电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目 多传感器融合的同时定位建图与多目标跟踪算法研究

专业学位类别	**交通运输**
学　　号	**202022100534**
作 者 姓 名	**李文宇**
指 导 教 师	**汪子君　副研究员**
学　　院	**航空航天学院**

分类号 TP242.6 密级 公开

UDC[注1] 629.3

学 位 论 文

多传感器融合的同时定位建图与多目标跟踪算法研究

（题名和副题名）

李文宇

（作者姓名）

指导教师 **汪子君 副研究员**

电子科技大学 成都

（姓名、职称、单位名称）

申请学位级别 **硕士** 专业学位类别 **交通运输**

提交论文日期 **2023 年 4 月 13 日** 论文答辩日期 **2023 年 5 月 29 日**

学位授予单位和日期 **电子科技大学 2023 年 6 月**

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Research on Multi Sensor Fusion Simultaneous Localization Mapping and Multi Object Tracking

A Master Thesis for Professional Degree Submitted to
University of Electronic Science and Technology of China

Discipline:	**Transportation**
Student ID:	**202022100534**
Author:	**Wenyu Li**
Supervisor:	**Associate Prof. Zijun Wang**
School:	**School of Aeronautics and Astronautics**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：李文宇　　日期：2023年6月27日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，同意学校有权保留并向国家有关部门或机构送交论文的复印件和数字文档，允许论文被查阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索及下载，可以采用影印、扫描等复制手段保存、汇编学位论文。

（涉密的学位论文须按照国家及学校相关规定管理，在解密后适用于本授权。）

作者签名：李文宇　　导师签名：

日期：2023年6月27日

摘 要

同时定位和地图构建技术 (SLAM) 是机器人领域的一个重要研究问题。其基本要求是机器人需要利用各种传感器数据在未知环境中估计机器人位姿和构建环境地图。然而，经典 SLAM 理论建立在静态环境的假设下。当机器人处在复杂动态的现实环境中时，场景中的运动物体会影响其定位精度和地图质量，出现轨迹偏移和地图鬼影等现象，这对现有的 SLAM 方案提出了挑战。

本研究旨在解决现实环境中动态物体对 SLAM 的影响问题，通过多种方法增强 SLAM 在动态环境下的定位能力，降低轨迹误差，并提升所建立地图的质量。具体而言，本文提出了一种视觉激光融合的同时定位建图与多目标跟踪 (SLAMMOT) 方案，以应对动态环境的挑战，解决了在动态环境下对机器人位姿、环境地图和动态目标位姿同时估计的复杂问题。本文的主要工作如下：

(1) 提出了一种基于深度补全的视觉激光融合 SLAM 算法。首先将激光雷达数据投影至相机平面，使用深度补全技术对稀疏深度进行补全。接着利用稠密深度和图像实现视觉里程计，利用激光点云实现激光里程计。最后，利用因子图对视觉激光里程计进行融合，输出机器人的位姿并构建环境地图。

(2) 提出了一种联合自运动估计和 3D 运动目标检测的迭代动态配准算法。首先基于 3D 目标检测获得目标的位置和大小，然后利用点云配准估计帧间运动，利用估计的位姿分割并估计出运动物体，将运动物体从点云中移除，只保留静止物体。最终，实现在动态环境下利用静态环境点云实现帧间稳定的点云配准。

(3) 提出一种紧耦合的同时定位建图和多目标跟踪算法。在输入点云中移除地面保持稳定估计，然后结合动态配准算法，设计激光里程计算法。结合联合概率数据关联滤波器在松耦合的情况下实现 SLAMMOT，结合全局最近邻关联实现因子图紧耦合的 SLAMMOT，最后输出环境地图、自身轨迹和动静态物体运动轨迹。

为了评估本文提出的算法方案的性能，将在公开数据集上通过比较所提出方法与传统 SLAM 方法在动态环境下的表现来验证其有效性，使用以下指标进行衡量：绝对轨迹误差 ATE，相对位姿误差 RPE，点云地图准确性。多个序列上的实验结果证明本文提出的方法可以在动态环境下实现准确的定位，生成准确的地图和机器人轨迹，在多种环境下定位精度和地图质量相比原有方案均有提升，从而实现更好的机器人自主感知和导航能力。

关键词：同时定位与地图构建，多目标跟踪，多传感器融合

ABSTRACT

Simultaneous Localization and Mapping (SLAM) is an important research problem in the field of robotics. Its basic requirement is for a robot to estimate its own pose and construct an environmental map using various sensor data in an unknown environment. However, classical SLAM theories are based on the assumption of a static environment. When a robot operates in a complex and dynamic real-world environment, the presence of moving objects in the scene can affect the accuracy of localization and the quality of the map, resulting in trajectory drift and ghosting phenomena, which pose challenges to existing SLAM solutions.

The purpose of this study is to address the impact of dynamic objects on SLAM in real-world environments. Multiple methods are proposed to enhance the localization capability of SLAM in dynamic environments, reduce trajectory errors, and improve the quality of the constructed map. Specifically, this thesis presents a solution called Multi Sensor Fusion Simultaneous Localization Mapping and Multi Object Tracking (SLAM-MOT), which aims to handle the challenges posed by dynamic environments. It tackles the complex problem of simultaneously estimating robot pose, environment map, and dynamic object poses in dynamic environments. The main contributions of this paper are as follows

(1) Proposed a visual-lidar fusion SLAM algorithm based on depth completion.. First, project the lidar onto the camera plane, use depth completion technology to complete sparse depth, use dense depth and image to implement visual odometry, use lidar point clouds to implement lidar odometry, and use factor graphs to fuse visual-lidar odometry to output self-pose and construct the environment map.

