

密级:公开

编号:18214872

中山大學

工程

硕士专业学位论文

基于语义信息的回环检测算法研究

Research on Loop Closure Detection Based on
Semantic Information

学位申请人: 朱亚琛

导师姓名及职称: 刘聪 副教授

专业、领域名称: 工程 (软件工程)

2020 年 6 月 5 日

中山大学硕士学位论文

基于语义信息的回环检测算法研究

Research on Loop Closure Detection Based on
Semantic Information

学位申请人： 朱亚琛

指导教师： 刘聪 副教授

专业名称： 工程（软件工程）

答辩委员会主席（签名）： 杨帆

答辩委员会委员（签名）： 成慧

田琪

谭宇

二〇二〇年六月五日

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：朱五琛

日期：2020年6月5日

学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文。

学位论文作者签名：朱五琛 导师签名：刘聪

日期：2020年6月5日

日期：2020年6月5日

论文题目：基于语义信息的回环检测算法研究

学 号：18214872

姓 名：朱亚琛

专 业：工程（软件工程）

导 师：刘聪 副教授

摘 要

近些年来，人工智能学科在各个领域都有着重要的研究价值，其中，作为高精尖科技代表的机器人科学也成为了重点研究方向。机器人首先要能够做到自主地感知周围环境信息并以此精确定位自身位置，这也是机器人能够自主导航移动的前提。所以机器人同时定位与建图（SLAM）是一项重要的研究课题。然而，若要进行高精度的 SLAM，一个高性能的回环检测模块是必不可少的，回环检测模块可以显著降低前端里程计带来的定位漂移误差。但是设计出一个在高召回率条件下仍有高检测精确率的回环检测算法是极具挑战性的。本文提出的方法以激光雷达作为主要传感器，通过检测环境中的语义信息，提取全局描述子和局部描述子的方式快速鲁棒地检测是否存在回环。

本文的工作主要包含以下几个方面：

（1）本文提出了一种在城市驾驶环境下仅使用激光雷达数据基于物体层面的回环检测方法。这个方法创新性的从场景中的语义信息来提取点云描述子，增强了点云描述子对描述物的描述能力。同时，本文的方法基于由粗到精的查找思想，提高了检测回环的效率。

（2）本文将图理论与语义信息相结合，提出了两种基于图构建的描述子，一个属于全局描述子，具有回环检测必要的旋转不变性，用来构建 kd-tree 并快速寻找和查询帧相似的一些回环候选帧；另一个属于局部描述子，用来计算两点云之间语义物体的一对一关系，处理两帧之间的数据关联问题。

（3）本文研究了其它四种传统回环检测方法并在大型公开数据集 KITTI 数据集与本文提出的方法进行大量了实验对比，从各个维度全方面的验证了本方法相比其他传统回环检测方法的先进性。

关键词： 同步定位与建图，回环检测，语义信息，激光雷达，机器人

Title: Research on Loop Closure Detection Based on Semantic Information

Student ID: 18214872

Name: Yachen Zhu

Major: Software Engineering

Supervisor: Associate Prof. Cong Liu

Abstract

In recent years, Artificial Intelligence has important research value in various fields. Among them, robotics, which is a representative of high technology, has also become a key research area. The robot must first be able to autonomously sense the surrounding environment information and accurately locate its own position, which is also the prerequisite for the robot to be able to autonomously navigate and move. Therefore, Simultaneous Localization And Mapping (SLAM) is an important research topic. However, to perform high precision SLAM, a high-performance loop closure detection module is essential. The loop closure detection module can significantly reduce the localization drift caused by the front-end odometry. But it is very challenging to design a loop closure detection algorithm with high detection accuracy under high recall. The method proposed in this thesis uses lidar as the main sensor to quickly and robustly detect whether there is a loop closure by detecting semantic information in the environment and then extracting global descriptors and local descriptors.

The work of this thesis mainly includes the following aspects:

(1) This thesis proposes a loop closure detection method based on object level using only LiDAR data in an urban driving environment.

This method innovatively extracts the point cloud descriptor from the semantic information in the scene, and enhances the point cloud descriptor's ability to describe the object. At the same time, the method in this thesis improves the efficiency of detecting loops based on the idea of searching from coarse to fine.

(2) Combining graph theory and semantic information, this thesis proposes two graph-based descriptors, one belongs to the global descriptor and has the necessary rotation invariance for loop detection, which is used to construct kd-tree and quickly find and loop candidate point clouds similar with the query one; the other belongs to the local descriptor, which is used to calculate the one-to-one correspondence relationship between the semantic objects between the two point clouds and handle the data association problem.

(3) This thesis studies four other traditional loop closure detection methods and conducts a large number of experimental comparisons with the method proposed in this thesis on the large public KITTI dataset, and verifies many aspects of this method compared with other traditional loop closure detection methods.

Keywords: Simultaneous localization and mapping, Loop closure detection, Semantic information, LiDAR, Robotics

目 录

摘 要.....	I
ABSTRACT	II
第 1 章 绪论	1
1.1 研究背景和意义.....	1
1.2 激光 SLAM 的研究现状.....	3
1.3 回环检测的研究现状.....	5
1.4 本文主要研究内容及创新点.....	8
1.5 本文的组织结构及章节安排.....	9
1.6 本章小结.....	10
第 2 章 激光 SLAM 基础研究	11
2.1 激光 SLAM 理论简介.....	11
2.2 激光雷达传感器.....	14
2.3 激光 SLAM 系统概述.....	15
2.4 本章小结.....	23
第 3 章 基于语义信息的回环检测算法研究	24
3.1 基于 NDT 的前端里程计.....	25
3.2 语义检测.....	27
3.3 图描述子生成.....	30
3.4 顶点匹配.....	32
3.5 几何一致性验证.....	34
3.6 本章小结.....	35
第 4 章 实验结果与分析	36
4.1 实验数据集简介.....	36
4.2 对比实验方法概述.....	36
4.3 实验参数设置.....	40
4.4 回环检测性能评估.....	41
4.5 噪声敏感度评估.....	43
4.6 初始位姿估计准确度.....	44
4.7 时间复杂度分析.....	48

4.8	本章小结.....	48
第 5 章	总结与展望.....	49
5.1	工作总结.....	49
5.2	研究展望.....	50
	参考文献.....	51
	附录 A 攻读硕士期间研究成果.....	55
	致 谢.....	56

第 1 章 绪论

本章主要内容为激光 SLAM 及回环检测的研究综述,从 SLAM 的背景知识开始,介绍激光 SLAM 及回环检测的相关知识,引出本论文的主要的研究内容,然后介绍本论文的创新点,最后一部分是本论文的结构组织简介。

1.1 研究背景和意义

现如今我国科技水平不断提升,高新技术产业蓬勃发展,人工智能俨然成为了当今的热门研究方向。这其中,为了解决我国劳动力不足,劳动力成本增加的问题,各大科技公司不断投入人力物力致力于机器替代人类工作的研究。机器人有着极为广泛的应用前景,它不仅可以在工业,农业,建造业发挥自己可以昼夜不停工作的优势,解放人类的双手,还可以在家政服务,快递物流等方面解放方便人类的日常生活。



(a)

(b)^①

图 1-1 SLAM 技术的实际应用

然而,若要实现机器人在各个应用领域中的成功落地,机器人必须要有实时

^① <http://news.iresearch.cn/content/2016/03/259391.shtml>

感知自身在周围环境中的位姿的能力。这也是实现机器人自主导航，无人驾驶的重要基础。同时定位与建图（Simultaneous Localization And Mapping, SLAM）是指机器人可以在未知环境下对自身定位并且可以增量式地构建周围的环境地图[1]。目前 SLAM 有两种主流解决方案，一个是基于相机的，一个是基于激光雷达的，尽管许多基于视觉的回环检测方法在近些年已经被提出和应用[2][3][4]，但是基于相机的方法有诸多局限性，它们可能会得到不可靠的回环检测结果尤其是在光照条件[5]或者视角变换剧烈的时候[6]。激光雷达传感器不像相机的工作原理那样会导致这些问题，激光雷达是通过生成高分辨率和精确的测量信息的 3D 点来感知周围环境的。它不但能在光照条件不好的情况下工作，而且可以提供比图像更多的结构性信息，同时这也提供了在激光雷达点云数据中做目标检测的基础。因此，基于激光雷达的回环检测方法成为了近年来的研究热点。

一个完整的基于激光雷达的 SLAM 方法主要包含四个模块：前端里程计，回环检测，后端非线性优化和建图。在进行 SLAM 期间，前端扫描匹配里程计不可避免地会累积漂移误差。这样建出来的的地图不是全局一致的。此时，回环检测技术就是机器人能够精确定位建图的关键。一个可靠的回环检测方法关于 SLAM 整体的定位准确度和建图效果是至关重要，它能够显著地提升 SLAM 系统的鲁棒性，消除前端里程计带来的漂移误差。

回环检测（Loop Closure Detection）是指机器人可以准确的判断出它曾经走到过的地方[6]。回环检测问题可以被理解为一个判断两个点云相似性的问题。匹配机器人当前获取的点云数据与存储在数据库中的历史点云信息便可以判断是否存在回环。匹配过程常常分为两步，点云描述子提取和点云描述子相似性度量。点云描述子是将点云中有辨识性的特征编码成一维的直方图向量，用于描述代表整个点云。一个好的点云描述方法可以有效表征整个点云，从而提升回环检测算法的性能。相似性度量一般选用欧氏距离，从而利用 k-dimension tree（kd-tree）数据结构来进行快速搜索[47]。

深度学习近年来在计算机视觉领域取得了重大突破，从而受到了人们的广泛关注。回环检测是一个依据激光雷达点云数据来进行地点识别的问题，其中也需要深度学习技术的辅助，从而在复杂的点云数据中提取出富含丰富信息的语义特征。传统的点云分割算法依靠人工设置的参数来提取特征，不仅工作量冗余而且提取出的物体常常带有噪声，不够精确，深度学习的引入高效解决了复杂场景下

从点云中提取语义信息的问题[12]。将深度学习与 SLAM 结合研究也成为了现在热门的研究方向。

1.2 激光 SLAM 的研究现状

SLAM 本身是由 SFM 发展变化而来的, SFM (Structure From Motion) 主要解决的是传统三维重建问题。SFM 所使用的传感器也可以是激光雷达和相机传感器, 不过 SFM 相比于 SLAM 它们的不同之处在于 SFM 是将传感器数据采集到存储器, 然后在离线的情况下处理。Smith, R 和 Cheeseman[35][35]在 1988 年最先提出了同时定位与建图的概念。由于 SLAM 技术无论在军事上的定位导航还是日常生活中的机器人应用都有着重大的发展前景, 国内外的学者们因此对 SLAM 中的定位算法、地图构建、特征匹配、回环检测、数据关联等问题展开了大量深入细致的研究。SLAM 领域与其他领域也有很多交叉, 如和机器人相关的计算机视觉, 图论和概率估计等问题也是近些年来火热的研究方向, SLAM 与这些学科正交叉发展。

基于激光雷达的 SLAM 方法根据整体优化框架的不同, 主要分为基于滤波方法的激光 SLAM 方法和基于非线性优化的激光 SLAM 方法。1988 年 Smith 等人[36]提出了 EKF-SLAM, 即基于扩展卡尔曼滤波的 SLAM 方法。扩展卡尔曼滤波本身在 SLAM 领域中就是一个经典的定位建图框架, 该方法将 SLAM 中运动模型和观测模型的误差假设为高斯噪声是这个方法能成功实践的一个重要前提。然而, 正是在实际应用中很多噪声不符合高斯分布的特性, 所以此方法在实际操作起来比较受限, 需要一些改进算法保护 EKF 能在特殊场景中成功运行。而且, 基于扩展卡尔曼滤波的激光 SLAM 方法所构建的环境地图并非传统栅格地图, 而是难以复用的特征点地图, 不利用使机器人构建出更高层面的路径规划与导航避障功能算法; 2002 年 Montemerlo 等人[37][38]提出了基于 EKF-SLAM 改进的 FastSLAM 方法, 该算法使用了粒子滤波 (Rao-Blackwellized Particle Filtering, RBPF) 替代扩展卡尔曼滤波算法进行前端里程计的位姿估计。粒子滤波相比于扩展卡尔曼滤波的一大优势是它不局限于运动模型和观测模型的高斯噪声假设, 而且对错误的位姿估计结果更加鲁棒。粒子滤波采用粒子对机器人的运动进行采样, 再通过观测模型计算出每个粒子的权重, 从而使用概率的方法得到更加精确的位姿估

计结果。但是，粒子滤波通常需要消耗极大的计算资源，因为它需要放出大量的粒子进行采样拟合来得到更加鲁棒的位姿估计结果，而且对于粒子滤波还有一个局限性在于难以避免的粒子耗散问题。粒子耗散指粒子滤波的重采样过程中可能会损失粒子多样性，使得重采样后有重要权重价值的粒子被错误剔除掉，影响拟合结果。针对这一问题，Montemerlo 等人[39]提出了 FastSLAM 2.0。Grisetti 等人[40]在 2007 年提出了基于 FastSLAM 改进的 Gmapping 方法以及解决粒子耗散问题的自适应重采样方法，即粒子数下降到一定阈值时再进行重采样，使重采样的次数适度减少，降低有价值的粒子被滤除掉的风险。然而此方法过于依赖前端里程计提供的初始位姿估计，初值的精度对建图的精度和最终的位姿估计结果有直接影响。而且 Gmapping 仅适用于水平地面、小场景下的 SLAM 如室内场景，而在室外尤其是地势起伏不平的区域进行建图定位则需要大量粒子才能对机器人的运动进行良好估计，但这样会造成极高的计算成本。

还有一类激光 SLAM 方法，是基于非线性优化的 SLAM 方法，这种方法也是现今流行 SLAM 方法的主流发展方向。1997 年 Lu F 等人[41]提出了图优化框架实现同时定位建图。此方法将前端里程计估计出的激光雷达关键数据帧当作图中的顶点，将关键帧之间的位姿约束关系当作边，在离线的情况下使用非线性最小二乘法，通过将累计误差均摊到各个顶点上以最小化建图过程中出现的累计误差。然而，这种方法没有意识到 SLAM 过程中所要优化计算的图是具有稀疏性的，导致算法具有非常高的计算复杂度。Konolige K 等人[42]在 2010 年提出了 Karto SLAM，此方法同样基于图优化框架，但它的先进之处在于它使用了非迭代乔列斯基分解方法对稀疏系统解耦运算从而大大降低了优化过程的计算量。2011 年 Kohlbrecher S 等人[43]提出了 Hector-SLAM 方法。Hector-SLAM 方法不需要使用里程计，而是使用高斯牛顿方法和帧到地图的方式来解决扫描匹配问题，所以该方法适用于无人机或者在地势不平的区域内建图。但 Hector-SLAM 方法对初值同样敏感，且由于该方法没有回环检测模块，使得它在大范围场景的建图效果并不十分理想。2016 年 Google 公司提出了 Cartographer 算法[44]，该方法是基于 Karto SLAM 改进而来的，Cartographer 不仅支持 IMU、激光雷达的多传感器融合定位，还使用了 CSM 方法避免暴力搜索时陷入局部最优值，但由于这样的策略会加大计算量，所以 Cartographer 提出了分支定界方法加速搜索回环的过程，减少计算。

上述基于非线性优化的激光 SLAM 方法都是 2D SLAM 方法。虽然 Google 同样维护了 Cartographer 的 3D SLAM 版本，但由于计算的复杂度较高，在大场景下容易产生内存不足的应用限制，算法并不完善。而 3D 激光雷达 SLAM 大多数也是沿用非线性优化框架的。伯克利大学提出了 3D 激光 SLAM 算法 BLAM^②(Berkeley Localization And Mapping)，它的前端里程计使用 GICP[45] 计算两帧点云间激光雷达的初步运动变换关系，然后利用帧到地图的匹配得到精确的位姿变换关系。后端构建因子图模型使用非线性优化方法对位姿进行优化。2014 年 Zhang J 等人提出了 LOAM (Lidar Odometry And Mapping) 算法[46]，此方法利用从点云帧里提取出的特征点进行点云扫描匹配，利用里程计得到初始的位姿估计，然后用帧到地图的匹配方式完成精准的定位建图。此方法具有较高的精准度并且在室内外场景均可以鲁棒运行，同时其算法复杂度较低，非常适用于机器人自主定位建图任务。虽然 LOAM 的定位建图比较精准，但其缺少回环检测模块，一旦出现较大的位姿估计误差很难得到一致性地图。对此，2018 年 Shan T 等人提出了 LeGO-LOAM[21]，其改进了 LOAM 提取特征的方式并利用两步计算优化了用于计算两帧间位姿变换的点云匹配算法，更重要的是它加入了后端非线性化优化模块，巧妙运用了前端提取出的点云特征进行快速回环检测，位姿估计效果相比 LOAM 更加鲁棒。

1.3 回环检测的研究现状

早期的 SLAM 是没有回环检测这个模块的，大部分是利用里程计推算出机器人在环境中的位姿同时生成完整的环境地图。然而，里程计每次进行点云的扫描匹配都是在两点云之间寻找一个最优的变换关系，并没有考虑到与历史各个位姿之间的联系，所以不可避免地会产生累计误差，顾名思义，SLAM 的过程越长，关键帧越多，误差也就累积得越大，尤其是在大范围场景下建图，累计误差会显著影响定位建图的质量，在运动较长时间后，机器人在里程计估出的位姿信息很可能与真实位姿相差甚远。回环检测模块便是针对 SLAM 中这一问题的解决办法。当机器人发现它来到了一个之前来过的地方时，现在的位姿关系和上次的位姿关

