总结ICP.PL-ICP.NICP.IMLS-ICP并比较其异同

ICP

迭代最近点算法（Iterative Closest Point，ICP）是点云配准的经典算法。

> 点云匹配算法是为了匹配两帧点云数据，从而得到传感器（激光雷达或摄像头）前后的位姿差，即里程数据。

icp基本思想

有两组点云集合X,P

$$X = x_1, x_2, \cdots, x_{N_x}$$

$$P = p_1, p_2, \cdots, p_{N_x}$$

其中x_i和p_i表示点云坐标，N_x和N_p表示点云的数量。

求解旋转矩阵R和平移向量t使得下式结果最小。

$$E(R,t) = \frac{1}{N_p}\sum_{i=1}^{N_p}\|x_i - Rp_i - t\|^2$$

由于激光点云很难确定点之间的匹配关系，所以一般通过最近点来确定点云的匹配关系，然后通过不断地迭代来缩小误差，最终得到误差方程最小地旋转矩阵R和t。

算法流程

1）寻找对应点

通常使用编码盘的里程计数据或者假设车体进行匀速运动得到位姿差，即当前机器人在上次机器人坐标系中的位姿。将此R和t作为ICP算法的first guess，帮助算法寻找点云对应点。

2）根据对应点计算R和t

这一步就是根据找好的对应点构建误差方程。普通的ICP 是使用**点到点的距离作为误差**的。误差方程如下：

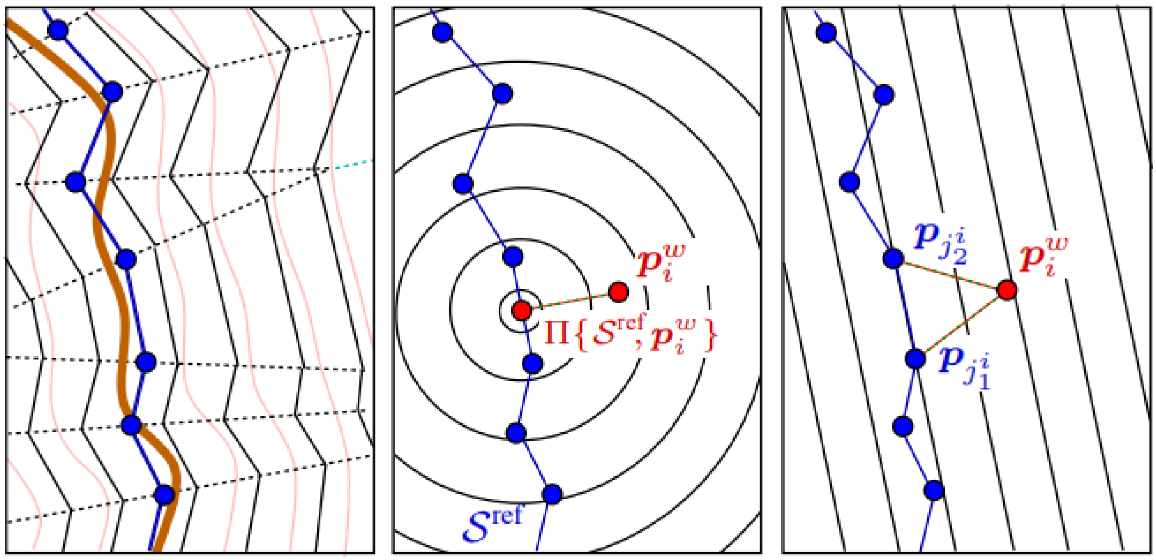
$$E(R,t) = \frac{1}{N_p}\sum_{i=1}^{N_p}\|x_i - Rp_i - t\|^2$$

然后根据SVD等算式求解出R和t

3）将R，t再作为算法地初始解，进行最近点匹配，最小二乘求解。

PL-ICP

PL-ICP相对于PP-ICP最大的区别是其改进了误差方程。PL-ICP是点对点的距离作为误差而PL-ICP是采用点到其最近两个点连线的距离。



(a) Distance to curve and to polyline　(b) Point-to-point metric　(c) Point-to-line metric

变量说明：

- P_i^w: 点云集合P中点i的三维坐标
- P_{j1}^i、P_{j2}^i： 点云集合X中，距离P_i^w最近的两个点
- n_i^T: 点P_{j1}^i、P_{j2}^i 组成直线的法向量

PL-ICP的目标函数

构造最小二乘问题

$$\min_{T_k} \sum_i \left\| n_i^T \left(x_{j2}^i - T_k p_i \right) \right\|^2$$

其中，T_k为变换矩阵,$T_k p_i = R_k p_i + t_k$

算法流程

1、把当前帧的数据p_i进行初始位姿进行变换，得到$T_k p_i$;

2、对于当前帧中的点$T_k p_i$,在参考帧中找到最近的两个点x_{j1}^i和x_{j2}^i，并计算直线法向量n_i^T和投影点x_j^i;

3、计算当前帧中的点$T_k p_i$与两个投影点x_{j1}^i和x_{j2}^i，去除误差比较大的点;

4、最小化误差函数$\min_{T_k} \sum_i \left\| n_i^T \left(x_{j2}^i - T_k p_i \right) \right\|^2$