(2) Proposed an iterative dynamic registration algorithm that combines self-motion estimation and 3D motion object detection. Based on 3D object detection to obtain the position and size of the object, estimate frame-to-frame motion using point cloud registration, segment and estimate moving objects using estimated poses, remove moving objects from the point cloud, and retain stationary objects to achieve frame-stable point cloud registration in dynamic environments using static environment point clouds.

(3) Proposed a tightly coupled simultaneous localization and mapping (SLAM) algorithm with multi-object tracking. Remove the ground from the input point cloud to main-

tain stable estimation, then combine with the dynamic registration algorithm and design the lidar odometry algorithm. Use the joint probabilistic data association filter to achieve SLAMMOT in a loosely coupled manner, and use global nearest neighbor algorithm to achieve tightly coupled SLAMMOT with factor graphs. Finally, output the environment map, ego-trajectory, and moving and static object trajectories.

In order to evaluate the performance of the proposed approaches in this paper, their effectiveness will be validated by comparing their performance with traditional SLAM methods in dynamic environments using publicly available datasets. The following metrics will be used for evaluation: Absolute Trajectory Error (ATE), Relative Pose Error (RPE), and point cloud map Accuracy. Experimental results on multiple sequences demonstrate that the proposed method can achieve accurate localization, and generate precise maps and robot trajectories in dynamic environments. The proposed method outperforms the existing approaches regarding localization accuracy and map quality in various environments, enhancing the robot's autonomous perception and navigation capabilities.

Keywords: Simultaneous Localization and Mapping(SLAM), Multi Object Tracking(MOT), Multi Sensor Fusion

目 录

第一章 绪 论

1.1 研究工作的背景及意义

同时定位和地图构建 (Simultaneous Localization and Mapping,SLAM) 技术可以通过系统搭载的传感器获取周围环境，对环境信息进行处理实现自身定位与环境地图构建，在机器人、自动驾驶等领域有着重要作用。当前的 SLAM 系统大多依赖于静态环境假设，只包含刚性与静止的物体，并将动态的物体视为噪声。但在现实世界的复杂场景下，非刚性、动态的物体是不可避免其存在的，因此对周围场景进行感知和理解，具有十分重要的意义。如在自动驾驶的场景中，汽车不仅必须定位自身，还必须可靠地感知其他车辆和路人，以避免碰撞。在 AR/VR 场景下，需要实时感知动态物体，去实现虚拟对象和现实世界的物体的交互。真实世界中无处不在的动态场景对传统的 SLAM 提出来巨大挑战。

当场景为静态或动态物体较少时，由于 SLAM 系统将动态信息其视为离群值，可以使用一些鲁棒的定位方法以保持自身准确的位姿估计，如随机抽样一致性算法 (RANdom SAmple Consensus，RANSAC)、蒙特卡洛定位 (Monte Carlo Localization,MCL) 和鲁棒核函数等方法。但当处于复杂场景或高动态场景时，使用传统的 SLAM 方法会导致定位出现较大偏差甚至定位失败。

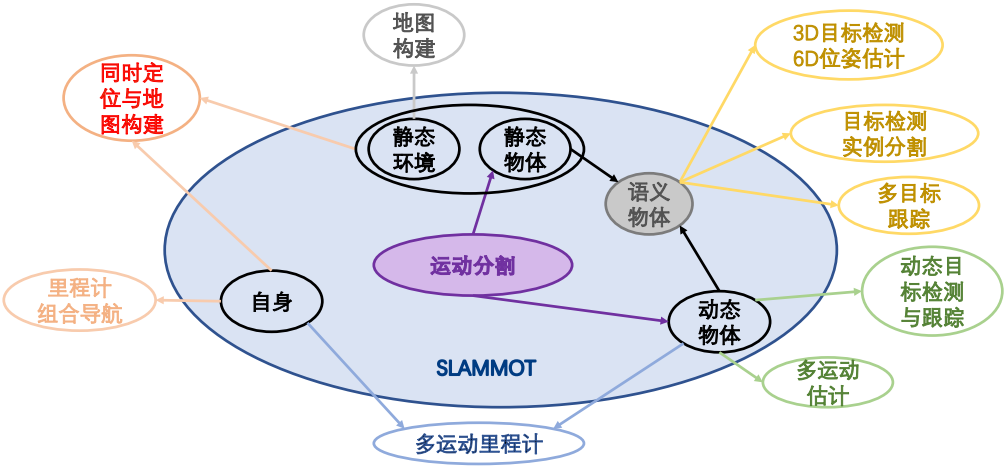


图 1-1 SLAMMOT 涉及领域

面对复杂动态的环境，有两类较多被应用以改进 SLAM 的思路：一种是检测包含动态物体的数据并移除[1,2]；一种是对动态物体进行检测与跟踪[3]。两者的思路都是通过对动态场景理解提高 SLAM 在复杂动态环境下的表现。区别在于前者滤除了动态区域，仅利用静态环境的信息进行定位，而后者利用动态信息对物