^② <https://github.com/erik-nelson/blam>

系可以构成一条特殊位姿约束关系, 累计误差便可以通过非线性优化方法平摊到各个位姿节点上, 从而得到符合一致性的地图。

然而, 在机器人定位建图的过程中, 检测出它是否重复经过过一个地方是非常具有挑战性的。在前端里程计不断估计机器人的运动状态的过程中, 机器人本身的位姿的不确定性会不断增加, 如果依赖于前端里程计的结果, 那么可能会因为噪声而难以检测到真正的回环的后果。所以一个独立于前端里程计的检测回环的方法是至关重要的。回环检测主要是通过检测时召回率为 100% 时回环检测算法的精确率大小。这样做的原因是因为在 SLAM 过程中, 回环检测算法若将不是回环的地方误判为是一个回环的情况, 要比将本身存在回环但因为算法性能原因没有检测到的情况有着更大的代价, 若在优化的过程中出现了前者的情况, 经过非线性优化后可能会使地图错乱, 甚至还不如不进行回环检测得到的地图准确。另一方面, 由于激光雷达传感器通常运行频率为 10 赫兹, 所以在经过同一地方时传感器会采集多个类似的点云数据, 在实际应用中, 此时回环检测模块只需要从这些点云帧中判断出一次这个地点存在真实的回环, 那么就可以达到修正累计误差的目的, 所以回环检测算法有较低的精确率也通常是可以接受的。

1.3.1 传统的基于激光雷达数据的回环检测

一种类型的方法使用局部点云特征来做场景识别。Bosse 和 Zlot [7] 从点云中随机选取的 keypoints 中提取区域形状描述子, 然后使用投票策略查找 keypoints 最近的邻居来识别地点。Steder 等人[8]将点云数据转换为距离图像, 然后提取每个 keypoint 的特征描述子向量。然后构建 kd-tree 来高效处理这个拥有很高维度的向量。为了达到一个更高的场景识别性能, 在他们的延伸工作中引进了 NARF 局部描述子和词袋模型匹配方法[9]。Johnson 等人提出了 Spin Image 方法[33], 利用了关键点邻域的局部表面形状。Huber 基于 Spin Image 提出了一种全局点云配准的方法[34][34], 该方法也可以用于回环检测。它通过 Spin Image 局部表面一致性来区分当前点云帧与历史点云帧是否匹配。

另一种传统的基于激光雷达数据的场景识别方法专注于提取点云的全局特征。并且这种全局描述子经常是以直方图的形式表示的, 比如 VFH[16], ESF[17] 和 Z-projection[15]方法。还有一种基于高斯分布变换 (Normal Distributions Transform, NDT) 算法的描述点云形状的方法[11]将点云离散为一个个立方体后,

可以从每个立方体的协方差矩阵确定每个体素单元的形状属性。之后通过组合拼接所有的体素单元,就可以获得代表整个点云的表面形状直方图作为全局描述子。Giseop Kim[14]提出了一种 Scan Context 全局描述子,它将每个点云先按角度分割成若干个扇区,然后再利用若干个同心环分割这些扇区得到许多离散的块。Scan Context 描述子将这些块按扇区和同心环数编码到 2D 矩阵中,矩阵中的值存储的是每个块中点云在竖直方向上的最大高度。由于在匹配过程中匹配这种 2D 矩阵类型的描述子需要大量的计算成本,因此 Kim 从 Scan Context 中按照同心环中包含三维空间点的数量提取了一个一维向量以构建 kd-tree 并加快了搜索过程。He 等人[13]在多视图 2D 平面上投影点云中的 3D 点,并将从这些平面派生的所有奇异矢量连接起来作为点云全局描述子。Rohling 等人[19]提出了点云中的点范围分布一维直方图作为全局点云描述子。然后,将 Wasserstein 距离用于比较直方图的相似性以进行位置识别。

1.3.2 使用更高层次的描述子进行回环检测

大体来讲,局部描述子缺乏对整个点云全局的表征能力,全局描述子容易陷入旋转不变性问题。所以 Shan 等人[21]创新性地从地面点中提取了边特征,从分割后地点中提取了面特征。然后使用 ICP 匹配[49]特征关系来找到回环。Dube 等人[20]提出了一个叫做 SegMatch 的场景识别方法,SegMatch 通过匹配点云聚类出的点云块从而得到一个更通用鲁棒的回环检测方法。然而,SegMatch 需要太多的计算资源以至于全部的回环检测框架运行频率只有 1 赫兹。

1.3.3 利用图匹配进行回环检测

图匹配在处理两帧间数据关联问题有着重要地位。大体上来说,图表征是一个用来描述物体和物体间拓扑关系的通用方法。利用这种方法,寻找两个相似场景中物体间的对应关系就可以转换为计算两幅图之间顶点和边的一一对应关系。不幸的是,寻找这个问题的精确解是一个 NP 难问题。Bailey 等人[22]生成了一个对应关系图并寻找这个对应关系图中的极大团来找到精确解。然而,这个方法仅适用于匹配的图中有较少的顶点。

为了避免大量的计算消耗,一个传统的解决办法是取而代之地使用近似解。一个方法是使用基于随机游走技术的核方法。一种尝试是使用基于随机游走技术的图核方法[23]。对于图中的每个顶点,在执行匹配步骤以找到顶点到顶点的对

应关系之后，它会生成大量的行走序列作为顶点描述子。此外，Fisher 等人[24]使用[25]中提出的图核方法比较关系图之间的相似度以进行场景识别。

1.4 本文主要研究内容及创新点

总的来说，传统的基于激光雷达的检测方法使用局部关键点或者其他的全局特征来提取点层面的点云描述子，然后匹配查询点云的描述子和历史所有点云的描述子来最终识别出已经来到过的地方。然而，这些方法在局部细节上关注太多而无视了更高层面的特征约束。不像通过几何点或其他点层面的特征来定义一个曾经到过的地方，人类会从一个更宏观的角度理解整个场景，换句话说，是通过识别场景中物体及它们之间在三维空间的相对位置关系来确定的[18]。

受到了人类定义一个场景的方式的启发，本文提出了一个全新的回环检测方法。这个方法使用场景中的语义物体来检测回环。首先利用获取到的这些语义物体的位置来形成两个描述子，其中一个是全局描述子，另一个是局部描述子。全局描述子用来快速寻找和查询帧相似的一些回环候选帧；局部描述子用来计算两点云之间语义物体的一对一关系。最后，使用几何一致性验证步骤来确定是否存在回环。

本文是第一个使用物体层面的语义信息来在 3D 激光雷达数据检测回环的工作。本文主要创新点及贡献如下所示：

- 本文提出了 GOSMatch，一个在城市驾驶环境下仅使用激光雷达数据基于物体层面的回环检测方法。本方法结合了具有特殊性质的语义信息来提取点云描述子，不仅增强了点云描述子的描述能力，还提高了某一点云数据相较于其他点云的辨识度。此外，GosMatch 方法基于由粗到精查找的思想，先粗略的找出与当前点云相似的回环候选帧列表，再通过更加细致的比对，筛选出与查询点云最相近的历史点云并加以判断，提高了回环检测的效率。
- 本文提出了一组基于语义信息间的空间位置关系的全局描述子和局部描述子。它们利用了激光雷达传感器良好的测距性能，使用语义信息间的欧几里得距离形成描述子，使得全局描述子具有旋转不变性，同时全局描述子是一个一维向量，可利用 kd-tree 数据结构加速最近邻搜索算法从而提升查找回环的效率；局部描述子用于描述点云中的某一语义信息，可利用描述子之间

的距离可以计算出不同点云之间同一语义物体的对应关系，处理两帧间的数据关联问题。

- 本文对比研究了 VFH、Z-projection、M2DP 和 Scan Context 四种传统回环检测方法并在大型自动驾驶公开数据集 KITTI 数据集上与本文提出的方法 GosMatch 进行了回环检测实验对比，使用精确率和召回率指标衡量各个算法的性能。之后又分别从噪声敏感度、初始位姿估计准确度和时间复杂度等三方面的实验，验证了本方法突出的抗噪声能力、准确的回环初始位姿估计以及高效的执行效率。

1.5 本文的组织结构及章节安排

本文正文包含五章内容，正文章节组织方式如下：

第一章是激光 SLAM 和回环检测的综述，介绍了部分 SLAM 和回环检测的知识和相关研究背景，分析了 SLAM 和回环检测的发展历程，阐述了 SLAM 和回环检测的研究现状，在这些知识的基础上引出了本文的主要研究内容和创新点，最后是本文的组织结构和章节安排。

第二章是对激光 SLAM 和回环检测技术的研究，介绍了激光 SLAM 的基本框架和回环检测的基本原理。同时分析了激光雷达传感器的工作原理，阐述了进行快速的回环检测需要的数据结构以及图优化框架的基本数学原理。

第三章是 GOSMatch 方法的具体实现，首先从高层次的视角介绍了 GOSMatch 算法的工作流程，然后分析了基于 NDT 算法的前端里程计的实现，接着从语义检测、图描述子生成、节点匹配和几何一致性检测四个模块详细地说明了 GOSMatch 的工作原理。

第四章是实验结果与分析，将 GOSMatch 算法与其他四个传统回环检测在大型公开数据集 KITTI 数据集上进行了对比验证，从准确率和精确率两个方面验证了 GOSMatch 方法的先进性。接着又从对噪声的敏感度、初始位姿估计的准确度和时间复杂度三个方面详细分析了 GOSMatch 算法。

第五章是总结和展望，从总体上回顾了 GOSMatch 算法的工作原理，阐述了使用语义信息进行回环检测的优越性，并总结了 GOSMatch 算法的优势和不足，为将来的研究工作奠定基础。

1.6 本章小结

本章是激光 SLAM 和回环检测的综述章，介绍了课题的研究背景和部分 SLAM 知识，阐述了激光 SLAM 和回环检测的技术联系。介绍了激光 SLAM 和回环检测的研究现状。然后针对传统回环检测方法的局限性引出了本文的主要研究内容及创新点，最后是本文的组织结构和章节安排。

第 2 章 激光 SLAM 基础研究

本章为激光 SLAM 系统概述，首先用数学语言阐述了 SLAM 解决的即时定位建图问题。然后介绍了激光雷达传感器的测距工作原理，接着阐述了激光 SLAM 的工作流程以及回环检测技术的重要地位，最后介绍了图优化框架的基础知识。

2.1 激光 SLAM 理论简介

SLAM 技术主要解决的是机器人在未知环境下自主定位建图的问题。当一个机器人处在未知环境中，它要利用各个传感器反馈的信息确定自己的具体位姿，并同时生成构建出周围环境的地图。SLAM 技术主要研究两个大方向，一是精确感知定位机器人本身在环境中的位姿，另一个是得到一个可以精确描述周围环境的空间地图。然而，SLAM 领域存在许多具有挑战性的问题。由于机器人携带的传感器本身会有测量上的难以避免的误差，机器人在运动过程中感知环境就会有误差干扰，为定位带来了越来越大的不确定性，导致机器人难以构建出与真实空间环境相近的地图。虽然目前一些方法在室内环境等有限范围且静态的场景有鲁棒的建图定位表现，但是在室外有动态物体的大范围场景下定位建图依然是一个待解决的难题。

2.1.1 SLAM 的数学描述[52]

通常一个机器人在时刻 t 的位姿用 \mathbf{x}_t 来表示，机器人在三维空间中运动，则 \mathbf{x}_t 是一个 6 自由度的 4×4 变换矩阵 T ，其中包含了机器人在 t 时刻在世界坐标系中的三维坐标 x, y, z 和一个描述机器人朝向的 3×3 旋转矩阵 R 。机器人在一系列运动过程中，将连续的运动通过传感器在多个离散的時刻采集的数据进行描述，机器人从开始运动到结束运动，它在各个時刻的位姿可以表示为：

$$\mathbf{X} = \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$$

其中 \mathbf{x}_0 为已知的初始位姿， T 为机器人结束运动的時刻。

通常，机器人会携带可以测量运动状态的传感器，如惯性测量单元或者轮式里程计。根据这些传感器的读数便可以推算出机器人在两時刻之间的运动过程，

这个读数不一定是直接反映俩位姿之间的几何变换关系，而一般是速度，加速度和角速度。经过一定的数学推导，这些数据总能抽象出机器人在 $t - 1$ 时刻到 t 时刻运动控制方面的信息 \mathbf{u}_t 。一系列运动过后，可得运动控制序列：

$$\mathbf{U} = \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$$

这个序列描述了机器人从初始位姿 \mathbf{x}_0 一直到 \mathbf{x}_k 过程中的所有相对运动状态，使用 \mathbf{U} 能够恢复机器人的运动轨迹。然而，在实际应用中，这些运动控制信息是有噪声 \mathbf{W} 干扰的，因此仅靠 \mathbf{U} 分析运动轨迹会与真实轨迹有偏差。根据机器人在 $t - 1$ 时刻的位姿及 $\mathbf{u}_t, \mathbf{w}_t$ ，我们可以预测机器人 t 时刻的状态，并得出运动方程：

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t) \quad (2.1)$$

机器人在周围环境中运动时，在 t 时刻可以感知到环境中的路标 y_j ，同时可以得到路标 y_j 与当前机器人状态 \mathbf{x}_t 之间的测量关系 \mathbf{z}_t ，机器人经过一段时间的运动，可得到测量序列：

$$\mathbf{Z} = \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k$$

在每次测量机器人位姿与路标之间的距离关系时，同样也不可避免的会产生测量误差 \mathbf{v}_t ，与运动方程相对应的，我们可以从测量序列中抽象出一个测量方程：

$$\mathbf{z}_t = h(y_j, \mathbf{x}_t, \mathbf{v}_t) \quad (2.2)$$

SLAM 过程可以使用概率问题进行描述，由运动方程模型可以得到 $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ ，即机器人在已知 $t - 1$ 时刻下自身的位姿和由 $t - 1$ 到 t 时刻的运动控制信息 \mathbf{u}_t 的条件下，位姿 \mathbf{x}_t 的概率分布；由观测方程模型可以得到 $p(\mathbf{z}_t | \mathbf{x}_t)$ ，即机器人在已知当前 t 时刻的位姿的条件下观测到 \mathbf{z}_t 的概率分布。利用两个模型得到的概率分布，我们可以使用数学方法贝叶斯滤波推算出机器人在任一时刻下的位姿。

2.1.2 贝叶斯滤波[53]

机器人在 $t - 1$ 时刻通过传感器能够观测到一帧外部环境数据，然后运动一定时间到达地图中一个新的位置，此时通过观测外部环境又可以得到一组观测数据，同时可以得到一组运动控制信息，表示两地之间的位置之差。然而，仅通过观测数据计算两位姿之差和仅通过运动控制信息计算两位置之差都不够准确的，需要一种方法将两种信息融合更新位姿，已得到一个置信度更高的位姿更新值。一个常用的解决算法就是贝叶斯滤波算法 (Bayes Filter Algorithm) [53]。贝

叶斯滤波算法根据观测数据和运动控制数据的概率分布，通过预测和修正两步，可以计算出机器人位姿状态的后验概率分布。

贝叶斯滤波算法是一个递归算法，即时刻 t 的机器人状态的后验概率分布是根据时刻 $t - 1$ 机器人状态的后验概率分布计算的。贝叶斯滤波算法的伪代码如下表 1-1 所示：

表 1-1 贝叶斯滤波算法伪代码

Algorithm 1 Bayes_filter ($Bel(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t$):

1. *for all* x_t *do*
2. $\bar{Bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) Bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$
3. $Bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \bar{Bel}(\mathbf{x}_t)$
4. *endfor*
5. *return* $Bel(\mathbf{x}_t)$

$Bel(\mathbf{x}_t)$ 表示综合了所有过去的观测数据 $\mathbf{z}_{0:t}$ 和所有运动控制信息 $\mathbf{u}_{0:t}$ 的条件下，机器人位姿状态的后验概率分布； $\bar{Bel}(\mathbf{x}_t)$ 表示没有综合当前观测数据 \mathbf{z}_t 的条件下机器人状态的后验概率。由 $Bel(\mathbf{x}_{t-1})$ 计算 $\bar{Bel}(\mathbf{x}_t)$ 的过程称为预测，由 $\bar{Bel}(\mathbf{x}_t)$ 计算 $Bel(\mathbf{x}_t)$ 的过程成为修正。

表中第 3 行展示了预测步骤的计算过程。机器人状态的预测置信度 $\bar{Bel}(\mathbf{x}_t)$ 根据 $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ 与 $Bel(\mathbf{x}_{t-1})$ 的乘积对 $t - 1$ 时刻机器人的状态 \mathbf{x}_{t-1} 进行积分求得；第 4 行展示了修正步骤的计算过程。机器人在 t 时刻的状态后验概率 $Bel(\mathbf{x}_t)$ 最终由 n 、在 \mathbf{x}_t 的条件下观测到 \mathbf{z}_t 的概率及预测置信度 $\bar{Bel}(\mathbf{x}_t)$ 的乘积得到。其中 n 是一个归一化常数，因为 $p(\mathbf{z} | \mathbf{x})$ 与 $\bar{Bel}(\mathbf{x}_t)$ 的乘积通常不是一个概率，需要数学上的归一化常数 n 来将其总和凑成 1，使其符合概率学性质。

贝叶斯滤波算法本身不是一个实际可用的算法，在实际应用中不能编程实现，它仅是表达了一个可行的数据融合滤波的算法思想。由于它是基于概率学的，在实际使用中可以通过近似的方法得到后验概率分布，如卡尔曼滤波、无迹卡尔曼滤波、粒子滤波等算法。实际上这些算法都是基于贝叶斯滤波的思想衍生而来的，只是这些不同的滤波算法有着不同的鲁棒性、运行效率和准确性。贝叶斯滤波是基于马尔科夫假设的，实现贝叶斯滤波的前提是机器人在每一时刻的状态 \mathbf{x} 都包

含了在此之前所有的历史状态信息。这个假设虽然是合理的，但是在实际应用中马尔科夫性仅是实际场景的近似，在某些极端情况下是不满足这个前提假设的。

2.2 激光雷达传感器



图 2-1 激光雷达传感器^③

激光雷达传感器主要使用竖排多个红外激光束来实现感知功能。在工作时，激光雷达会以一定频率自转，使得红外激光束可以扫描到周围 360° 的环境。激光束打到障碍物时，由于光的反射效应，会有一部分光反射回激光雷达传感器上，这使得激光雷达可以根据 TOF 测距原理准确地测量出传感器与周围障碍物间的距离。同时，不同质地的障碍物反射光的能力也不相同，激光雷达也可以根据反射强度的不同标记不同介质的障碍物。TOF 测距原理如图 2-2 所示：

