

激光SLAM理论与实践-第五期 第四次作业思路讲解





第一题



1.实现imls_icp.cpp中的computeNormal()函数

参考PPT第20页NICP中法向量的计算

• 找到点 p_i 周围半径R球形空间内的所有点 V_i

$$\mu_i^s = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_j \in \mathcal{V}_i} \mathbf{p}_i$$

$$\Sigma_i^s = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_j \in \mathcal{V}_i} (\mathbf{p}_i - \mu_i)^T (\mathbf{p}_i - \mu_i)$$

$$\Sigma_i^s = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T \qquad \lambda_2 > \lambda_1$$

• 曲率的定义:

$$\sigma_{\rm i} = \frac{\lambda_{\rm l}}{\lambda_{\rm l} + \lambda_{\rm 2}}$$

• 法向量的定义:最小特征值对应的特征向量。

```
//计算法向量,第一个特征向量即为最小特征值对应的特征向量,矩阵的第一列 Eigen::SelfAdjointEigenSolver<Eigen::Matrix2d> eig(sigma); //特征值升序排列 Eigen::Matrix2d eigvectors = eig.eigenvectors(); normal = eigvectors.col(0);
```

第一题



2.实现imls_icp.cpp中的ImplictMLSFunction()函数的高度计算部分参考PPT第27页

 $I^{P_{\mathbf{k}}}(x)$ 定义如下:

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

第一题



3.实现imls_icp.cpp中的projSourcePtToSurface() 参考PPT第28页

• $点x_i$ 在曲面上的投影 y_i 为:

$$x'_{i} = Rx_{i} + t$$
 $y_{i} = x'_{i} - I^{P_{k}}(x'_{i})\vec{n}_{i}$

第二题



调用CSM库中的ICP匹配接口

可以参考第二章的代码: /odom_ws/src/main.cpp

需要注意的:第二章函数的返回值是Eigen::Vector3d

```
//求两帧之间的icp位姿匹配
Eigen::Vector3d Scan2::PIICPBetweenTwoFrames(LDP& currentLDPScan,
Ligen::Vector3d tmprPose)
```

第三题



方法	ICP	PL-ICP	NICP	IMLS-ICP	
对应关系	最近点	线性曲面近似	加入法向量曲率	隐式构建曲面	
范数	欧氏距离	点到分段直线 欧氏距离	欧氏距离和法 向量空间距离	到曲面距离	

	Method	Setting	Code	<u>Translation</u>	Rotation	Runtime	Environment	Compare				
1	<u>V-LOAM</u>	::1		0.54 %	0.0013 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)					
J. Zhang a	J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast, IEEE International Conference on Robotics and Automation(ICRA) 2015.											
2	LOAM	***		0.55 %	0.0013 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)					
J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014.												
3	SOFT2	ďď		0.57 %	0.0010 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)					
4	TVL-SLAM+	88 ::		0.58 %	0.0015 [deg/m]	0.3 s	1 core @ 3.0 Ghz (C/C++)					
5	CT-ICP	***		0.59 %	0.0014 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)					
6	HELO	:::		0.64 %	0.0019 [deg/m]	0.1 s	8 cores @ 2.5 Ghz (C/C++)					
7	FMLO	:::		0.64 %	0.0013 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)					
8	SOFT-SLAM	88		0.65 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)					
I. Cvišić, J	I. Cvišić, J. Ćesić, I. Marković and I. Petrović: SOFT-SLAM: Computationally Efficient Stereo Visual SLAM for Autonomous UAVs. Journal of Field Robotics 2017.											
9	MULLS	***	<u>code</u>	0.65 %	0.0019 [deg/m]	0.08 s	4 cores @ 2.2 Ghz (C/C++)					
10	MULLS-SLAM	:::		0.67 %	0.0020 [deg/m]	0.1 s	4 cores @ 2.2 Ghz (C/C++)					
11	IMLS-SLAM	13.5		0.69 %	0.0018 [deg/m]	1.25 s	1 core @ >3.5 Ghz (C/C++)					

J. Deschaud: IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018.

第四题



提高精度: 去除outlier

去除各种无效点

找更优秀的特征点

提高速度:搜索结构

gpu加速

尺度金字塔



感谢各位聆听 / Thanks for Listening •