体进行检测跟踪，增加了 SLAM 系统的感知能力。后者被称为 SLAMMOT(SLAM and Moving Object Tracking)[4]。如图 1-1所示，SLAMMOT 需要在动态环境下同时估计自身运动、物体运动以及重建静态环境，因此会涉及有关自定位和环境感知诸多领域的技术。简单来说，以估计自运动和重建静态环境为核心的技术是导航定位和 SLAM 技术，以感知场景中物体为核心的技术是以深度学习为代表的目标检测跟踪方法。而与 SLAMMOT 较为接近的是多运动里程计和多运动估计问题[5]，主要是同时对场景中多个物体的运动进行拟合和估计，但 SLAMMOT 相比二者更鲁棒，对物体的运动的估计也更精确，因为 SLAMMOT 是同时考虑了自身、静态环境和动态物体三者耦合的估计。

去除动态物体的 SLAM 简化了计算，可以简单滤除动态信息并保证了定位精度，但环境感知变得不再完整，无法实现对动态物体的感知。往往需要借助其他算法再次对场景中的物体进行检测与跟踪，才能实现动态避障与自主运动。而依赖 SLAM 定位的场景往往是低算力嵌入式平台 (区别与有高精度组合导航方案的平台)，这就因重复计算造成了平台算力浪费，SLAMMOT 的提出正好解决了此类问题，并且可以通过二者的结果得到更精确更稳定的结果。

传统 SLAM 无法在动态场景下保持准确的自身位姿估计，SLAMMOT 的出现不但解决了动态场景下定位的问题，还能感知并跟踪场景中多个物体。SLAMMOT 的核心在于动态物体的检测和跟踪，这一问题在过去是困难的。使用基于深度学习的 3D 目标检测可以实现准确的目标感知[6]，在此基础上判断检测到的物体是否处于运动，即可将环境分割为包含静态物体的静态环境和动态物体两部分，再利用动态和静态的信息分别实现定位、建图和跟踪任务。

SLAM 与 MOT 在定义上都属于状态估计问题，这就为二者的结合提供了理论基础，如基于卡尔曼滤波、粒子滤波等方法都可以实现 SLAM 与 MOT。随着图优化理的发展，基于滤波的 SLAM 逐渐被基于图优化方法所取代。其中的因子图[7]是一种强大通用的工具，可以将多种环境信息建模为不同的因子，融入统一的框架。特别的是，因子图相比传统图优化适合更广义的状态估计问题，也逐渐出现利用因子图解决目标跟踪的问题的方法[8,9]。可以预见到，基于因子图完全可以实现 SLAM 与 MOT 的融合，以此增加 SLAM 在动态场景下的感知与定位能力。国际上近几年逐渐开始对基于因子图的 SLAMMOT 进行研究[10–12]，适合更复杂更现实的大规模场景，体现出更高的精度。

1.2 国内外研究历史和现状

1.2.1 SLAM 基本方案

对于利用外部传感器实现 SLAM 的具体方案，都可以视为直接法和特征法。对于特征法而言，从输入数据中提取特征并估计并同时特征点位置和自身位姿，如基于视觉 ORB 特征的 ORB SLAM2[13]，基于 3D 激光点特征的 LOAM[14]。直接法是利用传感器原始数据进行定位，不提取特征，这有利于在特征退化、非结构化场景下进行定位，如视觉 SLAM 中的 LSD-SLAM[15] 利用最小化光度误差求解相机位姿和地图点，激光 SLAM 中 Fast-lio2[16] 直接利用原始点云数据进行最小化距离的配准。

但以上各种 SLAM 方案，无论其使用什么特征或传感器，均建立在静态环境假设的前提下。当场景中存在少量动态物体时，可以通过鲁棒的估计方式使得 SLAM 的定位和建图结果保持稳定，但动态物体的影响并未完全消除，在定位过程中自身轨迹会发生漂移，建立的地图中也会存在物体运动的鬼影，这也限制了 SLAM 在现实环境中的部署和应用。

1.2.2 动态环境下的 SLAM

SLAM 可以从输入数据中提取特征，在静态环境中表现良好。然而，在具有动态物体的环境中，如汽车、行人、动物等的环境中，性能会显著下降。因为特征可能来自于这些物体，从而使定位不太可靠[17]。

传统的运动目标检测算法由于自身和物体同时发生运动，如帧差法、光流法等无法准确地分割出运动目标，即便通过外部传感器补偿自身运动，也因为分割方法限制无法获得物体完整的表示。基于深度学习的目标检测和语义分割算法可以使用先验语义知识从图像或点云中提取物体，后续通过检测、跟踪、移除动态物体以改进 SLAM 算法，优于传统的运动目标检测方法。

Yu[1] 在 DS-SLAM 中结合语义分割和运动一致检测，减少了动态物体对 SLAM 的影响。Bescos[2] 提出一种 DynaSLAM 方法，主要联合多视图几何和实例分割检测运动物体并移除，从而实现稳定的定位，并且可以重建并修复被动态物体遮挡的区域。Vincent[18] 在 DOTMask 中，在视觉 SLAM 的基础上使用实例分割算法提取物体轮廓，然后使用扩展卡尔曼滤波器跟踪提取到的目标并进行移除，可以以较快的速率实时运行。

对动态环境下 SLAM 的研究主要集中在视觉 SLAM 领域，这主要受益于基于深度学习的目标检测和分割方法的发展。在激光 SLAM 领域，近几年也逐渐出现相关研究。Chen[19] 提出 SuMa++ 算法，使用点云语义分割网络判断投影后的点云

的状态并移除动态点云，然后在点云配准中构建语义约束。Pfreundschuh[20] 提出一种无监督的运动目标检测网络，并利用移除运动目标后的点云进行定位。

目前动态环境下 SLAM 的解决思路主要是检测并移除动态物体，这一方法在大多数场景下均能取得较好的结果。但其局限在于，在高动态环境下移除动态物体有可能导致可利用的信息过少，从而间接影响 SLAM 的算法性能。此外，大部分动态 SLAM 使用基于深度学习的目标检测方法，这使得动态 SLAM 受限于检测算法的精度，一旦检测出现错检或漏检，物体就会被错误保留或剔除。