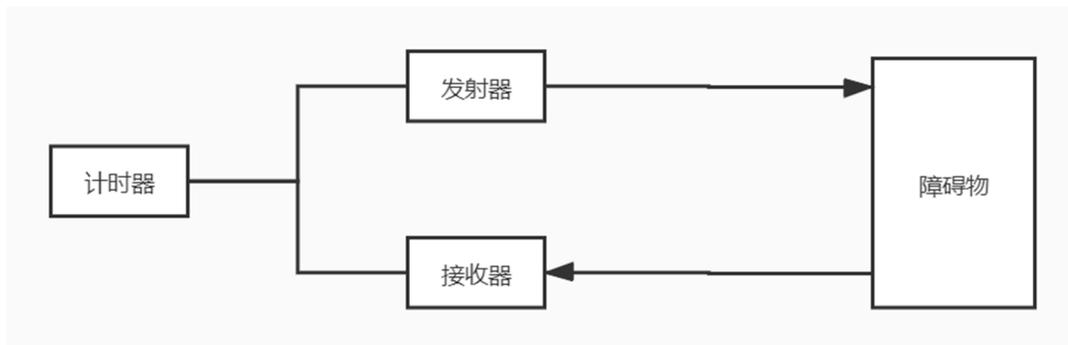


图 2-2 TOF 工作原理

^③ <https://velodynelidar.com/products/hdl-64e/>

激光雷达上的计时器会记录激光雷达发射出激光束的时间 T_{start} ，光束通过距离 D 打到障碍物上，根据光的反射原理，反射的光再次通过距离 D 返回到激光雷达上，同时计时器记录下此刻的返回时间 T_{end} ，依据发送时间和返回时间的时间间隔 $\Delta T = T_{end} - T_{start}$ ，可以根据公式 $D = c * \frac{\Delta T}{2}$ 计算出激光雷达和障碍物之间的距离 D 。

本文所使用的激光雷达是 Velodyne 公司出产的 HDL-64E 型号激光雷达，如上图所示。该型号激光雷达拥有 64 条激光线数，每根激光扫描束的探测范围达 120m，并且探测误差小于 2cm；每根扫描束旋转 360° 后可以生成最多 4500 个点，即最大水平角分辨率为 0.08° ；垂直视野范围为 26.8° ($-24.8^\circ \sim +2^\circ$)，垂直角分辨率为 0.4° 。Velodyne HDL-64E 激光雷达每秒钟可生产 220 万个 3 维数据点，可以用高分辨率点云描述周围环境信息，为识别场景中的语义物体提供了基础，所以该款雷达在无人驾驶汽车上也有着广泛应用。

2.3 激光 SLAM 系统概述

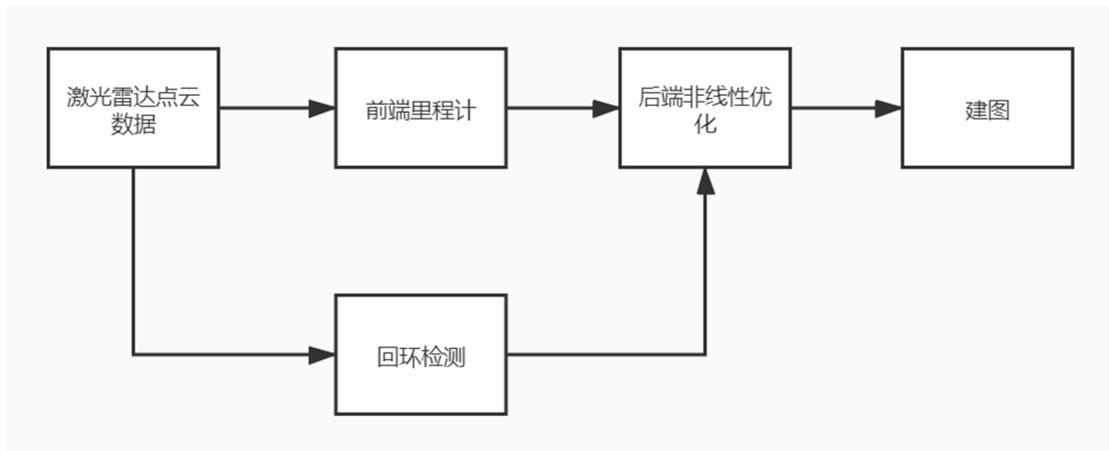


图 2-3 激光 SLAM 工作流程图

2.3.1 前端里程计

里程计，顾名思义，就是能够获取机器人走过的距离以及各个时刻机器人所处的姿态信息的功能，类似汽车上的里程仪表。激光雷达里程计（Lidar Odometry, LO）主要通过配准两相邻激光点云数据从而估计机器人的运动状态，其本质就是不断求解激光雷达在三维空间中的变换矩阵。如图所示，图(b)是图(a)中的激光雷达向前移动了一段距离后采集到的点云数据，虽然通过肉眼的观

察可以大致判断出两幅图之间的位姿变换关系，但是我们无法轻易地获取旋转角度和平移距离的具体数值，而在实际应用中，激光里程计就可以计算出相对准确的旋转角度和平移距离，从而大致估计出激光雷达的运动。

若要计算出两帧之间激光雷达传感器的运动，那么就必须要用到点云配准方法。最常见的方法是Besl和Mckay在1992年提出的ICP(Iterative Closest Point)算法。在给定激光雷达前后获取的两帧点云数据 P, Q 下，ICP 可以迭代地求解出两帧点云间的位姿变换关系从而将两帧点云配准。假设前后两帧点云中 3 维数据点的个数分别为 N_p, N_Q ，那么 ICP 算法最终就是要求解出旋转矩阵 R 和平移矩阵 t ，使得

$$q_i = R * p_i + t \tag{2.3}$$

其中 $q_i \in Q, p_i \in P$ 。

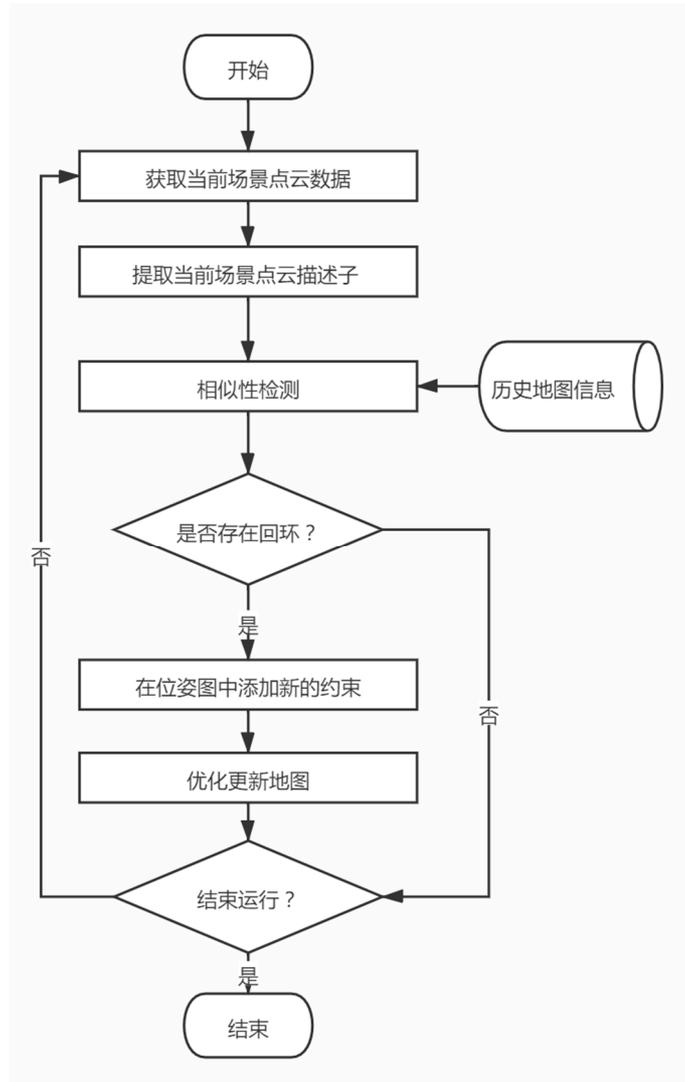


图 2-4 回环检测基本流程图

2.3.2 回环检测

回环检测的基本流程如图 2-4 所示。激光雷达获取当前场景下的点云数据，然后从点云数据中提取点云描述子以描述整个场景。之后利用历史点云生成的地图信息与当前点云描述子进行相似性检测判断是否存在回环，若发生了回环则在位姿图中加入新的约束边，并优化更新地图。

2.3.2.1 K 近邻搜索

使用全局描述子进行帧到帧的回环检测的一般流程是先从点云中提取点云全局描述子，这个全局描述子一般是一个 n 维向量。至此，点云帧之间是否存在回环的问题就可以转换为点云描述子之间的相似性问题。将当前帧之前所有的历史帧描述子全部保存起来，利用 K 近邻搜索 (K-Nearest Neighbour, KNN) 的方式从历史帧描述子数据库中进行快速查找到与当前帧全局描述子最相近的 K 个点云描述子，根据这 K 个点云描述子与当前帧全局描述子之间的距离是否满足人工设定好的阈值，从而判断出当前帧与历史帧之间是否存在回环。

描述子之间的距离有很多种度量方法，最常见的是欧几里得距离，以点 $x = (x_1, \dots, x_n)$ 和点 $y = (y_1, \dots, y_n)$ 为例：

欧几里得距离计算公式：

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (2.4)$$

$$= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.5)$$

曼哈顿距离计算公式：

$$d(x, y) = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \quad (2.6)$$

$$= \sum_{i=1}^n |x_i - y_i| \quad (2.7)$$

切比雪夫距离计算公式：

$$d(x, y) = \max_i (|x_i - y_i|) \quad (2.8)$$

余弦相似度计算公式：

$$d(x, y) = \frac{xy}{|x||y|} \quad (2.9)$$

$$= \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (2.10)$$

2.3.2.2 基于 kd-tree 的 KNN 搜索实现

实现 KNN 算法最简单的方法就是使用枚举法，也就是暴力求解当前帧全局描

述子到所有历史帧全局描述子之间的距离，从中选取距离最近的 K 个样本。这种算法虽然很容易实现，但是其时间复杂度明显过高，在实时性要求较高的 SLAM 回环检测任务中难以达到满意的时间效率。而且在每次到来新的一帧查询点云时都要遍历所有历史数据才能得到 K 个距离最近的候选帧，会对计算资源产生极大的消耗。

为了解决时间复杂度过高的问题，本文使用了 kd-tree 数据结构来加速搜索与当前帧邻近的候选帧。kd-tree 基于二分思想，使用二叉树结构实现，将数据逐层划分。一颗含有 N 个元素的 kd-tree 的高度只有 $\log N$ 层，极大地减少了搜索时需要查找的范围。

假设使用的全局描述子是一个 128 维的向量，包含所有历史帧全局描述子的 kd-tree 的构建过程如下：

1. 计算所有描述子在每一维上的数据方差，选择出方差最大的那一维 d ，因为方差大说明在这一维度上的数据比较发散，从这一维度上进行划分可以得到一个相对平衡的 kd-tree。

2. 在 d 维上找到所有数据的中位数 m ，随后根据 m 对数据集进行划分，将 d 维上比 m 小的数据划分到左子树，比 m 大的数据划分到右子树。

3. 对左右子树重复步骤 1 和 2，直到所有形成的左右子集合不能再被划分为止。

2.3.2.3 基于 kd-tree 的最近邻查找算法

1. 查询帧全局描述子 Q 从 kd-tree 的根节点开始进行最近邻查找，由上至下根据和树节点的比较结果访问 kd-tree 直到访问到叶子结点为止。 Q 在与树中节点进行比较时是使用 Q 数据中第 k 维度上的数值与树中中值 m 作比较，如果 $Q(k) > m$ ，则访问右子树，否则访问左子树。当访问到叶子结点时，计算出 Q 与叶子结点数据的距离，并记下当前最近邻点和其最短距离 dis 。

2. 回溯。回溯是为了判断未被访问的子树分支里是否存在与 Q 更接近的数据。若 Q 与它的兄弟子树之间的距离小于 dis ，则该分支里存在某一数据离 Q 更加接近。因此访问该子树，重复步骤 1，更新最近邻点和最短距离 dis 。如果 Q 与它的兄弟子树之间的距离大于 dis ，则另一分支内不存在更加接近 Q 的数据。

回溯过程自下而上，直至回溯到 kd-tree 的根节点为止。

2.3.3 后端图优化

图优化方法目前是 SLAM 主流的后端优化方法。基于滤波方法的 SLAM 是基于马尔科夫假设的，它仅考虑了相邻时刻机器人状态的联系，并认为前一刻的状态包含了所有历史信息。而图优化方法考虑了当前状态和所有历史状态的联系，并使用非线性优化方法来优化当前状态和所有历史状态。

2.3.3.1 非线性最小二乘问题

最小二乘就是最小化模型预测值和样本之间的误差，即

$$\min e(x) = \min \frac{1}{2} \sum_{i=0}^n (y_i - f_i(x))^2 \quad (2.11)$$

其中，其中 y_i 是第 i 个样本的真实值， $f_i(x)$ 是第 i 个样本的模型预测值。在 SLAM 问题中， y_i 和 $f_i(x)$ 对应于观测值和状态的估计值， $e(x)$ 是描述两者之间误差的函数。通过最小化 $e(x)$ 我们就可以得到一个与真实轨迹更加接近的函数模型。

若 $e(x)$ 的形式比较简单，那么可以通过直接求导的方式求得函数的解析解，此方法无需迭代过程，可直接得到最优估计。令导数为 0，求其极值可得：

$$\frac{d e(x)}{d x} = 0 \quad (2.12)$$

求得值可能是极大值、极小值或者鞍点，稍加验证便可得到最优解。但如果 $e(x)$ 的形式非常复杂，正如在 SLAM 问题中，机器人的位姿状态是使用旋转平移矩阵来表达的，他们的运算形式是基于李代数的，求导过程非常复杂。对于这样的问题，通用的数学方法是使用迭代的方式从初值一步步找到一个可以满足误差条件的局部最优解，并认为这个解即是最优解。基本的求解步骤为：

1. 给定一个初值 \mathbf{x}_0 。
2. 对于每一次迭代，根据梯度下降的方向计算一个增量 $\Delta \mathbf{x}_k$ ，使得目标函数 $e(\mathbf{x}_k + \Delta \mathbf{x}_k)$ 可以达到极小值。

3. 若 $\Delta \mathbf{x}_k$ 小于阈值 α ，算法终止，否则更新 $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$ ，返回步骤 2。

算法通过迭代更新的方式求解避免了对目标函数直接求导的过程，将不可解问题转化成了可解问题。在这个更新的过程中，不断计算增量 $\Delta \mathbf{x}_k$ ，直到在某次迭代时它满足了阈值限制，完成了对 \mathbf{x} 的最优估计。但是本方法也有一个局限性就是初值对求解的结果有着很大的影响，一个较好的初值可以让算法迭代更新到一个更好的局部最优解上。

2.3.3.2 高斯牛顿法和列文伯格-马夸尔特算法

高斯牛顿法是用来解决非线性最小二乘问题的算法，它是基于牛顿法改进得到的。它的基本思想是先通过一阶泰勒展开使得非线性函数 $f(\mathbf{x})$ 部分线性化。

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} \quad (2.13)$$

其中 $\mathbf{J}(\mathbf{x})$ 为雅可比矩阵，是函数 $f(\mathbf{x})$ 关于变量 \mathbf{x} 的一阶导数。与上述提到的非线性优化方法的思路一致，高斯牛顿法也是通过寻找一个合适的增量 $\Delta\mathbf{x}$ 使得函数 $\|f(\mathbf{x} + \Delta\mathbf{x})\|^2$ 最小。按照此思想可得新的目标函数：

$$\Delta\mathbf{x}^* = \operatorname{argmin}_{\Delta\mathbf{x}} \frac{1}{2} \|f(\mathbf{x} + \Delta\mathbf{x})\|^2 \quad (2.14)$$

该方程将求解 $f(\mathbf{x})$ 的最小值转换成了求解 $f(\mathbf{x} + \Delta\mathbf{x})$ 的最小值，将新的目标函数展开：

$$\frac{1}{2} \|f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}\|^2 = \frac{1}{2} (f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x})^T (f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}) \quad (2.15)$$

$$= \frac{1}{2} (\|f(\mathbf{x})\|_2^2 + 2f(\mathbf{x})^T \mathbf{J}(\mathbf{x})\Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})\Delta\mathbf{x}) \quad (2.16)$$

为了求这个函数的极值，我们需要对其求导，并令导数等于 0：

$$2\mathbf{F}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + 2\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})\Delta\mathbf{x} = 0 \quad (2.17)$$

整理可得：

$$\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})\Delta\mathbf{x} = -\mathbf{J}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) \quad (2.18)$$

可简化为：

$$\mathbf{H}\Delta\mathbf{x} = \mathbf{g} \quad (2.19)$$

其中 $\mathbf{H} = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})$ ， $\mathbf{g} = -\mathbf{J}(\mathbf{x})^T \mathbf{f}(\mathbf{x})$ ， \mathbf{H} 矩阵其实是对海塞矩阵的一种近似。

基于方程 (2.19)，可以得到高斯牛顿法的基本步骤：

1. 设定初始值 \mathbf{x}_0
2. 对于 k 次迭代，求出对应的雅可比矩阵 $\mathbf{J}(\mathbf{x}_k)$ 和误差 $f(\mathbf{x}_k)$
求解增量方程 $\mathbf{H}\Delta\mathbf{x} = \mathbf{g}$ 。
3. 如果解出的 $\Delta\mathbf{x}$ 小于阈值则算法终止，否则更新 $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$ ，重复步骤 2。

对增量方程的求解过程是在整个算法中最重要的部分。然而，若要使得求出的解为函数局部最小值，那么 \mathbf{H} 矩阵必须是正定的。但是在实际应用时， \mathbf{H} 矩阵很可能是一个半正定矩阵，导致算法不收敛或者计算出的结果不准确。针对这一问题，在 SLAM 中的优化方法我们通常选用列文伯格-马夸尔特算法。

