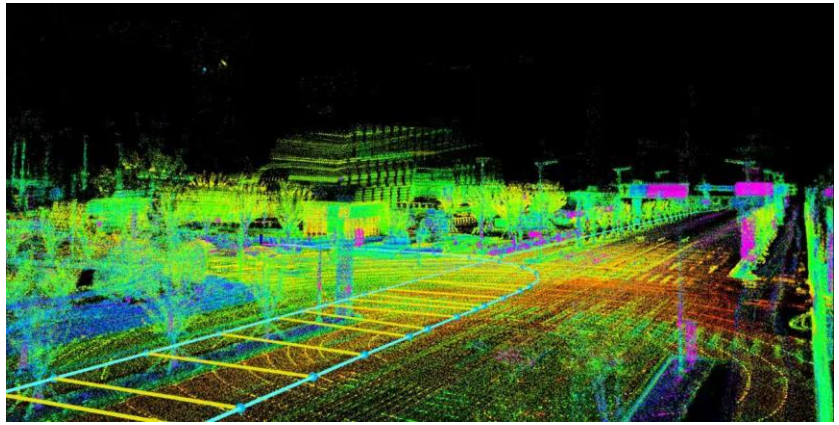
微信扫码加入星球

自动驾驶中实战基础之点云去畸变与对齐

Camera + LiDAR + Radar + IMU



主　讲　人：爱喝苦咖啡的小阿飞

公　众　号：3D 视 觉 工 坊

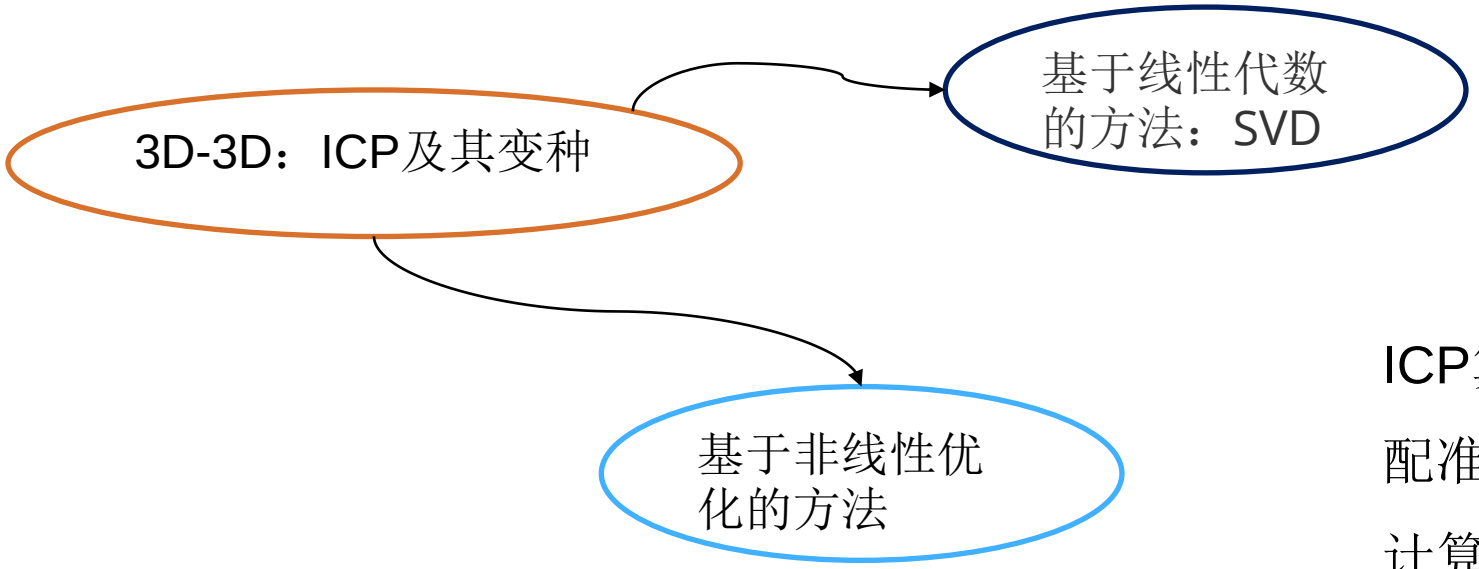
内容

一、 3D-3D求解方法

二、 3D-3D原理推导

三、 3D-3D求解实现

一、3D-3D求解方法



ICP算法是本质上是基于最小二乘法的最优配准方法。该算法重复进行对应关系点对，计算最优刚体变换，直到满足正确配准的收敛精度要求。改进的ICP方法针对最近点选择上采用Point to Point、Point to Plane、Point to Projection等一些方法完成，或者针对收敛函数做一些改变。