1.2.3 同时定位建图和多目标跟踪

动态环境下 SLAM 在处理运动目标时大多先估计自运动，再去检测运动目标。动态环境下要准确的自运动估计需要将运动目标从环境中分离，而判断分割运动目标又需要准确的自运动估计结果。二者类似 SLAM 诞生时“鸡生蛋，蛋生鸡”的问题，因此需要平衡二者之间的关系。可以看出，运动目标检测和同时定位建图之间存在着一种关系，连接两种算法的即为准确的自运动估计。

Wang[3] 在 2003 年首次提出同时定位建图和目标跟踪的问题和解决方法。作者认为 SLAM 和 DATMO(Detection And Tracking of Moving Objects) 是可以共同处理并相互改进的。后续大多工作都在其提出的理论基础上进行改进。如 Chung 提出的 SLAMMOT-SP[21]，Choi[22] 提出基于 Rao–Blackwellized 粒子滤波 (RBPF) 的方法，并使用交互多模型 (IMM) 算法实现目标跟踪。Wang 提出一种名为 4D SLAM 的算法[23]，基于 LeGO LOAM 构建，并使用 UKF-IMM-JPDA 滤波器。Ma[24] 提出 MLO，融合几何和语义信息，估计自身运动并跟踪动态物体。Tian 提出一种名为 DL-SLOT 的方法[25]，使用 3D 激光雷达同时进行定位和目标跟踪，并利用 G2O 在后端做滑窗优化。

在视觉 SLAM 中，也有类似的研究，主要起源于多体运动恢复结构 (Multi Body Structure From Motion,MBSFM) 或多体运动估计问题[26]。在视觉 SLAM 中，对 SLAMMOT 的通常处理是将其分为多运动分割和对目标跟踪重建两部分。多运动分割的目的是对拥有相同运动状态的特征点进行聚类，可以使用子空间聚类[27]、多模型拟合[28] 等方法。目标跟踪重建的目的是输出自身和物体的轨迹，同时重建出静态环境和物体的结构。

如 Kundu[29] 在 2011 年利用 MBSFM 实现了特征跟踪、运动分割、视觉 SLAM、动态物体跟踪等多个任务。但系统被限制运行在平面上，也就是说无法估计物体的 SE(3) 位姿。Sabzevari[5] 提出基于多轨迹矩阵的投影因子分解的多运动估计方法，并利用车辆运动学约束来提高效率。Wang 和 Luo[27,30] 使用无语义信息的多

运动分割对场景中的不同运动模型进行分割和拟合，并通过降维提高子空间聚类的性能。

Henein 在 2018 年提出一种将刚性物体的运动建模到因子图中的方法[31]，随后 Zhang 和 Henein 在此基础上加入了光流估计和实例分割，提出了完整的视觉 SLAMMOT 系统，并利用因子图在后端进行优化[10,32,33]。在 AirDOS 中，Qiu 等[12]尝试利用动态物体进行定位，而非简单将其移除。所使用的方法是将物体由刚体建模为铰接物体，这种形式使得他们的工作可以建立在 Zhang 等人的工作基础上并进行扩展。实验结果显示，可以通过将物体建模为铰接物体的方式提升物体在动态环境下的定位准确性，实现了利用动态物体进行定位的效果。

SLAMMOT 过去主要是在 SLAM 的基础上对动态物体进行跟踪，一定程度上解决了目标检测器对 SLAM 带来的影响。而近年一部分研究[10,12]发现动态物体不仅可以被简单移除，而且可以再次利用并辅助自身定位，这样不但增强了机器人在动态环境下的定位能力，还建立起了机器人感知和定位之间的联系，有利于构建更自主智能、具有语义感知和空间感知的机器人系统。国内外目前在利用动态物体信息的 SLAMMOT 的研究上还处于探索阶段，以何种形式表示物体，如何更好的利用语义物体提供的信息，如何在动态环境下保持对自身位姿和物体位姿的准确估计，如何联合优化使自身信息和物体信息可以相互促进，均有待研究。

1.3 论文研究内容及章节安排

本文针对于动态环境下的 SLAM 问题展开研究，研究内容和章节安排如下：

第一章介绍本文研究的内容和研究意义，对国内外在动态环境下的 SLAM 和多传感器融合 SLAM 领域的相关言工作进行介绍，介绍了论文各章节的研究内容和章节安排。

第二章对于动态环境下 SLAMMOT 涉及的理论进行介绍。首先介绍了 SLAM 和 SLAMMOT 问题的定义和经典理论，然后介绍了多目标跟踪的基本方法和流程，最后是基于因子图模型的 SLAM 问题的构建和求解方法。

第三章提出一种基于深度补全的视觉激光多传感器融合 SLAM。介绍了使用的深度补全方法和视觉里程计、激光里程计的设计思路，然后建立因子图优化模型进行求解。

第四章提出一种名为动态配准的点云匹配方法，实现在动态环境下对自运动估计和运动目标的检测。首先介绍了目标检测和移除方法，其次提出了动态配准算法的实现流程和动态配准的迭代过程。