列文伯格-马夸尔特算法 (Levenberg-Marquardt Method) 简称 LM 算法，它

基于信赖区域理论对高斯牛顿法进行了改进。高斯牛顿法为了线性化将函数进行一阶泰勒展开。然而这个展开带来的效果只是在展开点小范围内比较好，也就是说这种求解方式适用于在展开点附近近似，所以 LM 算法加入了信赖区域方法约束这个范围，它设定了一个阻尼因子 ρ ：

$$\rho = \frac{f(x+\Delta x)-f(x)}{J(x)\Delta x} \quad (2.20)$$

该阻尼因子是衡量近似的函数 $f(x + \Delta x)$ 和真实函数 $f(x)$ 之间的差异大小。如果 ρ 的值过大，就缩小信赖区域范围；如果值过小，就放大信赖区域范围。LM 算法的基础步骤如下所示：

1. 设定迭代初始值 x_0 及一个信赖区域优化半径 μ
2. 对第 k 次迭代，求解优化方程：

$$\min \frac{1}{2} \|f(x) + J(x)\Delta x_k\|^2, \quad s. t. \|\Delta x_k\|^2 \leq \mu \quad (2.21)$$

3. 根据计算 ρ 的公式算出 ρ 的大小。
4. 如果 $\rho > \frac{3}{4}$ ，则 $u = 0.5u$ ；如果 $\rho < \frac{1}{4}$ ，则 $u = 2u$ 。如果 ρ 大于阈值 ε ，

则认为本次迭代结果可信，更新 $x_{k+1} = x_k + \Delta x_k$ 。

5. 判断结果是否已达到收敛条件，如果收敛则算法终止，否则返回到第 2 步重复计算。

2.3.3.3 图优化理论简介[48]

图优化方法将 SLAM 过程中的优化问题使用图的形式表达，其中图的顶点表示机器人的位姿状态，图中的边表示两位姿之间的变换关系，这样的图在图优化方法里称为位姿图。

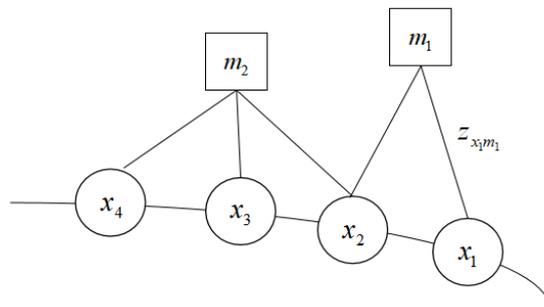


图 2-5 机器人运行轨迹及路标位姿图

2.3.3.4 构建图模型

SLAM 的前端里程计可以给出两个位姿的变换关系，这个位姿变换关系可以在图中标识为边。在构建 SLAM 图时，机器人在任一 k 时刻都能根据观测方程：

$$\mathbf{z}_k = h(\mathbf{x}_k) \quad (2.22)$$

返回传感器的测量数据，但是因为传感器在测量时有噪声存在， \mathbf{z}_k 不可能和 $h(\mathbf{x}_k)$ 完全相等，它们之间存在误差变量：

$$\mathbf{e}_k = \mathbf{z}_k - h(\mathbf{x}_k) \quad (2.23)$$

此时， \mathbf{x}_k 为待优化变量，优化目标函数为 $\min f(\mathbf{x}_k) = \|\mathbf{e}_k\|$ ，通过 LM 算法进行非线性求解即可得到 \mathbf{x}_k^* 使得误差最小。

在 SLAM 图中，观测方程构成的边有很多种形式，如机器人两位姿之间的变换关系或者机器人在位姿 \mathbf{x}_k 出使用激光传感器测量出了当前位姿与某路标点的距离和角度信息。在这里统一抽象成观测方程来表达。下图展示了机器人运动一段时间后形成的 SLAM 图， $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ 表示了机器人在各个时刻的位姿， $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \dots, \mathbf{m}_n\}$ 表示机器人在运动过程中观测到的路标集合。 $\mathbf{z}_{\mathbf{x}_i \mathbf{m}_j}$ 表示了机器人在 \mathbf{x}_i 状态下对路标 \mathbf{m}_j 的观测。对于机器人两个相邻的位姿之间的观测，使用 \mathbf{z}_{ij} 表示两位姿变换关系的均值，使用 $\mathbf{\Omega}_{ij}$ 表示两位姿变换关系的信息矩阵。在这里，信息矩阵指的是协方差矩阵的逆，信息矩阵的对角项表示了对对应误差的重要程度，值越大，代表对应的权重就越大，一般是由激光传感器本身设定好的参数得到。 \mathbf{z}_{ij} 是由 SLAM 前端里程计推算得到。

2.3.3.5 优化图模型

在构建完包含了全部位姿关系的 SLAM 图之后，全图的目标函数可以表示为

$$\min F(\mathbf{x}) = \sum_{k=1}^n \mathbf{e}_k(\mathbf{x}_k)^T \mathbf{\Omega}_k \mathbf{e}_k(\mathbf{x}_k) \quad (2.24)$$

由于机器人位姿通过旋转平移的变换属于非线性变换，所以求解方程(2.24)属于非线性最小二乘问题。一般都会选择 LM 算法对其进行数值求解。

假设给定一个初始点 $\widetilde{\mathbf{x}}_k$ 和一个增量 $\Delta \mathbf{x}$ ，则原误差 $\mathbf{e}_k(\widetilde{\mathbf{x}}_k)$ 变为 $\mathbf{e}_k(\widetilde{\mathbf{x}}_k + \Delta \mathbf{x})$ ，然后对其进行一阶泰勒展开：

$$\mathbf{e}_k(\widetilde{\mathbf{x}}_k + \Delta \mathbf{x}) \approx \mathbf{e}_k(\widetilde{\mathbf{x}}_k) + \frac{d\mathbf{e}_k}{d\mathbf{x}_k} \Delta \mathbf{x} = \mathbf{e}_k + \mathbf{J}_k \Delta \mathbf{x} \quad (2.25)$$

其中 \mathbf{J}_k 雅克比矩阵，是 \mathbf{e}_k 关于 \mathbf{x}_k 的一阶导数。公式(2.25)通过在估计点进行泰勒展开做了线性假设，在 $\Delta \mathbf{x}$ 较小的条件下使用一阶导数逼近真实函数。

同样将目标函数展开可得：

$$\begin{aligned}
F_k(\tilde{\mathbf{x}}_k + \Delta\mathbf{x}) &= \mathbf{e}_k(\tilde{\mathbf{x}}_k + \Delta\mathbf{x})^T \boldsymbol{\Omega}_k \mathbf{e}_k(\tilde{\mathbf{x}}_k + \Delta\mathbf{x}) \\
&\approx (\mathbf{e}_k + \mathbf{J}_k \Delta\mathbf{x})^T \boldsymbol{\Omega}_k (\mathbf{e}_k + \mathbf{J}_k \Delta\mathbf{x}) \quad (2.26) \\
&= \mathbf{e}_k^T \boldsymbol{\Omega}_k \mathbf{e}_k + 2\mathbf{e}_k^T \boldsymbol{\Omega}_k \mathbf{J}_k \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{J}_k^T \boldsymbol{\Omega}_k \mathbf{J}_k \Delta\mathbf{x} \\
&= \mathbf{C}_k + 2\mathbf{b}_k \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}_k \Delta\mathbf{x}
\end{aligned}$$

其中 \mathbf{C}_k 就是在为加入增量 $\Delta\mathbf{x}$ 时目标函数的取值，所以对因变量的初始点加了一个增量 $\Delta\mathbf{x}$ 后，目标函数的变化为：

$$\Delta F_k = 2\mathbf{b}_k \Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H}_k \Delta\mathbf{x} \quad (2.27)$$

在这里重新声明一下优化的目标就是要找到一个合适的 $\Delta\mathbf{x}$ ，使得 ΔF_k 是一个极小值，所以我们使 ΔF_k 对 $\Delta\mathbf{x}$ 进行求导，并令导数为 0，可得

$$\frac{dF_k}{d\Delta\mathbf{x}} = 2\mathbf{b} + 2\mathbf{H}_k \Delta\mathbf{x} = 0 \quad (2.28)$$

$$\Rightarrow \mathbf{H}_k \Delta\mathbf{x} = -\mathbf{b}_k \quad (2.29)$$

综合所有边，可得：

$$\mathbf{H} \Delta\mathbf{x} = -\mathbf{b} \quad (2.30)$$

其中 \mathbf{H} 是整个系统的海塞矩阵，然后使用非线性优化方法高斯牛顿法或者 LM 算法对这个方程迭代求解，直到收敛得到最优解。

2.4 本章小结

本章详细地介绍了激光 SLAM 系统的运行原理，主要从前端里程计、回环检测、后端非线性优化三个方面着手，使用数学语言描述了激光 SLAM 的工作原理。同时本章分析了快速检测回环用到的基础数据结构 kd-tree 以及图优化相理论相关的非线性优化基础。

第 3 章 基于语义信息的回环检测算法研究

第 2 章由激光 SLAM 系统概述引出了回环检测的基本概念，并阐述了回环检测技术相关的理论基础。本章将在第 2 章的基础上说明 GOSMatch 算法的实现细节，GOSMatch 的工作流程如图 3-1 所示。GOSMatch 包含四个模块：语义检测，图描述子生成顶点匹配和几何一致性检测。首先，我们从原始点云中分割出语义信息。然后，建立基于直方图的点云描述子来进行快速最近邻搜索。接下来，我们将查询点云形成的无向完全图中的每个图顶点的顶点描述子与检索到的候选对象中对应的图顶点描述子进行比较，最终通过几何验证步骤中得出 6D 初始姿态估计和一个用于衡量查询点云帧和候选帧之间的损失值 loss。依据这个 loss 值便可以判断出当前点云是否与历史点云帧构成回环。

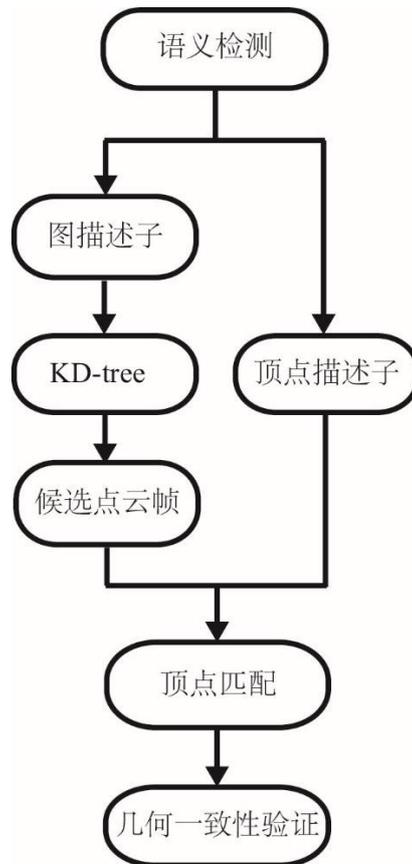


图 3-1 GOSMatch 算法流程图

3.1 基于 NDT 的前端里程计

虽然本回环检测算法并不需要前端里程计提供的全局位姿信息，但是语义检测模块需要利用相邻点云数据帧的局部里程计信息剔除掉造成干扰的语义信息。这里首先介绍基于正态分布变换算法的里程计实现原理。

正态分布变换算法（NDT）是一种基于概率学方法进行点云配准的扫描匹配方法，它最早由 Biber 等人[50]提出，并成功应用于 2D SLAM，后由 Magnusson[51]推广到三维场景使用。经过大量实验，Magnusson 证明力 NDT 方法在计算效率以及准确性上优于传统的 ICP 配准方法。并且 NDT 还有一个优势在于它可以牺牲计算结果的精度从而大大减少运算时间，加速配准效率。

其实，使用点云来描述一个场景中的曲面是存在许多缺点的，将曲面简化为点云的形式表示丢失了如曲面平滑度、孔、方向等去面特征。而 NDT 则是一种利用概率密度函数，将点云信息映射到光滑平面来表示。具体来讲，一个点云会被分割成很多部分，每个部分提取出不同的局部概率密度函数表示，不同的概率密度函数携带着不同曲面形状的描述信息。使用 NDT 算法进行两点云扫描匹配的工作流程如图 3-2 所示：

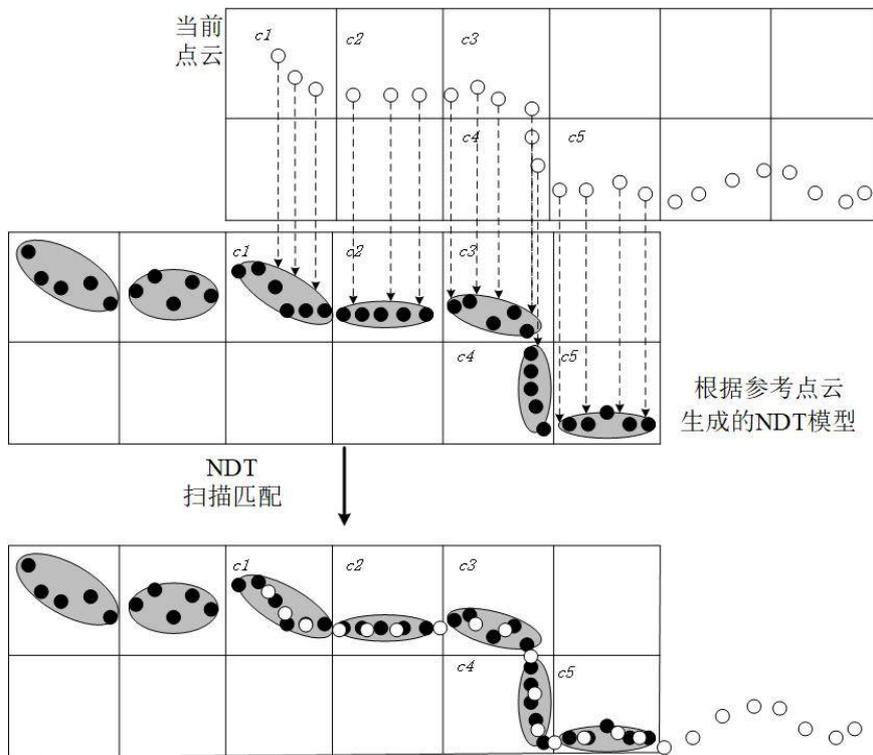


图 3-2 NDT 点云匹配算法示意图

NDT 算法的第一步就是要建立描述一帧点云的概率模型。首先将点云所在的空间划分为体素单元。在三维情况下体素单元通常是正方体，正方体的边长决定了一个完整空间可以划分出正方体的个数，所以正方体的边长决定了 NDT 算法划分空间的分辨率，这个边长是一个人工设置的参数，通常依算法运行的实际环境设定，且该参数对算法性能有较大影响，若参数数值过大，则 NDT 对点云的描述能力不强，配准准确性大大降低；若数值过小，则分辨率过大，使得算法不容易收敛，且增大了计算消耗，导致计算效率降低。划分好体素单元后，根据每个体素单元中激光雷达获得的点的分布计算出描述该体素单元的概率模型。假定每个体素单元中的点都是符合三维正态分布的，那么体素单元中任一点 x 的概率密度为：

$$f(x) = \frac{1}{(2\pi)^{3/2}\sqrt{|\Sigma|}} \exp\left(-\frac{(x-\mu)^T \Sigma^{-1}(x-\mu)}{2}\right) \quad (3.1)$$

其中 μ 和 Σ 表示该体素单元中高斯分布的均值向量和协方差矩阵。系数 $\frac{1}{(2\pi)^{3/2}\sqrt{|\Sigma|}}$ 是用于归一化的，保证该概率密度函数的积分值等于 1。 μ 和 Σ 的值可由公式 (3.2) 和公式 (3.3) 得出：

$$\mu = \frac{1}{m} \sum_{k=1}^m y_k \quad (3.2)$$

$$\Sigma = \frac{1}{m-1} \sum_{k=1}^m (y_k - \mu)(y_k - \mu)^T \quad (3.3)$$

其中 m 表示该体素单元中所有点的个数。 y_k 为第 k 个数据点。

使用正态分布来近似点云曲面有很多优势，高斯概率密度函数是连续可导的，方便计算。而且使用这种方式近似点云曲面，不但能够清晰地描述该局部点云在空间中的位置，还携带着该曲面的平滑度和方向等几何信息。具体地讲，曲面平滑度和方向可以由协方差矩阵的特征值和特征向量估计出来。如果三个特征值的大小差异不大，则当前的高斯分布函数描述的曲面形状是球面；若有一个特征值远远大于另外两个特征值，则这个曲面形状接近一条直线；若有一个特征值远远小于另外两个特征值，则这个曲面形状是一个平面。需要注意的是，如果一个体素单元中的所有点都是处在同一平面或共线的，那么计算出来的协方差矩阵是奇异的。而且在三维的情况下，若只通过最多 3 个点计算得到的结果总是奇异的。所以在计算局部概率密度函数时只挑选含有 5 个以上点的体素单元计算。

NDT 算法的目标是找到当前点云与参考点云的位姿变换关系，最大化当前点云帧中的点出现在参考点云表面的概率。假设当前点云 $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)$ ，两

点云间的位姿变换关系用 \mathbf{p} 表示, 设 $T(\mathbf{p}, \mathbf{x}_k)$ 为利用位姿变换关系 \mathbf{p} 转换当前点云中的点 \mathbf{x}_k 到参考点云坐标系中的空间转换函数。结合每个体素单元的局部概率密度函数, 我们可以得到一个似然函数:

$$L = \prod_{k=1}^n f(T(\mathbf{p}, \mathbf{x}_k)) \quad (3.4)$$

最大化该似然函数则可得到最佳变换关系 \mathbf{p}^* , 在求解过程中, 我们可以利用取对数的方法等效的将该似然函数简化为:

$$-\log L = - \sum_{k=1}^n \log (f(T(\mathbf{p}, \mathbf{x}_k))) \quad (3.5)$$

算法进行到这一步, 由于目前还没有可以直接求解出闭式解的方法, 这里就可以使用 LM 算法进行迭代求解, 最终求得当前点云与参考点云的最佳位姿变换 \mathbf{p}^* 。