ICP：iterative closest point

假设有两组点，如下：

$$P = \{p_1, \dots, p_n\}, \qquad P' = \{p'_1, \dots, p'_n\}$$

寻找一个欧式变换R, t：

$$P = RP' + t$$

- SVD方法

假设两组点是匹配好的两组点，根据前面描述的 ICP 问题，定义第 i 对点的误差项：

$$E_i = p_i - (Rp'_i + t)$$

构建最小二乘问题，求使误差平方和达到极小的R, t：

$$\min_{\boldsymbol{R},\boldsymbol{t}} J = \frac{1}{2}\sum_{i=1}^{n} \|(\boldsymbol{p}_i - (\boldsymbol{R}\boldsymbol{p}_i' + \boldsymbol{t}))\|_2^2.$$

首先，定义两组点的质心:

$$p = \frac{1}{n}\sum_{i=1}^{n}(p_i), \quad p' = \frac{1}{n}\sum_{i=1}^{n}(p_i').$$

在误差函数中，将两组点作去质心处理，得如下式子：

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{n}\|p_i - (Rp_i' + t)\|^2 &= \frac{1}{2}\sum_{i=1}^{n}\|p_i - Rp_i' - t - p + Rp' + p - Rp'\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{n}\|(p_i - p - R(p_i' - p')) + (p - Rp' - t)\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{n}(\|p_i - p - R(p_i' - p')\|^2 + \|p - Rp' - t\|^2 + \\ &\quad 2(p_i - p - R(p_i' - p'))^T(p - Rp' - t)).\end{aligned}$$

注意到交叉项部分中，$(p_i - p - R(p_i' - p'))^T$在累加求和之后是为零的，因此优化目标函数可以简化为:

$$\min_{R,t} J = \frac{1}{2}\sum_{i=1}^{n}\|p_i - p - R(p_i' - p')\|^2 + \|p - Rp' - t\|^2.$$

计算每个点的去质心坐标：

$$\boldsymbol{q}_i = \boldsymbol{p}_i - \boldsymbol{p}, \quad \boldsymbol{q}_i' = \boldsymbol{p}_i' - \boldsymbol{p}_i'$$

于是，将 $\min_{\boldsymbol{R},\boldsymbol{t}} J = \frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{p}_i - \boldsymbol{p} - \boldsymbol{R}(\boldsymbol{p}_i' - \boldsymbol{p}')\|^2 + \|\boldsymbol{p} - \boldsymbol{R}\boldsymbol{p}' - \boldsymbol{t}\|^2$. 转换为如下式子：

$$E_1(\mathrm{R},\mathrm{t}) = \frac{1}{2}\sum_{i=1}^{n}\|\boldsymbol{q}_i - R\boldsymbol{q}_i'\|^2 = \frac{1}{2}\sum_{I=1}^{n}\left(\boldsymbol{q}_i^T\boldsymbol{q}_i + \boldsymbol{q}_i'^T R^T R\boldsymbol{q}_i' - 2\boldsymbol{q}_i^T R\boldsymbol{q}_i'\right) \quad (1)$$

则，最终转换为：

$$\min J(R,t) = \operatorname{argmin} E_1(R,t) = \operatorname{argmax} \sum_{i=1}^{n} \boldsymbol{q}_i^T R\boldsymbol{q}_i' \quad (2)$$

1) 定理：若有正定矩阵CC^T，则对于任意正交矩阵R，有$Trace(CC^T) \geq Trace(RCC^T)$;

2) Trace(AB) = Trace(BA);

$$\sum_{i=1}^{n} \boldsymbol{q}_i{}^T R\boldsymbol{q}_i') = \sum_{i=1}^{n} \text{Trace}(R\boldsymbol{q}_i'\boldsymbol{q}_i{}^T) = \text{Trace}(\sum_{i=1}^{n} R\boldsymbol{q}_i'\boldsymbol{q}_i{}^T) \quad (3)$$

最终问题转换为找到最佳的R，使得上式最大。

$$\text{Trace}(\sum_{i=1}^{n} R\boldsymbol{q}_i'\boldsymbol{q}_i{}^T) = \boldsymbol{Trace(RQ)} \quad (4)$$