第五章在第四章工作的基础上，将动态配准扩展为一个完整的 SLAM 系统。

首先设计了激光 SLAM 算法和多目标跟踪，然后将 SLAM 和 MOT 分别建模为因子图模型中的节点和因子，再分别建立了松耦合和紧耦合的同时定位建图和目标跟踪算法。对于紧耦合的模型，利用因子图进行优化，实现同时利用动静态物体定位的 SLAM。

第六章对全文进行总结并对未来的研究进行展望。

第三四五章是本文的核心贡献和创新点，对于提出的算法，在章末都进行实验分析和算法对比。

第二章 动态环境下 SLAMMOT 理论基础

本章对本文涉及到的核心问题，同时定位建图和目标跟踪 SLAMMOT，涉及的基础理论进行介绍。包括 SLAM 的基础理论，多目标跟踪方法，然后介绍同时定位建图和目标跟踪理论。由于文章多次使用因子图对 SLAM 进行优化，因此也对因子图理论进行简单介绍。

2.1 同时定位建图与目标跟踪

SLAMMOT 的提出是为了解决动态环境下的 SLAM 定位问题。关于 SLAMMMOT 的历史，本文已在1.2.3节进行了详细介绍。作为理论基础，本节对 Wang[4] 等人提出的基于贝叶斯理论的 SLAMMOT 进行介绍。在第五章，将介绍本文在 Zhang[10]2020 年理论的基础上提出的结合多目标跟踪的因子图 SLAMMOT。

2.1.1 基于贝叶斯理论的 SLAM

首先做出以下规定：以 k 表示离散时刻，u_k 表示控制输入，z_k 表示激光观测，x_k 表示机器人状态，$\boldsymbol{M}_k$ 表示拥有 l 个路标 $m^1,\ldots,m^l$ 的集合，并定义如式 (2-1) 所示的集合：

$$\begin{aligned} \boldsymbol{X}_k &\triangleq \{x_0, x_1, \ldots, x_k\} \\ \boldsymbol{Z}_k &\triangleq \{z_0, z_1, \ldots, z_k\} \\ \boldsymbol{U}_k &\triangleq \{u_1, u_2, \ldots, u_k\} \end{aligned} \tag{2-1}$$

按照经典 SLAM 的理论，SLAM 需要在给定观测 $\boldsymbol{Z}_k$ 和控制输入 $\boldsymbol{U}_k$ 的前提下估计机器人状态 $\boldsymbol{X}_k$ 和路标点位置 $\boldsymbol{M}_k$。那么在静态环境假设和当前运动具有马尔可夫性的假设的前提下，使用贝叶斯法则得到计算 SLAM 的递归公式如式 (2-2) 所示：

$$\begin{aligned} &p\left(x_k, \boldsymbol{M}_k \mid \boldsymbol{Z}_k, \boldsymbol{U}_k\right) \propto p\left(z_k \mid x_k, \boldsymbol{M}_k\right) \\ &\cdot \int p\left(x_k \mid x_{k-1}, u_k\right) p\left(x_{k-1}, \boldsymbol{M}_{k-1} \mid \boldsymbol{Z}_{k-1}, \boldsymbol{U}_{k-1}\right) dx_{k-1} \end{aligned} \tag{2-2}$$

可以用贝叶斯网络直观地表示在 SLAM 过程中观测和运动之间的关系，由于机器人的位姿是时序变量，因此是一个动态贝叶斯网络，如图 2-1所示。按照原论文所述原理，重新绘制了示意图。

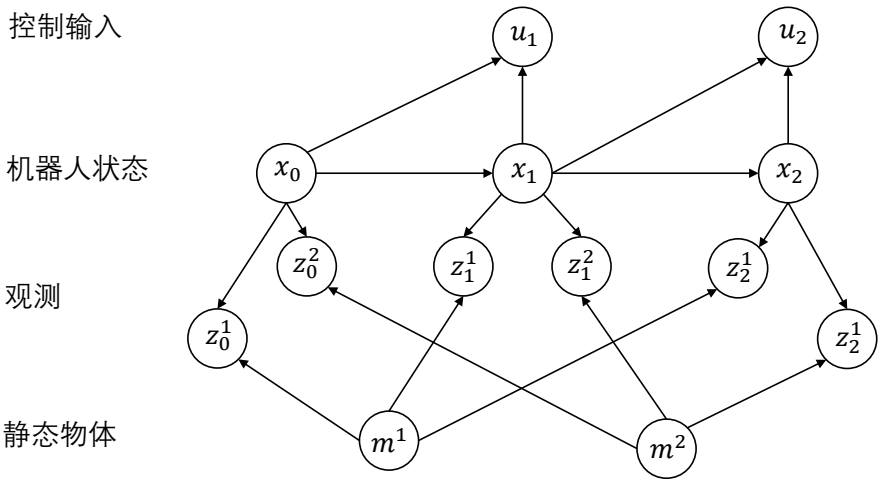


图 2-1 SLAM 的动态贝叶斯网络

2.1.2 基于贝叶斯理论的 SLAMMOT

运动目标跟踪同样可表示在贝叶斯框架下，根据贝叶斯法则可计算如式 (2-3) 所示。其中 o_k 为目标运动状态，s_k 为物体的运动模型，而此时通常假设自身是静止的。

$$p\left(o_k, s_k \mid \boldsymbol{Z}_k\right) \propto p\left(o_k \mid \boldsymbol{Z}_k, s_k\right) \cdot p\left(s_k \mid \boldsymbol{Z}_k\right) \tag{2-3}$$