3.2 语义检测

首先, 我们要确定在城市道路环境中什么类型的语义物体是有价值的。依据 [26] 中所论述的, 这样的语义物体必须是稳定的、有辨识性的和不依靠视点变化而变化的。稳定代表着物体是不会经常移动的, 如果该物体可以移动, 那么在第二次经过同一场景时, 该物体可能已经不存在了, 如果我们依靠这样的物体来判断是否重复到达过一个场景, 很可能得到不可靠的结果; 有辨识性的表明该物体不容易因为在做目标检测时检测出不稳定的结果, 使得该物体有时可以识别出有时却不可以或者误判为其他类型的物体, 所以最好该物体的结构性质是容易识别的; 不依靠视点变化而变化这一性质也非常重要, 在有回环存在时, 很可能我们并不是在同一个视角下观测到同一个物体的, 比如, 如果下一次路过同一个广告牌时激光雷达正好只能观测到它的侧边, 而第一次的观测是它的正面, 那么就会造成一次是线一次是面的观察结果, 会为两次识别结果带来严重的误差。综合考虑到这些性质, 我们在广泛的语义类别中主要检测停着的车, 树干和杆子。需要特别说明的是, 停着的车是一种有潜在移动能力的物体, 也就是说停着的车可以开走或者有新来的车会停下, 但是实际上在检测回环时它们的时间间隔都是比较小的。所以在此我们假设在检测回环的前后两帧中, 停着的车的位置不会发生剧烈变化, 即假设场景中不会有太多的车在一小段时间内会开走或驶入。

为了在 3D 激光雷达数据中做语义分割工作, 我们使用了一个叫做 RangeNet++ [27] 的最前沿的深度学习方法。RangeNet++ 可以实现快速且准确的点

云语义分割，是目前分割效果最好的深度学习分割算法[27]。这个端到端网络可以对原始点云中的每个点进行分割。它主要分为四个步骤：

第一步：将一帧输入点云数据通过球面投影的方式转换成一幅 2D 的 range image 表示。

第二步：通过一个 2D CNN 全卷积网络进行 range image 上的语义分割，网络架构如图所示：

第三步：利用分割好的 2D range image 数据恢复出原始点云所有点的语义标签，此时会有因为 CNN 蒙版的模糊问题和离散化导致物体呈现不该有的阴影。

第四步：利用基于 GPU 的 KNN 算法进行快速的 3D 后处理，利用投票策略修正错误的离散化和点云标签预测结果。

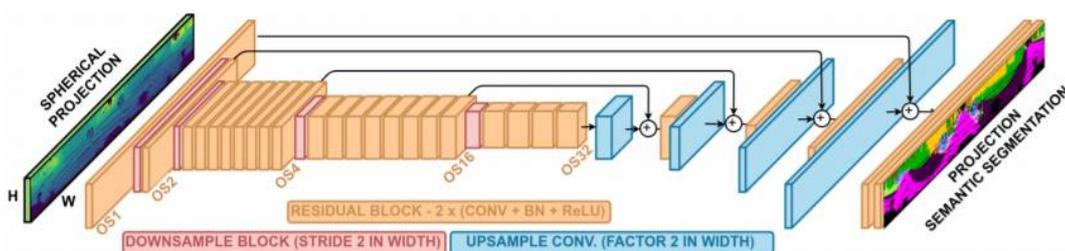


图 3-3 RangNet++神经网络架构[27]

一旦获得查询点云中所有车，树干，杆子的语义标签，我们就执行欧几里得聚类算法来检索点云中的语义物体。

欧几里德聚类算法的步骤如下所示：

1. 为输入点云数据 P 创建 kd-tree 表示；
2. 建立一个用于存放点簇的空列表 C ，以及一个需要检查的点的队列 Q ；
3. 然后，对 P 中的每个点 p_i ，执行以下步骤：

将 p_i 添加到当前队列 Q ；

对于 Q 中的每一个点 p_i ：

在点云数据 P 中作半径为 r ，球心为 p_i 的球体，搜索 p_i 的邻居点集合 P_i^k ；对于 p_i 的每一个邻居点，检查该点是否已被处理，如果没有，将其添加到 Q ；

当 Q 中所有点都被处理后，将 Q 添加到点簇 C 中，并将 Q 重置为空表。

4. 当 P 中的所有点已被处理并且现在是点簇 C 的一部分时，算法终止。

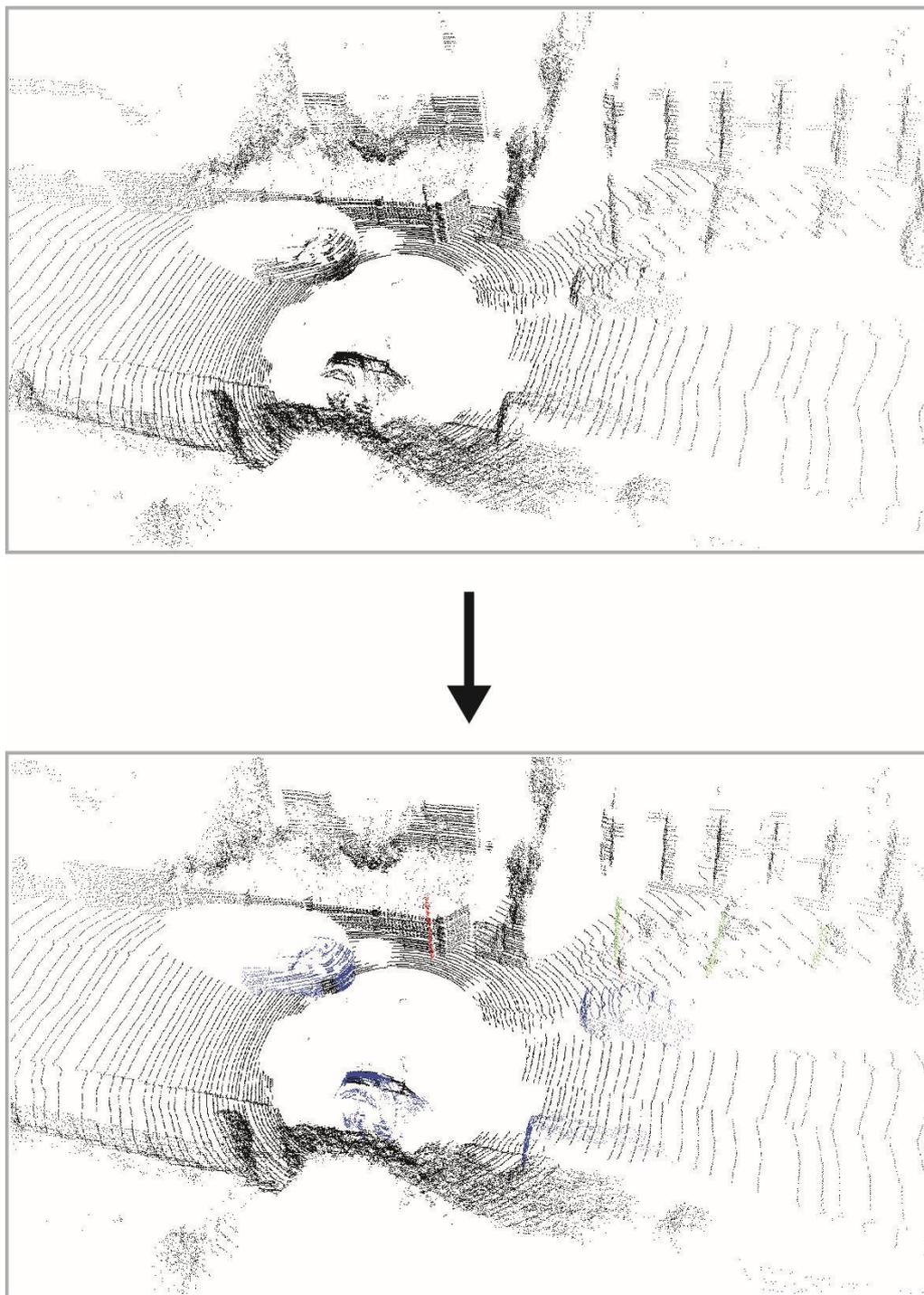


图 3-4 使用 RangeNet++和欧几里得聚类算法语义分割后的结果

图 3-4 展示了使用 RangeNet++和欧几里得聚类算法进行语义分割后的效果。其中车子使用蓝色标记，杆子使用红色标记，树干使用绿色标记。可以看到图中的四辆车均被完好的分割出来，既没有多余的将部分连结的地面点分成车子类型，也没有忽略本身属于车身的一些三维点；即使树干和杆子在这个场景中从粗细程度上来讲非常相似，但是语义分割算法还是具有足够高的辨识能力将两种类型的

物体进行分类。由此可见，两种算法结合起来可以从场景中精确地分割出语义物体。对于所有语义物体，本文计算它们的质心以表示它们在点云中的空间位置。质心的计算是将所有点云中的点的 x 、 y 、 z 坐标值相加，然后分别除以点云中点的个数得到最终的质心坐标，质心很有可能是点云中实际上本不存在的点。

需要注意的是，停着的车与树干和杆子不同，而 RangeNet++ 仅基于一帧点云的信息是无法确定检测到的车辆是停泊的还是正在移动着的，因此需要结合前端里程计来估计每个车辆的速度以简单地区分移动的车辆和停着的车辆。具体的，我们可以根据前端里程计计算出激光雷达获取的每一帧点云数据的位姿，在这里我们假设前端里程计在局部地图上是足够精确的，即考虑漂移误差趋近于 0。之后利用 RangeNet++ 将每一帧点云数据中的车辆识别出来，并依据前端里程计提供的点云位姿进行坐标变换，可以求解出当前帧中的车辆在全局地图坐标中的位置。两帧间的绝对时间间隔可以依据两点云数据的时间戳之差计算得到。至此，两帧中每辆车移动的距离和时间已经得到，可以大致估计出它们的移动速度，如该速度小于人工设定的阈值 ε ，那么我们就认为该车是静止不动的，保留该车的数数据，否则当作正在移动的车辆并剔除。需要注意的是，我们在下文中提到的车辆均为暂时是静止不动的。

3.3 图描述子生成

单个激光雷达扫描帧 P 由无向完全图 $G = \langle V, E \rangle$ 来描述即使用无向完全图 G 来表征一个点云数据，其中 V, E 分别代表顶点集合和边集合。我们将从语义检测模块获得的语义对象的位置视为图 G 中的顶点， E 集合中的边 $e_{ij} = \langle v_i, v_j \rangle$ 根据顶点 v_i 和 v_j 之间的欧几里得距离计算得到。 V 集合包含了图 G 中所有的顶点。这些顶点可以按它们的语义类别进行分类，则 V 中有三种具有不同语义类别的顶点（车辆，树干和杆子），与 V 相对应的， E 集合中的边就可以分为六种（车辆-车辆，树干-树干，杆子-杆子，车辆-树干，树干-杆子和杆子-车辆）。

描述整个点云的基于直方图的全局描述子一共由六个部分组成， E 中的每种类型的边分别构成六个部分之一。最后再通过合并的方式将六个部分描述子统一起来，得到完整的全局描述子。由于每个部分的计算步骤是类似的，这里举例

介绍使用图 3-5 中 $E^{pole-trunk}$ 中的边来计算六个部分之一的过程，其中 $E^{pole-trunk}$ 是 E 的子集，仅包含图中所有杆子-树干类型的边。需要注意的是，为了清晰的展示形成图描述子的过程，我们将本为 3D 的无向完全图简化为 2D 表示并且省略了图中的某些边。

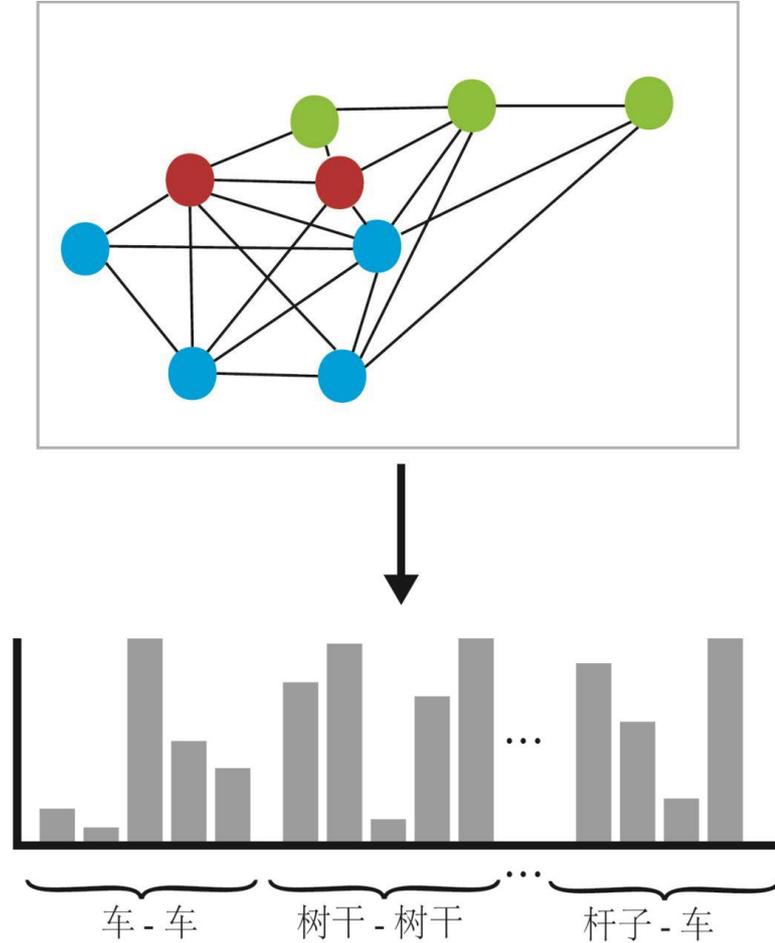


图 3-5 由语义物体构造图描述子

我们假设一个区间 I 从最短的边长度 l_{min} 到可能的最长的边长度 l_{max} :

$$I = [l_{min}, l_{max}] \tag{3.6}$$

我们将 I 分为互斥和分开的子区间值，每个子区间的大小可以通过以下公式计算，在这里我们假设桶的个数为 b 个：

$$\Delta I = \frac{1}{b}(l_{max} - l_{min}) \tag{3.7}$$

因此，每个桶对应于分离的子区间之一：

$$I_k = [l_{min} + k \cdot \Delta I, l_{min} + (k + 1) \cdot \Delta I] \tag{3.8}$$

然后，由 $E^{pole-trunk}$ 形成的全局描述子的这一部分可以表示为：

$$\mathbf{h}^{pole-trunk} = (h_0, h_1, \dots, h_{b-1}) \tag{3.9}$$

$$\text{其中} \quad h_k = \{e \in \mathbf{E}^{pole-trunk} : l(e) \in I_k\} \quad (3.10)$$

一旦处理了其他五种边,就可以通过连接六个部分来建立点云 P 的完整全局描述子 \mathbf{H}^{graph} :

$$\mathbf{H}^{graph} = (\mathbf{h}^{vehicle-vehicle}, \dots, \mathbf{h}^{pole-vehicle}) \quad (3.11)$$

这样设计得到的图描述子很好地发挥了激光雷达传感器的优势。从第2章可以了解到,激光雷达的测距原理决定了其测量激光雷达本身的位置与障碍物之间的距离是相对准确的。而本文提出的图描述子正是利用了各个物体间的欧几里得距离约束来对整个场景进行描述,根据三角测量原理,若可以在三维场景中得到3个物体的确切坐标,那么其他物体也可以依据三角测量原理确定具体坐标,而且物体间测距的越准确,得到的坐标信息就越精确。而在图描述子中,几乎所有的语义物体坐标都是可以确定的,所以利用各个物体间的距离代表整个场景在理论上是具有唯一性的。在另一方面,图描述子不仅仅只使用了“距离”这一信息,还是用了更高层的语义信息。若在一个三维场景中将所有物体的质心抽象成点,利用质心与质心间的距离便可以唯一标识一个场景,那么将每个质心赋予语义会带来更强的描述性,增加了描述子描述场景的鲁棒性。

每个历史点云数据的图描述子都会被存储在一个数据库中,它们用于构建kd-tree 以在每次查询点云到来时执行标准 K 最近邻搜索算法,有效搜索最相似的 N 个循环候选对象。每当查询完成后,查询点云的图描述子会被插入到kd-tree 中以保存数据,方便下一个查询点云进行快速查询。

3.4 顶点匹配

在上一小节中,我们已经得到了与当前查询点云的欧几里得距离最近的 N 个点云。这 N 个点云仅仅是通过粗略的对比全局描述子筛选得到的,在这个过程中显然忽略了点云中的一些细节比较。顶点匹配是一个比较点云之间细节相似度的操作,从而从这 N 个候选点云中筛选出一个与查询数据最相似的点云。

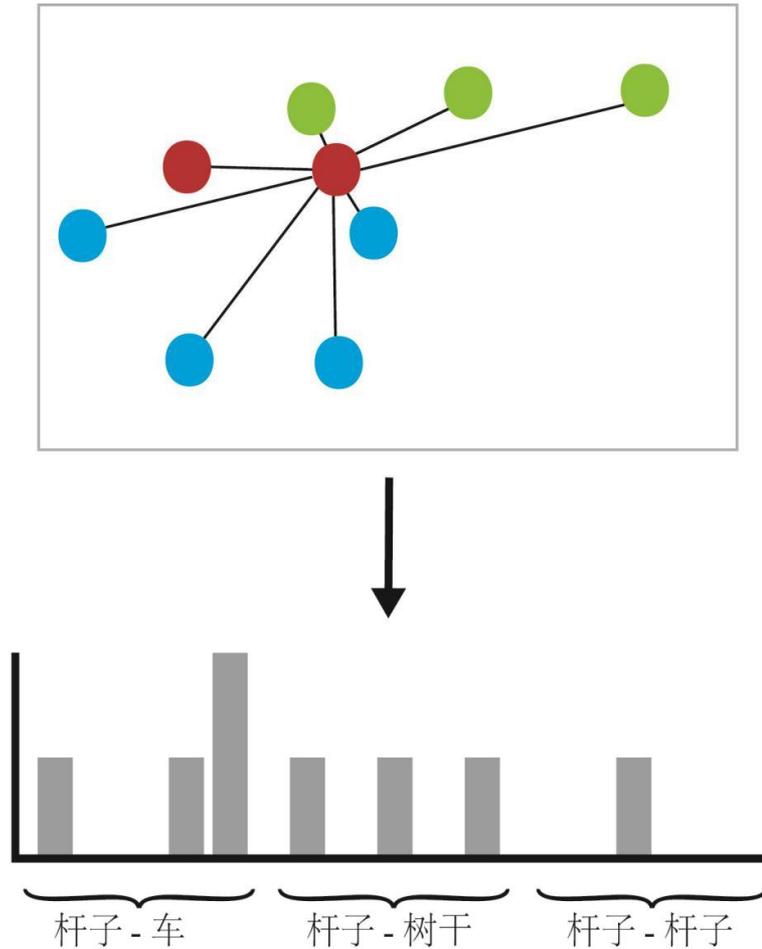


图 3-6 由顶点 v 相连的其他顶点构造顶点 v 的顶点描述子

在本节中，我们介绍顶点描述子以描述图中的一个顶点。顶点描述子是一种局部描述子，它的构造方法类似于上一节提及的全局描述子，顶点描述子也是基于直方图表示的。它们两者之间的区别在于，构建顶点描述子需要考虑的边不再是整个图包含的所有的边，而是仅与描述的顶点 v 相连接的边。例如图 3-6 所示，要为顶点 $v \in V_{pole}$ 构建顶点描述子，只需要考虑三种边，因为构成一条边的两个端点之一已经被确定为顶点 v 。之后的过程与图描述子类似，一种边可以形成顶点描述子的三个部分之一，可以描述为：

$$\mathbf{h}^{pole-trunk} = (h_0, h_1, \dots, h_{b-1}) \quad (3.12)$$

其中
$$h_k = |\{e \in \mathbf{E}^{pole-trunk} : l(e) \in I_k, e \in \mathbf{e}_v\}| \quad (3.13)$$