与ICP的区别

1、误差形式不同，icp以点到点的距离为误差，pl-icp为点到线的距离作为误差，pl-icp更符合实际

2、收敛速度不同，ICP为一阶收敛，PL-ICP为二阶收敛

3、PL-ICP求解精度高于ICP，特别是在结构化环境中；

4、PL-ICP对初始值更敏感，一般与里程计、CSM配合使用。

NICP

NICP基于实际情况中曲面的特征来对错误点匹配进行滤除，主要考虑的特征为法向量和曲率；

数学描述

p_i, p_j为点云集合P^c, P^r中点的三维坐标

n_i:点i处曲面的法向量

σ_i:点i处曲面的曲率

T:欧式变换矩阵

$\tilde{p} = \begin{bmatrix} p_i \\ n_i \end{bmatrix}$

$T \oplus \tilde{p} = \begin{bmatrix} Rp_i + t \\ Rn_i \end{bmatrix}$

$\mathcal{C}$:两个点云P^c, P^r中对应点对的集合

- 误差函数:
  $e_{ij}(T) = (\tilde{p}_i^c - T \oplus \tilde{p}_j^r)$
- 目标函数的定义为:
  $\sum_C e_{ij}(T)^T \Omega_{ij} \tilde{e_{ij}}(T)$
  $\Omega_{ij} = \begin{bmatrix} \tilde{\Omega}_i^s & 0 \\ 0 & \Omega_i^n \end{bmatrix}$
  $\tilde{\Omega}_{ij}$为信息矩阵

法向量和曲率的计算

- 找到点p_i周围半径R球形空间中的所有点v_i
  $\mu_i^s = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_j \in \mathcal{V}_i} \mathbf{p}_i$
  $\mathbf{\Sigma}_i^s = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_j \in \mathcal{V}_i} \left(\mathbf{p}_i - \mu_i\right)^T \left(\mathbf{p}_i - \mu_i\right)$
  $\Sigma_i^s = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T \quad \lambda_2 > \lambda_1$
- 曲率的定义:
  $\sigma_{\mathrm{i}} = \frac{\lambda_1}{\lambda_1 + \lambda_2}$
- 法向量的定义:
  最小特征值对应的特征向量
- 信息矩阵$\tilde{\Omega}_{ij}$的定义:
  $\Omega_i^s = (\sum_i^s)^{-1}$
  $\Omega_4^n = \begin{cases} R \begin{bmatrix} \frac{1}{\varepsilon} & 0 \\ 0 & 1 \end{bmatrix} \\ I \end{cases} \Bigg] R^T$

点匹配规则

- 没有法向量，拒绝
- 两点间距离大于阈值，拒绝
- 两点曲率之差大于阈值，拒绝
- 两点法向量角度插大于阈值，拒绝

目标函数的求解

目标函数为非线性最小二乘问题，通过LM方法来进行求解；

与icp的异同

- NICP根据点、法向量、曲率来进行配准
- NICP使用LM方法进行求解
- 效果要比ICP更好。

IMLS-ICP

IMLS(Implicit Moving Least Square)-ICP算法选择具有代表性的激光点来进行匹配，既能减少计算量的同时又能减少激光点分布不均匀导致的计算结果出现偏移。所以，IMLS-ICP算法从参考帧点云中把曲面重建出来，将当前帧点云与曲面进行配准。

代表点(Information Point)的选取

- 具有丰富特征的点，即为结构化的点：具有良好的曲率和法向量的定义。
- 曲率越小的点越好，因为曲率为0代表着直线，代表着最结构化的点，也代表着具有非常 好的法向量定义，能够提供足够的约束。
- 选点的时候需要注意选取的激光点的均衡以保证可观性，因为是平面匹配，不存在角度 不可观的情况。只需要考虑X方向和Y方向的可观性。要保证两者的约束基本上是一致的， 才能让结果不出现偏移。

曲面重建

P_k:前n帧激光数据组成的子图；
n_k:点云集P_k中的点p_i的法向量；

$I^{P_k}(x)$: $\mathbb{R}^3$空间的点x到点云集合P_K隐藏曲面的距离。

$I^{P_k}(x)$定义如下：

$$I^{P_k}(x)=\frac{\sum_{p_i\in P_k}W_i(x)\left((x-p_i)\cdot\overrightarrow{n_i}\right)}{\sum_{p_j\in P_k}W_j(x)}$$

其中，权重$W_i(x)$被定义为：

$$W_i(x)=e^{-\|x-p_i\|^2/h^2}$$

匹配求解

S_k:新一帧激光数据；

$I^{P_k}(x)$：当前帧S_k中的点x_i到曲面的距离；

$\overrightarrow{n_i}$： P_k中距离点x_i最近的点的法向量。

点x_i在曲面上的投影y_i为：

$$x_i^{'}=Rx_i+t \qquad\qquad y_i=x_i^{'}-I^{P_k}(x_i^{'})\overrightarrow{n_i}$$

对于新一帧激光数据S_k，通过最小化以下函数，求解姿态变换：

$$\sum_{x_i\in S_k}\left((Rx_i+t-y_i)\cdot\vec{n}_i\right)^2$$

与ICP的区别

- IMLS-ICP对点进行了筛选，并没有使用所有点
- IMLS-ICP优化的不是点到点的距离，而是点到曲面的距离
- 效果要比ICP更好。