式 (2-3) 表示目标跟踪问题可以分为模型估计 $p\left(s_k \mid Z_k\right)$ 和状态估计 $p\left(o_k \mid Z_k, s_k\right)$ 两步进行。实际中，往往将物体的模型定义为集合几个运动模型的集合，如匀速模型、恒加速模型、匀转速率模型等[34]。因此模型的估计可以简化为模型选择问题。对此，已有 GPBn(n 表示模型的个数)[35]、IMM[36] 等经典方法。IMM 大大减少了 GPB2 的计算量，性能又优于 GPB1，在移动机器人领域得到了大量的应用[23,37]，本节将随后对 IMM 进行介绍，并在5.3节中使用。

定义 k 时刻第 i 个物体的完整状态 $\boldsymbol{y}_k^i$ 及其集合 $\boldsymbol{Y}_k$。$\boldsymbol{y}_k^i$ 是包括其运动模型 s_k^i 和状态 y_k^i 的完整状态，也称广义物体[4]。这准物体可以处于静止、运动，也可能在静止和运动之间切换，因此可以表示 SLAM 中的路标和目标跟踪中的运动物体。

$$\begin{aligned} \boldsymbol{y}_k^i &\triangleq \left\{y_k^i, s_k^i\right\} \\ \boldsymbol{Y}_k &\triangleq \left\{\boldsymbol{y}_k^1, \boldsymbol{y}_k^2, \ldots, \boldsymbol{y}_k^l\right\} \end{aligned} \tag{2-4}$$

用广义物体集 $\boldsymbol{Y}_k$ 替换式 (2-2) 中的路标集 $\boldsymbol{M}_k$，则可以得到具有广义物体的

SLAM 的贝叶斯递归公式:

$$\begin{aligned} p\left(x_k, \boldsymbol{Y}_k \mid U_k, Z_k\right) \propto \quad & p\left(z_k \mid x_k, \boldsymbol{Y}_k\right) \iint p\left(x_k \mid x_{k-1}, u_k\right) p\left(\boldsymbol{Y}_k \mid \boldsymbol{Y}_{k-1}\right) \\ & \cdot p\left(x_{k-1}, \boldsymbol{Y}_{k-1} \mid Z_{k-1}, U_{k-1}\right) dx_{k-1} d\boldsymbol{Y}_{k-1} \end{aligned} \tag{2-5}$$

本文将式 (2-5) 称为广义 SLAMMOT，它可以在 SLAM 的框架下对广义物体进行建模，然而实际中对广义物体的运动模型进行准确估计计算成本很高，并且通常难以实现 [4]。本文在5.5节末介绍了这一问题，并提出一种优化方法以避免了广义 SLAMMOT 面对的问题。

Wang 对此提出的解决方法是 SLAM with DATMO，即将物体分为静止物体和运动物体分别跟踪，如其在 2003 年 [3] 首次提出的方法。而其 2007 年的工作 [4] 将 SLAM with DATMO 推广到了广义的 SLAMMOT，但仅在理论上进行了推导和介绍，主要使用的方案还是 SLAM with DATMO，可表示为式 (2-6)。

$$\begin{aligned} p\left(x_k, \boldsymbol{O}_k, \boldsymbol{M}_k \mid \boldsymbol{Z}_k, \boldsymbol{U}_k\right) \propto \quad & p\left(z_k^o \mid x_k, \boldsymbol{O}_k\right) p\left(\boldsymbol{O}_k \mid \boldsymbol{Z}_{k-1}^o, \boldsymbol{U}_k\right) \\ & \cdot p\left(z_k^m \mid x_k, \boldsymbol{M}_k\right) p\left(x_k, \boldsymbol{M}_k \mid \boldsymbol{Z}_{k-1}^m, \boldsymbol{U}_k\right) \\ = \quad & p\left(z_k^o \mid \boldsymbol{O}_k, x_k\right) \\ & \cdot \int p\left(\boldsymbol{O}_k \mid \boldsymbol{O}_{k-1}\right) p\left(\boldsymbol{O}_{k-1} \mid \boldsymbol{Z}_{k-1}^o, \boldsymbol{U}_{k-1}\right) d\boldsymbol{O}_{k-1} \\ & \cdot p\left(z_k^m \mid \boldsymbol{M}_k, x_k\right) \\ & \cdot \int p\left(x_k \mid u_k, x_{k-1}\right) p\left(x_{k-1}, \boldsymbol{M}_{k-1} \mid \boldsymbol{Z}_{k-1}^m, \boldsymbol{U}_{k-1}\right) dx_{k-1} \end{aligned} \tag{2-6}$$

式中，$\boldsymbol{O}_k = \left\{\boldsymbol{o}_k^1, \boldsymbol{o}_k^2, \ldots, \boldsymbol{o}_k^n\right\}$ 表示 n 个广义运动物体的状态，$\boldsymbol{o}_k^1$ 的定义见式 (2-4)。$M_k = \left\{m_k^1, m_k^2, \ldots, m_k^q\right\}$，表示 q 个静止物体的状态，定义如贝叶斯 SLAM 的路标集合。z_k^o 表示运动物体的观测，z_k^m 表示静止物体的观测。

式 (2-6) 表示 SLAM with DATMO 可以被拆解为运动物体和静止物体的独立后验。同时，注意到 $p\left(z_k^o \mid \boldsymbol{O}_k, x_k\right)$，表示 DATMO 应该考虑机器人位姿 x_k，说明 DATMO 和 SLAM 是相互影响的，从理论上证明了其 2002 年 [38] 首次提出的方法和实验结果。