其中 \mathbf{e}_v 表示连接到 v 的边。将这三个部分串联起来后，可以最终获得 v 的顶点描述子 \mathbf{H}^{vertex} ：

$$\mathbf{H}^{vertex} = (\mathbf{h}^{pole-vehicle}, \mathbf{h}^{pole-trunk}, \mathbf{h}^{pole-pole}) \quad (3.14)$$

然后通过欧氏距离匹配一查询点云 Q 中的顶点描述子和候选点云 S 中

的顶点描述子，以找到两点云之间图顶点的对应关系 C ，具体过程为：

1. 初始化对应关系集合 C 为空。
2. 对于查询点云 Q 中每一个顶点描述子 q_i ，遍历 S 中所有顶点描述子，将欧几里得距离最近的两描述子 q_i, s_j 记录到对应关系集合 C 中，并在 S 中去除掉 s_j ，防止出现两个 Q 中的顶点描述子匹配到同一个候选点云中的顶点描述子的情况。

3.5 几何一致性验证

该步骤集中于为每个循环候选者选择一组几何一致的对应关系。虽然两帧之间所有顶点描述子的对应关系已经找到，但是它们中未必全都满足空间上的约束关系。这一步的目的就是从这些对应关系中挑选出一个子集，使得这个子集中的对应关系满足几何一致性。一个典型的解决方案是使用 RANSAC 算法来完善对应关系集合[29]：

第一步：在对应关系集合中随机选择不少于 4 个对应关系。

第二步：利用 SVD[30]方法求解选取数据的旋转矩阵 R 和平移向量 t 。

第三步：将其它未被选择的对应关系按求得的 R, t 进行旋转平移变换，若得到的误差满足设定阈值 ϵ 则记为内点，并放到该 R, t 模型的内点集合中，反之则记为外点。

第四步：重复迭代第 1-3 步，直到迭代次数达到预设值，跳出迭代过程，并保留内点数量最多的内点集合。

第五步：利用保留的内点数量最多的集合生成新的旋转矩阵和平移向量模型 R^*, t^* 作为算法的最终结果。

R^*, t^* 的分别为 3 维矩阵和 3 维向量，为了方便计算，可以将这两个合并在一个变换矩阵 T 表示。

$$T = \begin{bmatrix} R^* & t^* \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.15)$$

对于每个候选者，我们使用以下公式评估闭环检测损失：

$$loss = \sqrt{\frac{\sum_{i=1}^{|C|} (T \cdot \begin{bmatrix} c_{iq} \\ 1 \end{bmatrix} - \begin{bmatrix} c_{ic} \\ 1 \end{bmatrix})^2}{|C|}} \quad (3.16)$$

其中 C 代表查询扫描与其循环候选之间的精细对应集， T 代表转换矩阵， c_{iq} 和 c_{ic} 代表 C 中分别属于查询扫描和候选扫描的第 i 个 3D 点。最后选择具有最小损失 $loss_{min}$ 的候选扫描，以根据阈值 β 判断是否存在环路闭合：

$$L_{\beta}(P_q, P_c) = \begin{cases} true & \text{if } loss_{min} < \beta \\ false & \text{otherwise} \end{cases} \quad (3.17)$$

要注意的是，仅当失 $loss_{min}$ 小于 β 时，这两个位置才被视为循环闭合，反之亦然。确认回路闭合后，我们将转换矩阵视为 6 自由度初始位姿估计。

3.6 本章小结

本章在第 2 章的内容的基础上，介绍了 GOSMatch 回环检测算法的具体实现，包含了语义检测、图描述子生成、顶点匹配和几何一致性检测四个模块。详细地阐述了 GOSMatch 利用语义信息形成点云描述子由粗到精查找回环的工作过程。

第4章 实验结果与分析

GOSMatch 利用了深度学习方法对点云数据进行语义分割得到场景中的语义信息，然后利用语义信息分别形成两种描述子，借助 kd-tree 和 RANSAC 算法由粗到精快速判断回环。本章将 GOSMatch 和其他的传统激光 SLAM 回环检测方法对比，验证 GOSMatch 检测回环性能。在此之后通过噪声敏感度测试、初始位姿估计准确度和算法时间复杂度评估实验全面衡量了 GOSMatch 回环检测方法的优越性。

4.1 实验数据集简介

KITTI 数据集[30]中共有 11 个具有数据采集车轨迹真值的序列。我们选择序列 00、05、07、08 来评估所提出的方法，因为它们都是具有回环并且都是在城市环境中采集的，这意味着它们都具有较丰富的语义信息，具备测试本文提出的算法的基本条件。在这四个序列中，08 是可用于测试算法旋转不变性性能的唯一序列，因为序列 08 中的所有真实回环都是数据采集车沿相反的方向行驶得到的。

RangeNet++通过使用来自 SemanticKITTI 数据集[32]的标记点云数据进行训练，该数据集为 KITTI 数据集中的每个序列提供密集的逐点标记，依靠这些人工作标注信息，我们可以训练出对点云中每个三维空间点进行语义分类的神经网络。具体来说，对于我们选择的每个序列，我们将使用其他剩余的 10 个序列来训练 RangeNet++的一个模型，因此我们对选择出的 4 个序列（00、05、07、08）分别训练出了 4 个不同的模型。

4.2 对比实验方法概述

4.2.1 VFH 方法[16]

作为全局描述子的 VFH (Viewpoint Feature Histogram) 是基于局部描述子 FPFH (Fast Point Feature Histogram) 得到的描述子。而 FPFH 又是 PFH

(Point Feature Histogram) 的改进方法。在介绍 VFH 方法之前, 首先介绍一下 PFH 与 FPFH 局部描述子。

4.2.1.1 PFH 局部描述子

PFH[54] 是一个局部描述子, 它用来描述点云中挑选出来的关键点。PFH 试图通过分析附近法线方向之间的差异来捕获该点周围的几何信息 (因此, 不精确的法线估计可能会产生低质量的描述子)。

首先, 该算法将附近的所有点配对 (不仅将选定的关键点与其邻居配对, 而且还将邻居与其自身配对)。然后, 对于每一对, 根据其法线计算固定坐标系。依据这个坐标系, 就可以使用 3 个角度变量对法线之间的差异进行编码。保存这些变量以及点之间的欧氏距离, 然后在计算所有对时将它们用直方图表示。最终的描述子是每个变量的直方图合并在一起 (总共 4 个)。

4.2.1.2 FPFH 局部描述子

PFH 虽然可以得出准确的结果, 但它有一个缺点: 它的时间复杂度太高, 难以实时处理点云。对于 k 个邻居的 n 个关键点, 其计算复杂度为 $O(nk^2)$ 。FPFH[10] (快捷点特征直方图) 因此应运而生, 改进的目的就是加快求解速度。

FPFH 仅考虑当前关键点与其邻居之间的直接连接, 从而删除邻居之间的其他连接。这将复杂度降低到 $O(nk)$ 。因此, 将所得的直方图称为 SPFH (Simplified Point Feature Histogram)。参照系和角度变量的计算则与 PFH 如出一辙。为了解决这些额外连接的丢失, 在计算完所有直方图之后, FPFH 需要执行一个附加步骤: 将点的邻居的 SPFH 与自己的 SPFH 合并, 并根据距离对其进行加权。这样做的效果是可以给出远至所用半径 2 倍的点的点表面信息。最后, 将 3 个直方图 (不使用距离) 连接起来组成最终的 FPFH 描述子。

4.2.1.3 VFH 全局描述子

VFH 基于 FPFH, 但与 FPFH 不一样的是, VFH 是一个全局描述子。由于后者对物体位姿有不变性, 因此 VFH 通过包含有关视点的信息来扩展它。而且, FPFH 对于整个点云簇估计一次, 而不是每个点估计一次。

VFH 由两部分组成: 视点方向分量和扩展的 FPFH 分量。为了计算第一个分量, 算法将找到点云对象的质心, 该质心是对所有点的 X , Y 和 Z 坐标求平均值的结果。然后, 计算视点 (传感器的位置) 和此质心之间的向量并将其标准化。

最终,对于点云中的所有点,都将计算此向量与其法线之间的角度,并将结果合并到直方图中。在计算角度时,矢量会转换到每个点,因为它会使描述子比例不变。

第二个分量的计算方式与FPFH相同(对于3个角度特征 α , φ 和 θ 产生3个直方图),但有一些差异:仅针对质心,使用计算出的视点方向向量作为其法线(如该点显然没有法线),并将所有点云簇中的点设置为相邻点。

4.2.2 Z-projection 方法[15]

Z-projection 与 VFH 描述子相似的一点是它也需要计算局部表面法向量。Z-projection 提供了两种计算局部表面法向量的方法。一种是对点云中的任意给定点,从整个点云中选取其最近的20个邻居,通过这20个邻居进行平面拟合以找到所挑选出的点的局部表面法线。另一种则是利用了Velodyne激光雷达装置激光束的几何关系近似局部表面法线的方法。对于一个给定的点,它的左右邻居是在该点之前和之后立即由同一激光束获取的点。上下邻居是由紧接在获取所关注点的激光束上方和下方的激光束获取的最近点。一旦选择了四个相邻点,就可以通过获取给定点与四个相邻点形成的位移矢量之间的向量积,然后对得到的向量积进行几何平均来估算局部表面法线。

我们假设机器人在足够平坦的地形上运动,因此传感器坐标系的Z轴被认为是垂直的。一旦有了局部表面法线,我们就将这些法线向量的点积与Z轴方向单位向量相乘。实际上,采用这种点积是量化机器人周围环境中结构的一种方法。这些点积的值介于-1和1之间。与X-Y平面平行的法线向量导致点积具有较低或接近零的值,而与Z轴平行的法线向量导致点积具有接近-1和1的值。

4.2.3 M2DP 方法[13]

在回环检测任务中,描述子在3维空间中的旋转平移不变性是很重要的,为了保证M2DP拥有这个性质,作者使用输入点云的质心当作形成M2DP描述子时所参考的原点。在得到了参考原点后,使用PCA方法对点云进行预处理。这一步是要假设每一个点云有两个参考轴,利用这两个参考轴,点云就可以进行粗略配准。PCA最终的结果便可以得到形成描述子时参考坐标系的X轴和Y轴,利用这两个坐标轴配准便可以得到具有旋转不变性的点云。

在经过预处理后,接下来要从处理过后的点云中提取2D单一视角标签。每

一个单一视角可以看作一个 2D 平面 X ，单一视角标签就是通过将点云中的 3D 点投影到 X 平面，然后提取 2D 标签得到的。在这里， X 平面一定是一个过原点的平面。 X 的法向量记作 m 。一个法向量 m 可以由一组方位角 θ 和倾斜角 φ 唯一确定，所以，一组方位角和倾斜角 $[\theta, \varphi]$ 可以确定一个唯一的平面 X 。然后，将整个点云包括质心和坐标轴投影到由 $[\theta, \varphi]$ 确定的 X 平面上。为了得到投影到 X 上点的结构信息，算法会将 X 平面分成一个个块。分割算法如下：首先围绕点云的质心生成 l 个以 $[r, 2^2r, \dots, l^2r]$ 为半径的同心圆环，其中最大半径则为质心和离其最远点的距离。然后将这每个圆环分成 t 个块，这样就把一个平面分成了总共 $l \times t$ 个块。计算出每一个块中所包含点云中点的个数，这样就可以得到一个描述 X 平面上 3D 点投影的大小为 $l \times t$ 的向量 v_x 。然后，M2DP 将点云投影到 n 个 2D 平面上并提取出 $n \times l \times t$ 个向量用于描述投影过程。将这些向量用一个矩阵 A 表示出来，紧接着对矩阵 A 进行 SVD 分解，将左右奇异向量合并在一起得到最终的点云 M2DP 描述子。

相比其他基于法向量的描述子，M2DP 不需要计算每个点的法向量。这可以极大的减少形成描述子需要的计算资源的消耗。

4.2.4 Scan Context 方法[14]

Scan Context 首先像 M2DP 那样将点云分割成 N_r 个同心环，若激光雷达能获取到的最远的点距离为 L ，则每个同心环的宽度为 $\frac{L}{N}$ ，不同于 M2DP 的是，Scan Context 紧接着又将整个点云以激光雷达为中心均匀地放射出互相之间夹角为 $\frac{2\pi}{N}$ 的射线将点云分为 N_s 个扇区，扇区与同心环相交错，形成了一个不相交的块。此时需要注意的是，扇区是按照相同的方位角分割为成的，那么距离点云中心最远的块就会比相对近一些的块覆盖的面积大，然而每个块都会为构成 Scan Context 描述子贡献相同权重的信息量，所以理论上应该挑选出一种方法使得其可以补偿远处面积大，近处面积小会带来的描述偏差。因此，Scan Context 从每个不同的块提取出一个特殊点用于描述这个块的信息。因为激光雷达在越远的地方形成的点就越稀疏，所以这种方法是一种很好的补偿方法。

从每一个块中挑选出一个特殊的点用于描述一个块内的点云信息是有很多种可取的方法的，Scan Context 选择每个块中在 Z 轴上最高的点作为选取的方法。使用这种方法的好处是可以有效地汇总周围结构的垂直形状，而且无需进行

大量计算来分析点云的特征。另外，最大高度也表示了周围环境中的哪个部分对激光雷达传感器是可见的。最终 Scan Context 可以使用一个 $N_r \times N_s$ 大小的二维矩阵来表示，其中矩阵中的每个值都是对应点云块中最高点的高度。

不同于之前介绍过的三种描述子，Scan Context 是一个二维矩阵，所以用来衡量这个描述子之间的距离指标也不不同的，而其他三种都是一维向量描述子，都是可以直接使用欧几里得距离来测量两描述子之间的距离大小的。对于从查询点云中提取出的 Scan Context 描述子 I^q 和从候选点云中提取出的 I^c ，要比较他们之间的距离，需要使用按列对比的方式，每一列正好代表着点云中一个扇区的的信息，一列列计算出距离后，将距离加和就作为 I^q 和 I^c 描述子之间的距离。其中，计算两描述子每一列的距离使用余弦相似度计算，并且最终两描述子之间的距离使用 N_s 做归一化表示。然而，这种计算方式只适用于找到视角不变形成的回环，而对于那些视点发生变化的回环则是不合适的。为了解决这个问题，所以在每次一列一列地比较完得到结果后，将 I^c 描述子的列向后平移一列，最后一列移到第一列，然后再和 I^q 像之前的对比方式那样进行比较，当 I^c 描述子经过平移后回到最初始的矩阵后停止比较，并挑选出最小的距离值当作两描述子之间的距离，这样进行比较就满足了旋转不变性。

虽然上述计算 Scan Context 描述子距离之间的方法很鲁棒，但是这种方法很消耗计算资源，难以在实际应用中达到实时检测回环的目标。所以此算法还提出了另一个描述子 ring key 用于加速查询回环。ring key 是一个一维向量，向量的大小与 N_r 一致，ring key 中的每一个值包含了点云中每一个同心环中的块所平均包含的三维点数。这样 ring key 也是一个满足旋转不变性的向量。ring key 可以用于构造 kd-tree 来快速查询历史点云中有没有和查询点云具有相似 ring key 的点云。并筛选出部分相似的候选点云帧，提取出 Scan Context 描述子后，再用 Scan Context 精确匹配是否存在回环。

4.3 实验参数设置

在 KITTI 数据集中，有一种情况是汽车两次通过相同的十字路口，一次在十字路口的左上角，另一次在十字路口的右下角。显然这样的情况也属于回环。但是需要注意的是，这种情况下形成的回环往往距离较远。因此，在本文的回环检

测性能评估实验中，如果重复经过的两个地方的欧几里得距离小于 15m，则认为存在是真正的回环。对于每个查询点云数据，我们排除了在它之前紧相邻的 100 个点云数据，以防止查询点云与其在时间上相隔过近的邻居数据相匹配。Z-Projection 和 VFH 都是基于表面法线的方法，它们都需要法线计算的步骤。为了方便代码的实现，我们采用[15]中提到的第二种方式对 Z-Projection 计算法线，并将 Z-Projection 中的桶的数量设置为 202。VFH 算法中法线半径是唯一需要设置的参数，本实验中将其设置为 0.03m。依据 Kim 的文章，Scan Context 的候选帧数目设置为 50 时它具有相当不错的性能。对于 Scan Context 的其余参数，本次实验使用开源代码中建议的默认值。与 Scan Context 相似，对于 M2DP，我们使用开源代码中提到的参数。对于我们的方法，我们为所有验证实验均设置相同的参数： $v_{min} = 0$ ， $v_{max} = 60$ ， $b = 60$ ， $N = 10$ 。