即，寻找一个R，使得$Trace(R\boldsymbol{Q})$转化为$Trace(CC^T)$。

对$\boldsymbol{Q}$进行分解

$$\boldsymbol{Q} = U\Sigma V^T$$

$\boldsymbol{U}$和V都是方阵，都是酉矩阵，满足$U^TU = I$， $V^TV = I$。Σ为对角矩阵。

由此，得：

$$\boldsymbol{RQ} = \boldsymbol{R}U\Sigma V^T$$

$$= \boldsymbol{R}U\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}V^T$$

$$= \boldsymbol{R}U\Sigma^{\frac{1}{2}}(V\Sigma^{\frac{1}{2}})^T$$

则有：$R = VU^T$，使得$\sum_{i=1}^{n} \boldsymbol{q}_i{}^T R\boldsymbol{q}_i'$取最大值，此时$\boldsymbol{t} = \boldsymbol{p} - \boldsymbol{R}\boldsymbol{p}'$

于是，ICP 可以分为以下三个步骤求解:

1. 计算两组点的质心位置 p和p' ，然后计算每个点的去质心坐标:

$$q_i = p_i - p$$

$$q_i' = p_i' - p'$$

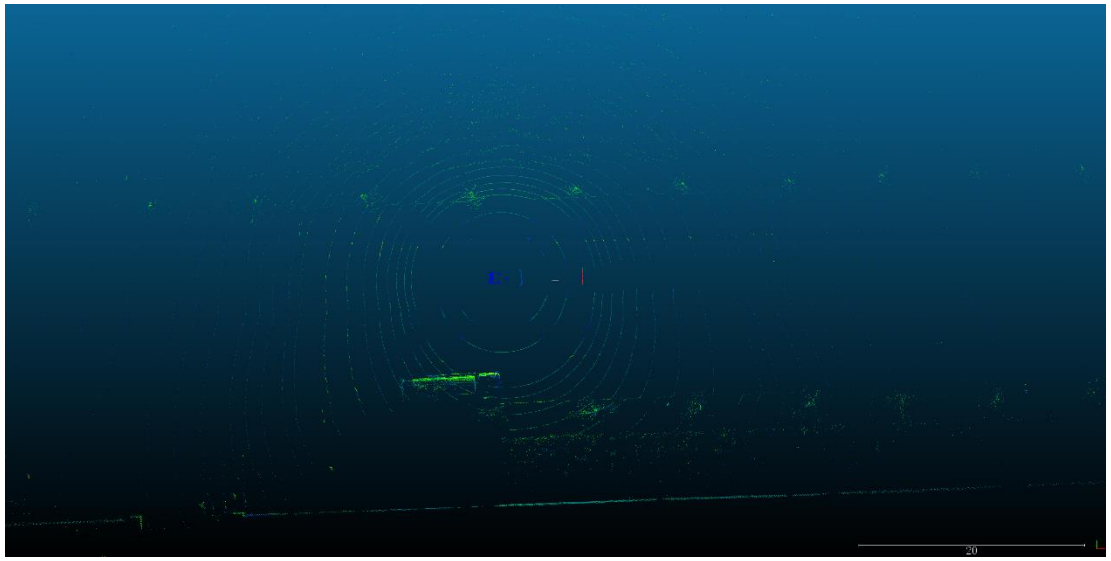
2. 根据以下优化问题计算旋转矩阵:

$$R^* = argmin\frac{1}{2}\sum_{i=1}^{n} || q_i - Rq_i' ||^2$$

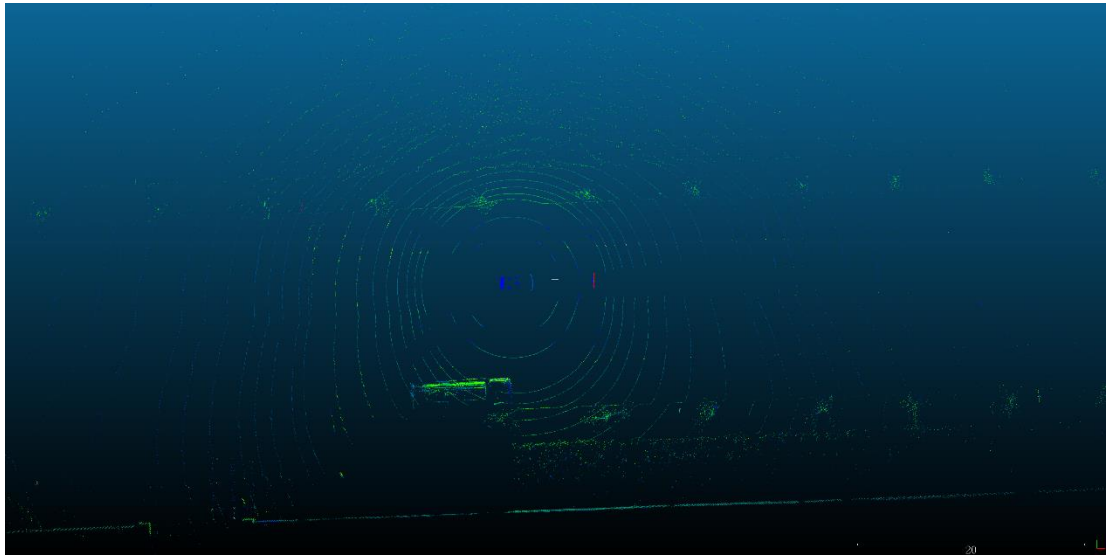
3. 根据第二步的R，计算 t:

$$t^* = p - Rp'$$

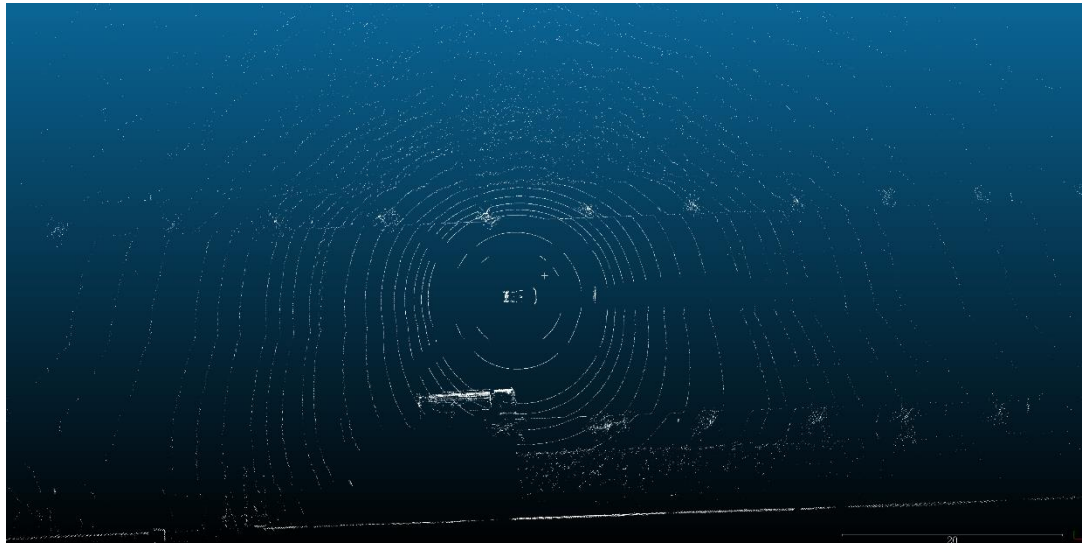
```
//创建ICP的实例类
pcl::IterativeClosestPoint<pcl::PointXYZ, pcl::PointXYZ> icp;
// 设定source 点云
icp.setInputSource(cloud_sources);
// 设定target点云
icp.setInputTarget(cloud_target);
//设置最大对应点的欧式距离，只有对应点之间的距离小于该设定值
//的对应点才作为ICP计算的对应点，基本上对所有点都计算了匹配点。
icp.setMaxCorrespondenceDistance(100);
//icp迭代条件设定，满足如下其中一个，即停止迭代
// 设置最大迭代次数，迭代停止条件之一
icp.setMaximumIterations(100);
// 设置前后两次迭代的转换矩阵的最大epsilion，
//一旦两次迭代小于最大容差，则认为已经收敛到最优解，迭代停止， default： 0。
// 迭代停止条件之二
// icp.setTransformationEpsilon(1e-6);
//设置前后两次迭代的点对的欧式距离均值的最大容差，default：-std::numeric_limits::max ()。
//迭代终止条件之三
// icp.setEuclideanFitnessEpsilon(1e-6);
// icp.setRANSACIterations(0);
//执行ICP转换，并保存对齐后的点云
```



Target点云



source点云



结果点云

1) 《视觉SLAM十四讲》；

2) Point cloud library (PCL)

购买该课程请扫描二维码

微信扫码加入星球

感谢聆听

Thanks for Listening