这种方案的核心在于对运动物体的检测，只要能准确分割出运动物体，就可以将广义 SLAMMOT 简化 SLAM with DATMO。然而运动分割的方法需要单独设计的，Wang 使用了运动一致性检验和运动物体地图两种方法进行分割 [4]。从式 (2-6) 可以看出，SLAM with DATMO 受到运动物体检测精度的限制。在贝叶斯的框架下，准确分割物体的状态需要进行广义 SLAMMOT 估计，这在实际中难以计算，并会使 MOT 降低 SLAM 本身的性能，而对于运动物体分割又缺乏理论支撑，只

能使用近似方法。

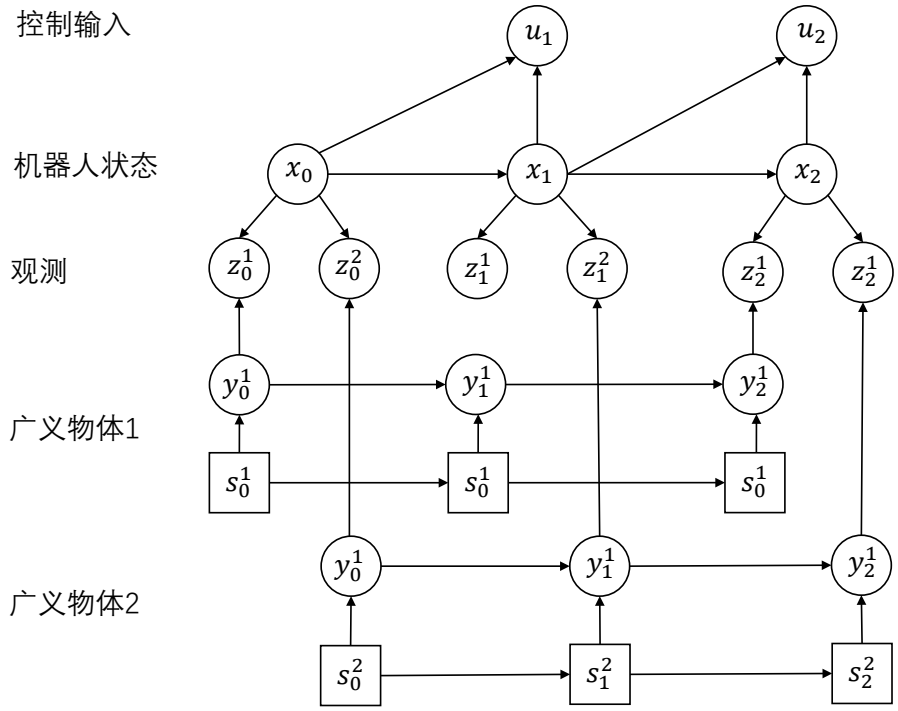


图 2-2 具有广义物体的 SLAMMOT 的动态贝叶斯网络

图 2-2显示了式 (2-5) 所示的具有广义物体的 SLAMMOT 的动态贝叶斯网络结构。此网络表示观测时间步长为三的情况下，具有两个广义物体，并估计自身和广义物体的状态。

2.2 多目标跟踪

多目标跟踪包括数据关联、状态估计、轨迹关联等步骤。定义系统的状态方程和观测方程为：

$$\begin{aligned} \boldsymbol{x}_k &= f_k\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_k \\ \boldsymbol{z}_k &= h\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{v}_k \end{aligned} \tag{2-7}$$

式中，$\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 分别是相互独立且为零均值的高斯白噪声，协方差为 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$。

不同目标可能有不同的运动模型。车辆在直线行驶和转弯时也表现出的不同的运动模型。大多数 3D 多目标跟踪的文献假定物体处于恒速运动，而由于物体运动状态和运动信息未知，为了更好地建模行人车辆地状态，处理因不同运动模型引起的不确定性，使用 IMM 滤波器在不同状态模型之间切换。

假设不同运动模型之间的切换是马尔可夫过程，r 个模型之间的切换由转移概

率矩阵 $\prod$ 控制：

$$\prod = \begin{bmatrix} \pi_{11} & \cdots & \pi_{1r} \\ \cdots & \cdots & \cdots \\ \pi_{r1} & \cdots & \pi_{rr} \end{bmatrix} \\ \sum_{j=1}^{r} \pi_{j|i} = 1 \tag{2-8}$$

式中，π_{ij}，或记为 $\pi_{j|i}$，表示从系统从模型 i 切换到模型 j 的概率。

定义 μ_k^i 为目标在 k 时刻处于模型 i 的概率，μ_k^{ij} 或 $\mu_k^{j|i}$ 表示目标在 k 时刻从模型 i 的概率转移到模型 j 的条件概率。若已知 $k-1$ 时刻各模型的目标状态估计 $\hat{\boldsymbol{x}}_k$、协方差矩阵 $\boldsymbol{P}_{k-1}$、模型概率为 μ_{k-1}、状态转移矩阵为 $\prod_{k-1}$。则 IMM 的一次循环过程为：