在本节中，在 KITTI 里程表数据集上评估了基于语义信息的回环检测算法 GOSMatch[30]。本文与其他最新的闭环检测算法（包括四个全局描述子）进行了比较：VFH，M2DP，Scan Context 和 Z-projection。对于 VFH 算法，本文使用了点云库（PCL）[31]中 VFH 的 C++实现；对于 M2DP 和 Scan Context 算法，本文使用了他们对应的论文中提及的开源 Matlab 代码。此外，本文提出的 GOSMatch 方法和 Z-projection 方法是在 Matlab 平台上编码实现的。所有实验均在具有 16 GB RAM 的 Intel Core i7-6820HQ 和具有 12 GB RAM 的 Nvidia Titan X 的桌面计算机上进行的。

4.4 回环检测性能评估

在第一个实验中，我们评估 GOSMatch 的回环检测性能。回环检测的结果可以分为四类，如表 4-1 所示：

表 4-1 回环检测结果分类

检测\真实	回环	非回环
回环	真阳性 (True Positive, TP)	假阳性 (False Positive, FP)
非回环	假阴性 (False Negative, FN)	真阴性 (True Negative, TN)

对于回环检测这个任务，我们希望 TP 与 TN 相比 FP 与 FN 尽可能高，这说明算法可以高效准确地判断出是否存在真实的回环。为了更好的评估算法的回环检测性能，我们通常使用精确率和召回率两个指标。精确率表示算法检测到的回环就是真实回环的概率，召回率表示检测出来的回环中有多少是真实的回环。

精确率计算公式(4.1)如下所示：

$$Precision = \frac{TP}{TP+FP} \tag{4.1}$$

召回率计算公式(4.2)如下所示：

$$Recall = \frac{TP}{TP+FN} \tag{4.2}$$

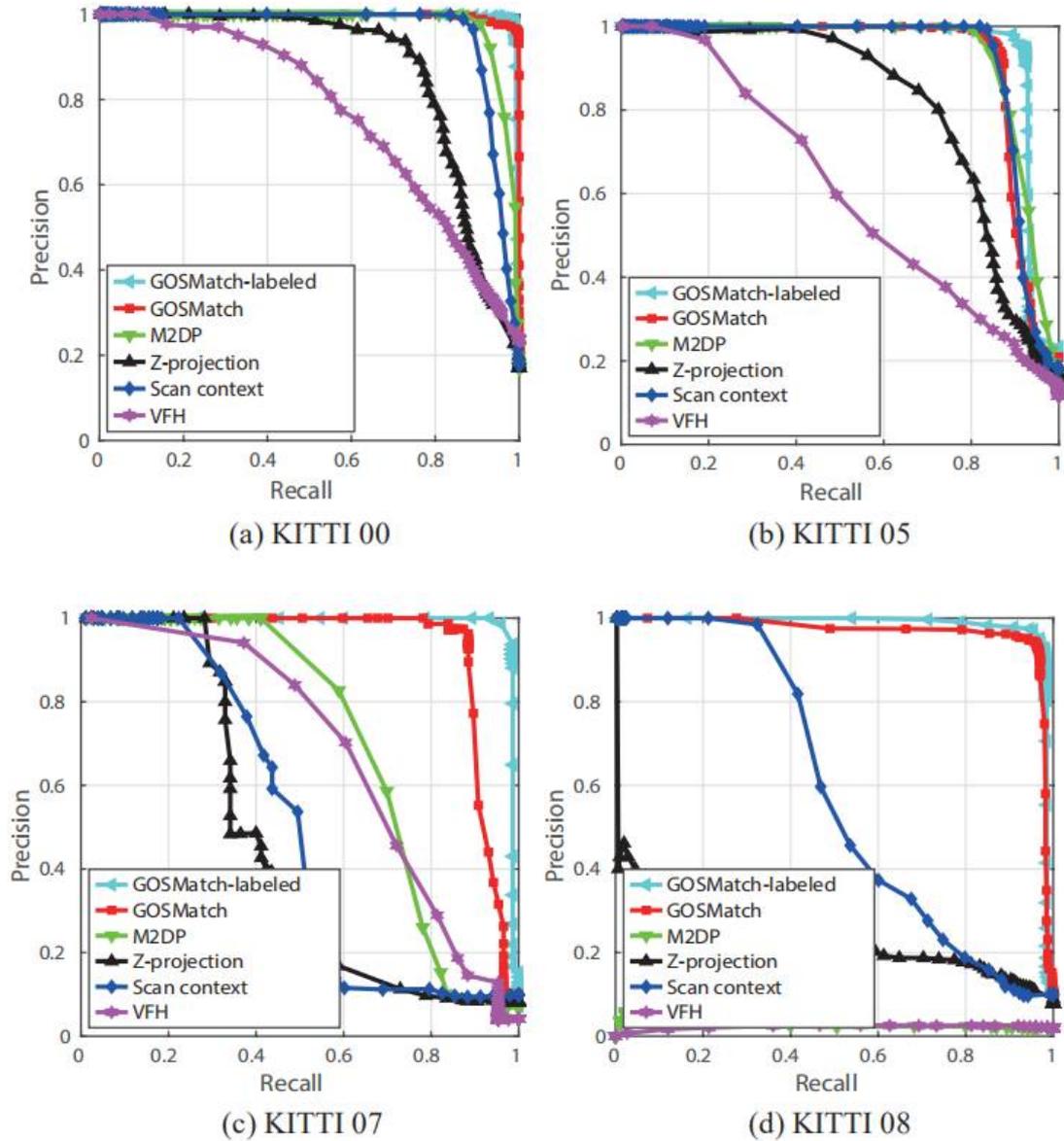


图 4-1 在 KITTI 数据集上各回环检测算法的精确率-召回率曲线

图 4-1 显示了精确率-召回率曲线。请注意，GOSMatch-labeled 使用带标签的点云数据作为输入，而其他五种方法使用原始点云数据作为输入。对于那些非基于语义信息的方法，尤其是 Z-Projection 和 VFH 方法，它们表现出了较差的性能。这是因为这些描述子在很大程度上取决于估算出的表面法线的准确性，而仅靠一些周围点的几何性质推算出法线往往带有很多偏差。此外，这些类型的描述子仅关心法线与点云的质心或与 Z 轴方向矢量所形成的角度，而忽略几何空间信息。在这种情况下，如果法线向量在两个不同位置之间的概率分布相似，则这些方法很容易得出辨识度非常低的点云描述子甚至是带有极大偏差的点云描述子。在我们的实验中，Scan Context 和 M2DP 取得了相对较好的结果。但是，他们无法应对具有挑战性的 KITTI 08 数据集。在 KITTI 08 中，所有的回环都是当汽车在相反方向上再次访问同一位置时生成的。这样的情况只有具有旋转不变性的回环检测算法才能高效的检测出回环。显然 M2DP 无法成功检测到 KITTI 08 中的回环，并且 Scan Context 的精确率-召回率曲线在此数据集的表现也急剧下降。

对于 GOSMatch 算法，总体而言，由于 RangeNet++ 的点分类错误的影响，GOSMatch 的性能比 GOSMatch-labeled 的性能稍差。即使这样，与其他方法相比，它们都表现出显著的鲁棒性和具有竞争性的旋转不变性。本文提出的方法受益于由有价值的语义信息形成的较稳定的点云描述子。即使视点发生很大变化，此类描述子也不会受到很大影响，因此可以执行可靠的位置识别。

4.5 噪声敏感度评估

本节重点评估 GOSMatch 的抗噪声能力。在此实验中，本文选择直接从 SemanticKITTI 提供的标记点云数据作为语义检测模块的输入数据以避免分类噪声的干扰。本文利用伯努利分布生成随机噪声，通过检验精确率的方式以评估 GOSMatch 算法对抗噪声的能力。具体的实验设计为：

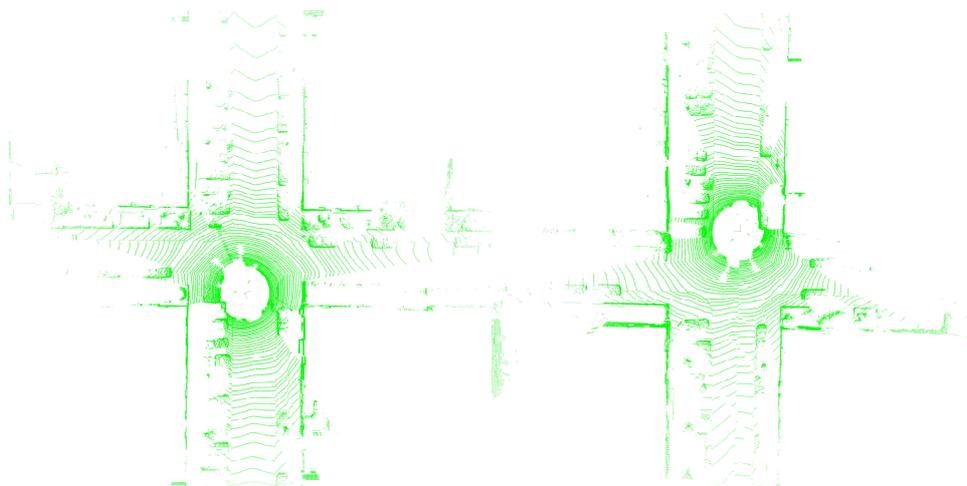
在每次扫描中随机移除 X 个对象，以模拟车辆位置的变化（例如，先前停放的车辆已经驶离）以及模拟一些错误的分类结果， X 是随机变量，且 $X \sim B(n, \frac{1}{2})$ ，其中 n 表示每次可以从点云数据中删除的最大语义物体数。本文在四个序列上分别测试了不同 n 值带来的影响，并对每个序列中的每个 n 值执行 100 次运行，

对得到的结果取均值。表 4-2 列出了在召回率为 100%时，不同 n 的值下的平均精确率结果。

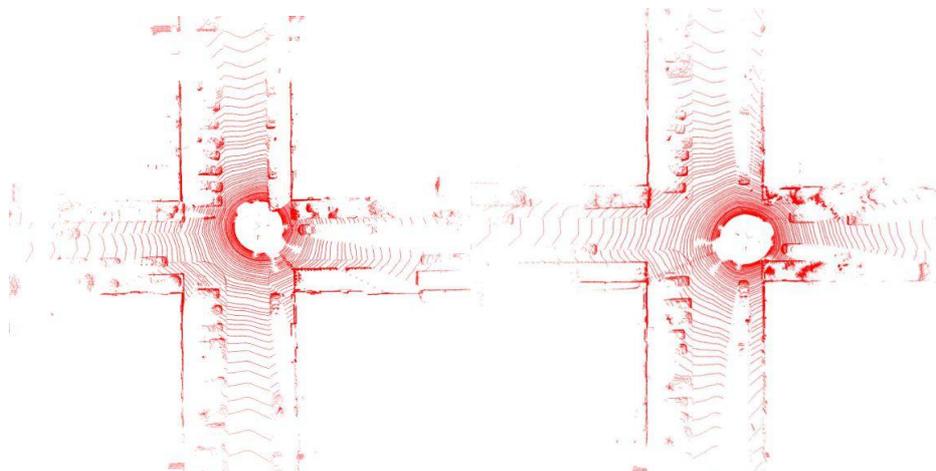
表 4-2 召回率为 100%时，不同 n 值下 GOSMatch 算法的平均精确率

	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$
KITTI 00	95.36%	94.49%	92.87%	92.61%	89.13%
KITTI 05	74.51%	71.14%	67.74%	55.58%	50.61%
KITTI 07	92.01%	89.21%	87.84%	88.61%	86.14%
KITTI 08	82.44%	77.91%	76.49%	68.64%	66.90%

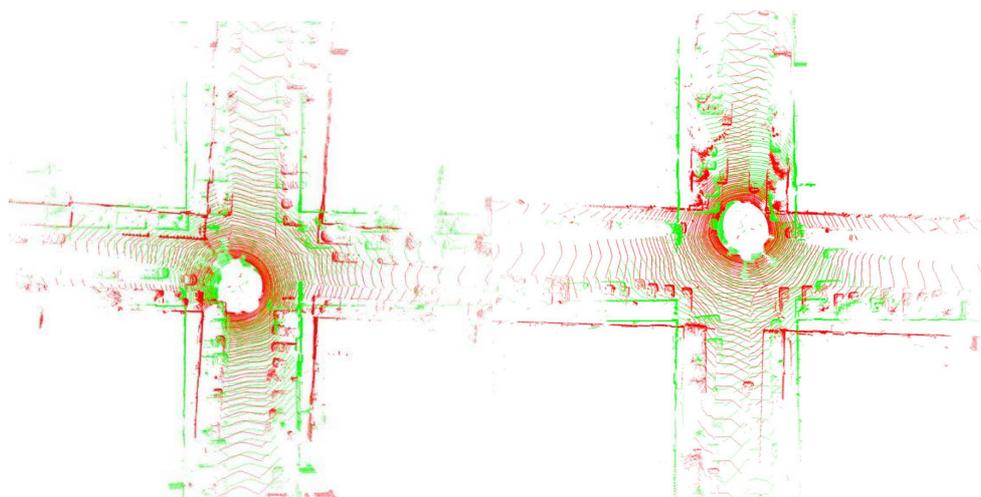
4.6 初始位姿估计准确度



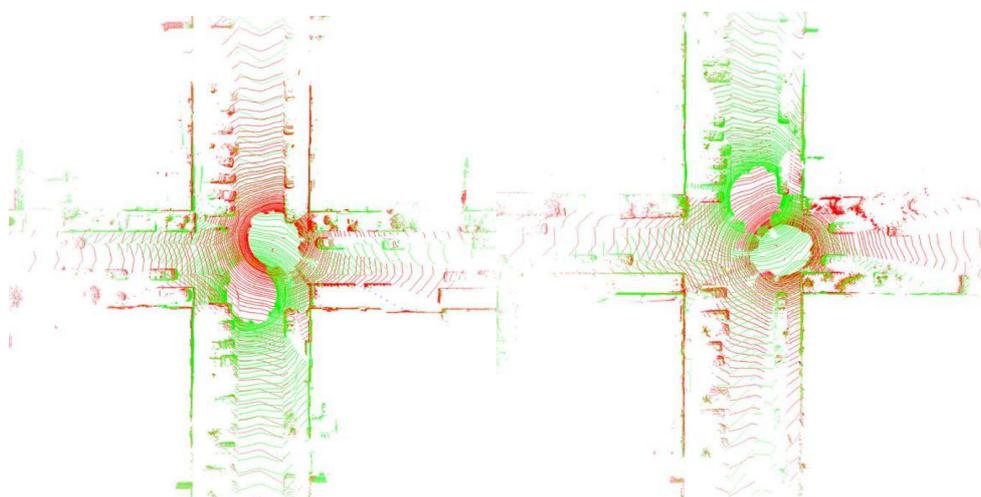
(a) 查询点云帧



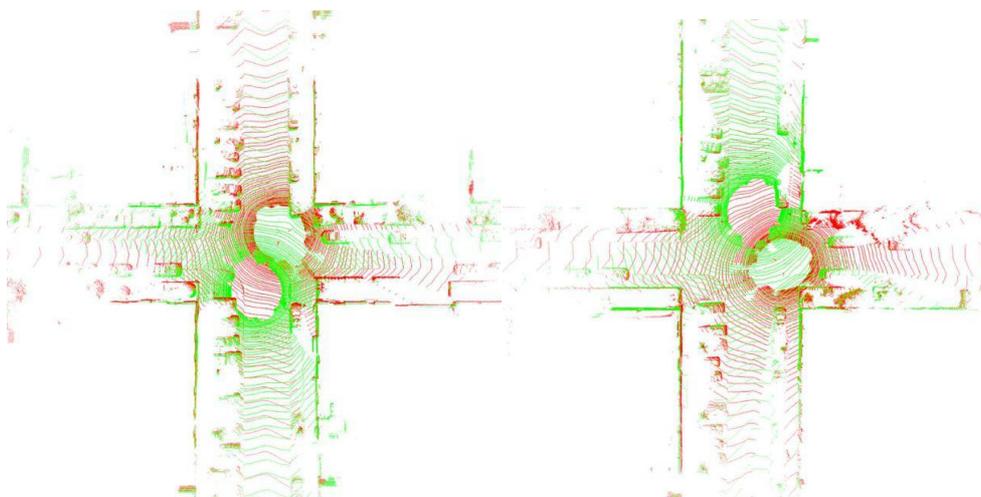
(b) 检测到的回环帧



(c) 没有初始位姿估计直接使用 ICP 配准两点云的结果



(d) 仅使用 GOSMatch 提供的初始位姿估计配准两点云的结果



(e) 带有初始位姿估计使用 ICP 配准两点云的结果

图 4-2 两个 KITTI 00 序列上检测到回环时生成的初始位姿估计示例

本部分的目的是评估 GOSMatch 提供的初始位姿估计的准确性。在实际应用中,初始位姿估计是很重要的。虽然可以使用更强大的配准算法,例如 ICP 和 NDT 算法来获得更可靠的位姿结果,但是这些配准算法都是基于非线性优化进行迭代优化得到结果的。也就是说,一个较准确的初值能够使这些算法解算出更加准确的位姿变换结果,防止陷入不那么优秀的局部最值中。

图 4-2 描绘了来自 KITTI 00 序列上关于初始位姿估计准确性的两个示例。其中图 4-2 (a) 中是两个查询点云帧,左图为第 1415 帧,右图为第 4540 帧。图 4-2 (b) 中是根据查询点云帧从历史点云数据中搜索到的对应回环帧,左图为第 586 帧,右图为第 116 帧。图 4-2 (c) 是不使用初始位姿估计直接配准查询点云帧和匹配到的回环帧的结果。图 4-2 (d) 是仅使用初始位姿估计将检测到的回环帧进行坐标变换与查询点云帧匹配的结果。图 4-2 (e) 是使用带有初始位姿估计的 ICP 算法配准两点云后的结果。

在图 4-2 (e) 中,结果显示可以使用带有初始位姿估计的 ICP 算法成功配准查询点云帧和检测到的点云帧。但是,在图 4-2 (c) 中,如果没有提供的初始位姿估计,则 ICP 配准过程很容易失败,使用 ICP 进行配准的结果甚至不如初始位姿估计值准确。此外,如图 4-2 (d) 所示,使用初始位姿估计的对齐结果接近于 ICP 算法优化的结果,显示了估计的初始位姿估计的高精度。实验结果表明,GOSMatch 在检测回环时能够给出精确的初始位姿估计。

在本次实验中,本文还测试并验证了初始位姿估计对 ICP 算法性能的影响。带有和不带有初始位姿估计的 ICP 的计算时间消耗和点对点 RMSE 分别在图 4-3 和图 4-4 中给出。我们可以看到,配准的过程从 GOSMatch 提供的初始位姿估计中受益匪浅。GOSMatch 提供的初始位姿估计不但缩短了配准过程的时间还降低了配准结果的误差。

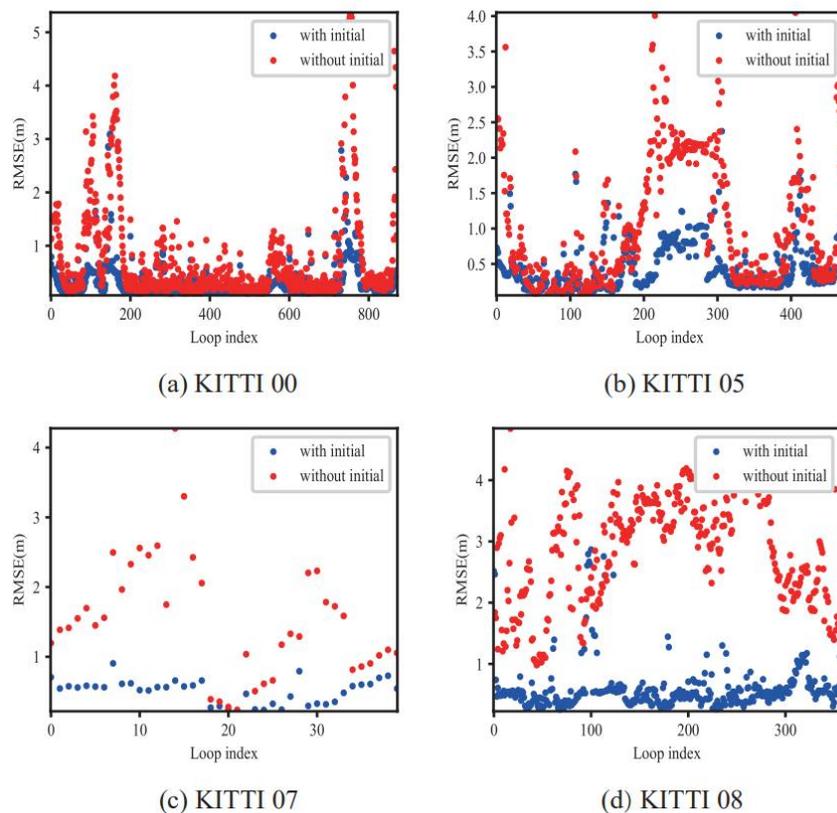


图 4-3 KITTI 各序列上有无初始位姿估计对 ICP 点到点的 RMSE 的影响

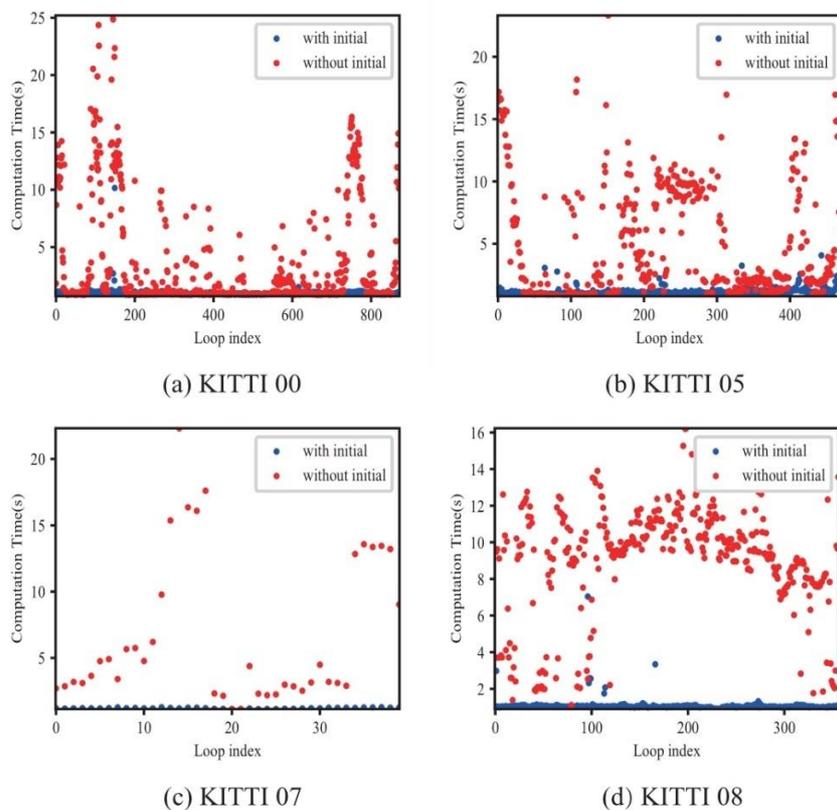


图 4-4 KITTI 各序列上有无初始位姿估计对 ICP 算法收敛时间的影响

4.7 时间复杂度分析

这个实验在 KITTI 00 上评估了不同回环检测方法的需要的计算时间，具体结果如表 4-3 所示。因为本文提出的方法与传统方法的不同之处在于 GOSMatch 需要语义检测步骤，所以本文将计算两个图类型描述子的时间和语义检测的时间分开显示在表 4-3 中。此外，需要注意的是 GOSMatch 中的语义检测模块是仅在 GPU 上执行的，其他所有方法的实验以及 GOSMatch 的其它模块部分均在 CPU 上执行。GOSMatch 检测回环的搜索时间包括 k 近邻搜索算法的时间消耗，匹配顶点描述子的时间消耗以及验证几何一致性的时间消耗。从表 4-3 的结果可以看出，GOSMatch 在进行回环检测时需要花费的时间在很大程度上取决于语义检测模块，若有更快的语义分割算法可以显着提高 GOSMatch 的执行效率。

表 4-3 KITTI 00 数据集上各回环检测算法消耗的平均时间对比(单位: 秒)

	描述子计算	检测回环	总用时
Scan Context	0.0424	0.0514	0.0938
M2DP	0.2461	0.0220	0.2681
Z-Projection	0.4045	0.0201	0.4246
VFH	0.5625	0.0379	0.6004
GOSMatch	0.0183 +0.3192*	0.0278	0.3653

*代表进行语义检测的时间

4.8 本章小结

本章的主要内容是对 GOSMatch 算法进行测评以证明其检测回环的性能，选择了 VFH、Z-projection、M2DP 和 Scan Context 作为对比方法，这些方法都是基于三维点几何特征的传统回环检测方法。测评中选择了回环检测的精确率和召回率作为算法的衡量标准，使用了 KITTI 大型公开数据集对各个算法进行了实验对比。从实验可以看出 GOSMatch 在召回率接近 100% 时，准确率相比其他算法依然只高不低，其回环检测性能超过了所有对比方法。同时在接下来的一系列实验可以看出，GOSMatch 具有很好的抗噪声鲁棒性、可以得到准确的回环初始位姿估计和较快的运行效率。

第 5 章 总结与展望

回环检测在 SLAM 工程中有着重要的地位。一个快速、鲁棒的回环检测算法能够带来更精确的定位结果，并且可以维护建出的地图具有一致性。在点云数据中，场景中的物体是用一个个密集离散的三维空间点来描述的，在这个描述的过程中本身丢失了一些高层面的结构信息，使用深度学习方法对这些离散的点簇赋予语义而后加以利用，便可以为帧点云数据带来更高的辨识度，而具有高辨识度的点云特征描述为实现高效的回环检测奠定了基础。

5.1 工作总结

本文首先分析了激光 SLAM 和回环检测的原理，以及必要的传感器硬件激光雷达的工作原理。选取 kd-tree 这一关键性的数据结构加速了查找回环的效率，紧接着设计实现了一种利用激光 SLAM 中的语义信息进行回环检测的方法 GOSMatch，它基于两种由语义信息构建的图描述子快速鲁棒地检测场景中是否存在回环。一种图描述子用于从历史点云中高效搜索相似的循环候选对象，另一种用于进一步细致的匹配，以解决两个位置之间相同语义物体的对应关系，两种类型的描述子可以由粗到精快速确定是否存在回环。

随后，本文的实验部分复现了 VFH、Z-projection、M2DP 和 Scan Context 等基于点结构信息的传统回环检测算法，并在自动驾驶大型公开数据集 KITTI 上与本文提出的算法进行对比。与这些回环检测方法不同，本文提出的方法在语义级别上执行，此方法不仅对环境变化具有鲁棒性，而且能够给出准确的 6D 初始姿势估计。详尽的评估实验结果显示了使用语义对象之间相对位置关系的潜力，这对于可靠的闭环检测非常有帮助。

最后，本文又通过抗噪声实验、初始位姿估计准确度评估和时间复杂度分析实验全方面多角度的对 GOSMatch 进行了评估验证，证明了本文提出的算法不仅可以鲁棒地应对环境噪声、提供查询点云帧与回环帧之间准确的初始位姿变换估计，在运行时间上也同样占据优势地位，体现了 GOSMatch 算法的先进性。

5.2 研究展望

本方法在以下方面还有进一步提升的空间：

5.2.1 多源传感器融合

在激光里程计部分，由于单方面依据激光雷达点云数据扫描匹配得到的运动估计可能会带有比较大的误差。IMU，轮式里程计等传感器可以提供的不同的位姿估计信息，利用无迹卡尔曼滤波算法将多源传感器信息融合优化可以得到更精准的位姿推算结果，从而降低前端里程计带来的误差。一旦激光里程计拥有更好的运动估计能力，那么利用运动滤波方法排除掉点云数据中正在运动的车也变得更加精确，从而提升最终的回环检测效果。

在回环检测方面，由于图像上的语义分割要比直接在激光点云上做语义分割更加准确，所以可以利用相机传感器辅助语义分割，但是仅仅依据相机传感器很难获取到分割得到的物体的空间位置坐标，所以需要利用激光雷达传感器确定分割出的物体较准确的三维坐标。要实现这样的功能，必须将激光雷达和相机传感器进行标定。

5.2.2 回环检测的实时性有待提升

目前本文提出的方法还不能实时的检测回环，所谓的实时是指算法应该可以在激光雷达获取两帧点云数据的间隔时间（即约 0.1s）便可以检测出当前点云帧是否与历史点云帧存在回环。然而本文的算法在实际应用中只能达到 2-3 Hz 的工作频率。事实上，在整个算法流程中，语义检测模块是最占用的时间的。如果使用语义分割效率更快的算法，或是深入研究一套匹配本文方法的语义分割框架，那么本文提出的方法就可以大幅减少检测回环的时间，达到实时进行回环检测的要求。

参考文献

- [1] S. Thrun et al., “Robotic mapping: A survey,” *Exploring artificial intelligence in the new millennium*, vol. 1, no. 1-35, p. 1, 2002
- [2] N. Sunderhauf and P. Protzel, “Brief-gist-closing the loop by simple means,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1234 - 1241
- [3] M. Cummins and P. Newman, “Fab-map: Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647 - 665, 2008
- [4] M. J. Milford and G. F. Wyeth, “Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643 - 1649
- [5] L. Chen, L. Sun, T. Yang, L. Fan, K. Huang, and Z. Xuanyuan, “Rgb-t slam: A flexible slam framework by combining appearance and thermal information,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5682 - 5687
- [6] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1 - 19, 2016
- [7] M. Bosse and R. Zlot, “Place recognition using keypoint voting in large 3d lidar datasets,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2677 - 2684
- [8] B. Steder, G. Grisetti, and W. Burgard, “Robust place recognition for 3d range data based on point features,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1400 - 1405
- [9] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard, “Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1249 - 1255
- [10] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3212 - 3217
- [11] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, “Appearance-based loop detection from 3d laser data using the normal distributions transform,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 23 - 28
- [12] A. Buyval, A. Gabdullin, R. Mustafifin, and I. Shimchik, “Realtime vehicle and pedestrian tracking for didi udacity self-driving car challenge,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2064 - 2069
- [13] L. He, X. Wang, and H. Zhang, “M2dp: A novel 3d point cloud descriptor and its application in loop closure detection,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 231 - 237
- [14] G. Kim and A. Kim, “Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802 -

4809

- [15] N. Muhammad and S. Lacroix, "Loop closure detection using small-sized signatures from 3d lidar data," in 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. IEEE, 2011, pp. 333 - 338
- [16] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010, pp. 2155 - 2162
- [17] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in 2011 IEEE international conference on robotics and biomimetics. IEEE, 2011, pp. 2987 - 2992
- [18] M. Kuse and S. Shen, "Learning whole-image descriptors for real-time loop detection and kidnap recovery under large viewpoint difference," arXiv preprint arXiv:1904.06962, 2019
- [19] T. Rohling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 736 - 741
- [20] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5266 - 5272
- [21] T. Shan and B. Englot, "Lego-loam: Lightweight and groundoptimized lidar odometry and mapping on variable terrain," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4758 - 4765
- [22] T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 3. IEEE, 2000, pp. 2512 - 2517
- [23] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1687 - 1694, 2018
- [24] M. Fisher, M. Savva, and P. Hanrahan, "Characterizing structural relationships in scenes using graph kernels," in ACM SIGGRAPH 2011 papers, 2011, pp. 1 - 12
- [25] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1 - 8
- [26] F. Yu, J. Xiao, and T. Funkhouser, "Semantic alignment of lidar data at city scale," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1722 - 1731
- [27] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2019
- [28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381 - 395, 1981
- [29] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," Josa a, vol. 4, no. 4, pp. 629 - 642, 1987
- [30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving?"

- the kitti vision benchmark suite,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 3354–3361
- [31] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in 2011 IEEE international conference on robotics and automation. IEEE, 2011, pp. 1–4
- [32] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” in Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV), 2019
- [33] Johnson, Andrew. (1997). Spin-Images: A Representation for 3-D Surface Matching
- [34] Huber D F. Automatic three-dimensional modeling from reality[M]. Carnegie Mellon University, 2002
- [35] Smith, Hamish R C, Cheeseman, Liam P. On the Representation and Estimation of Spatial Uncertainty[M]. Sage Publications, Inc. 1986
- [36] Smith R, Self M, Cheeseman P, et al. Estimating uncertain spatial relationships in robotics[C]. international conference on robotics and automation, 1987: 850–850
- [37] Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using FastSLAM[C]. international conference on robotics and automation, 2003: 1985–1991
- [38] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem[C]. national conference on artificial intelligence, 2002: 593–598
- [39] Montemerlo M, Thrun S, Roller D, et al. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges[C]. international joint conference on artificial intelligence, 2003: 1151–1156
- [40] Grisetti G, Stachniss C, Burgard W, et al. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23(1): 34–46
- [41] Lu F, Milios E. Globally Consistent Range Scan Alignment for Environment Mapping[J]. Autonomous Robots, 1997, 4(4): 333–349
- [42] Konolige K, Grisetti G, Kummerle R, et al. Efficient Sparse Pose Adjustment for 2D mapping[C]. intelligent robots and systems, 2010: 22–29
- [43] Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]. international symposium on safety, security, and rescue robotics, 2011: 155–160
- [44] Hess W, Kohler D, Rapp H H, et al. Real-time loop closure in 2D LIDAR SLAM[C]. international conference on robotics and automation, 2016: 1271–1278
- [45] Segal A, Hhnel D, Thrun S. Generalized-ICP[C]// Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 – July 1, 2009. 2009
- [46] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]. robotics science and systems, 2014
- [47] Bentley, Louis J. Multidimensional binary search trees used for associative searching[J]. Communications of the Acm, 1975, 18(9):509–517
- [48] Liang M, Min H, Luo R. Graph-based SLAM: A survey[J]. Jiqiren/Robot, 2013, 35(4):500
- [49] Besl P J, Mckay N D. A Method for Registration of 3-D Shapes[J]. Proceedings of Spie the International Society for Optical Engineering, 1992, 14(3):239–

- 256
- [50]Biber P , Strasser W . The Normal Distributions Transform: A New Approach to Laser Scan Matching[C]// Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. IEEE, 2003
- [51]Magnusson M. The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection[D]. Örebro universitet, 2009
- [52]Barfoot T D. State estimation for robotics[M]. Cambridge University Press, 2017
- [53]Thrun S. Probabilistic robotics[J]. Communications of the ACM, 2002, 45(3): 52-57
- [54]Rusu R B, Marton Z, Blodow N, et al. Persistent Point Feature Histograms for 3D Point Clouds[C]. international conference on intelligent autonomous systems, 2008

附录 A 攻读硕士期间研究成果

1. 论文

[1] Yachen Zhu, Yanyang Ma, Long Chen, et al. GOSMatch: Graph-of-Semantics Matching for Detecting Loop Closures in 3D LiDAR data. IEEE\RSJ International Conference on Intelligent Robots and Systems (审稿中)

2. 专利:

[1] 朱亚琛, 陈龙, 刘聪, 一种基于图描述子的激光 SLAM 回环检测系统及方法(公开号: CN110910389A)

致 谢

光阴荏苒，两年的研究生学习时光弹指一挥间就要过去了。我由衷地感谢我的导师刘聪老师和陈龙老师。在研究生期间，无论在科研的道路上还是生活上两位导师都给予了我很大帮助和支持。他们严谨的治学态度和对学术研究锐意进取的精神在我在学术道路上受阻时极大地鼓舞了我，让我有动力坚持下去。

感谢实验室课题组的各位小伙伴们在我完成硕士论文期间给予我的帮助。

感谢开源代码的贡献者及开源组织，他们的无私奉献不仅让我学习了很多知识，而且对我的毕业论文中的对比实验部分提供了很大的帮助。

感谢中山大学培养了我，在中山大学学习的这六年，我开阔了视野，增长了很多专业知识，结识了很多可爱的同学。我非常感谢中山大学提供给了我完成硕士毕业论文需要的实验条件，让我能顺利地完成实验。

特别感谢我的父母及家人朋友们，他们在我完成硕士毕业论文期间在学业上鼓励我不断前进，在生活上关心我照顾我，非常感谢他们的支持。

朱亚琛

2020年6月