(1) 模型输入交互

以模型概率 $\mu_{k-1}^{i|j}$ 作为权重分别计算交互后的状态估计和协方差的加权和，得到交互后的状态估计 $\hat{\boldsymbol{x}}_{k-1}^{0j}$ 和协方差矩阵 $\boldsymbol{P}_{k-1}^{0j}$，称为混合条件或混合初始条件，以上标 $0j$ 表示滤波器 j 属于混合状态。

$$\mu_{k-1}^{j|i} = \frac{\pi_{k-1}^{j|i}\mu_{k-1}^{i}}{\sum_{i=1}^{r}\pi_{k-1}^{j|i}\mu_{k-1}^{i}} \\ \hat{\boldsymbol{x}}_{k-1}^{0j} = \sum_{i=1}^{r}\mu_{k-1}^{j|i}\hat{\boldsymbol{x}}_{k-1}^{i} \\ \boldsymbol{P}_{k-1}^{0j} = \sum_{i=1}^{r}\mu_{k-1}^{j|i}\left[\boldsymbol{P}_{k-1}^{i} + \left(\hat{\boldsymbol{x}}_{k-1}^{i} - \hat{\boldsymbol{x}}_{k-1}^{0j}\right)\left(\hat{\boldsymbol{x}}_{k-1}^{i} - \hat{\boldsymbol{x}}_{k-1}^{0j}\right)^{\top}\right] \tag{2-9}$$

式中，c_{k-1}^{j} 表示输入交互后目标处于模型 j 的概率。

(2) 模型滤波

将交互输入值带入滤波器中，得到各模型的滤波估计值和协方差矩阵。由于建立的运动模型是非线性的，因此可以使用 EKF 或 UKF 进行估计，将在后续进行介绍。以下假设模型已经过线性化，$\boldsymbol{F}_{k-1}$ 为线性化后的状态转移矩阵。

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1}^{j} &= \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1}^{0j} \\
\boldsymbol{P}_{k|k-1}^{j} &= \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^{0j}\boldsymbol{F}_{k-1}^{\top} + \boldsymbol{Q}_{k-1}^{j} \\
\boldsymbol{G}_k &= \boldsymbol{P}_{k|k-1}^{j}\boldsymbol{H}_k\left(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}^{j}\boldsymbol{H}_k + \boldsymbol{R}_k\right)^{-1} \\
\hat{\boldsymbol{x}}_k^{j} &= \hat{\boldsymbol{x}}_{k|k-1}^{j} + \boldsymbol{G}_k\left(\hat{\boldsymbol{z}}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}^{j}\right) \\
\boldsymbol{P}_k^{j} &= \left(I - \boldsymbol{G}_k\boldsymbol{H}_k\right)\boldsymbol{P}_{k|k-1}^{j}
\end{aligned}
\tag{2-10}
$$

(3) 模型概率更新

模型残差 $\tilde{\boldsymbol{v}}_k^j$，协方差矩阵为 $\boldsymbol{S}_k^j$，可表示为：

$$
\begin{aligned}
\tilde{\boldsymbol{v}}_k^{j} &= \hat{\boldsymbol{z}}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}^{j} \\
\boldsymbol{S}_k^{j} &= \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}^{j}\boldsymbol{H}_k + \boldsymbol{R}_k
\end{aligned}
\tag{2-11}
$$

Dim 表示测量模型的维数，得到模型 *j* 的似然函数模型：

$$
\boldsymbol{\Lambda}_k^{j} \sim \mathcal{N}(\tilde{\boldsymbol{v}}_k^{j}, \boldsymbol{S}_k^{j}) = \frac{1}{\sqrt{(2\pi)^{Dim}\left|\boldsymbol{S}_k^{j}\right|}}\exp\left\{-\frac{1}{2}\tilde{\boldsymbol{v}}_k^{j\top}\boldsymbol{S}_k^{j\,-1}\tilde{\boldsymbol{v}}_k^{j}\right\} \tag{2-12}
$$

则模型概率可以按式 (2-13) 更新为：

$$
\begin{aligned}
c_k^{j} &= \pi_{k-1}^{j|i}\mu_{k-1}^{i} \\
C_k &= \sum_{i=1}^{r}\boldsymbol{\Lambda}_k^{j}c_k^{j} \\
\mu_k^{j} &= \frac{1}{C_k}\boldsymbol{\Lambda}_k^{j}c_k^{j}
\end{aligned}
\tag{2-13}
$$

(4) 模型输出整合各模型的状态估计和协方差，即

$$
\begin{aligned}
\hat{\boldsymbol{x}}_k &= \sum_{i=1}^{r}\mu_k^{i}\hat{\boldsymbol{x}}_k^{i} \\
\boldsymbol{P}_k &= \sum_{i=1}^{r}\mu_k^{i}\left\{\boldsymbol{P}_k^{i} + \left(\hat{\boldsymbol{x}}_k^{i} - \hat{\boldsymbol{x}}_k\right)\left(\hat{\boldsymbol{x}}_k^{i} - \hat{\boldsymbol{x}}_k\right)^{\top}\right\}
\end{aligned}
\tag{2-14}
$$

由于目标的运动模型是非线性的，因此无法直接用卡尔曼滤波进行处理。在处理非线性估计问题时，广泛使用的有扩展卡尔曼滤波、无迹卡尔曼滤波和粒子滤波等方法。对于常用的非线性估计方法进行介绍和推导。无迹卡尔曼滤波 UKF

基于无迹变换，与 EKF 使用的线性化方法不同，UKF 是直接求出状态分布的均值和协方差。相比于 EKF，UKF 避免了求雅可比矩阵，通过非线性函数传递所选的 *Sigma* 点。然后可以从这些传播的 *Sigma* 点恢复新的高斯分布。相比于 EKF，UKF 的估计精度更高。

UKF 的计算过程如下：

(1) 无迹变换

若 $k-1$ 时刻 $\boldsymbol{x}_{k-1}$ 的状态估计为 $\bar{\boldsymbol{x}}_{k-1}$，对应的协方差矩阵为 $\boldsymbol{P}_{k-1}$。选择 $2n+1$ 个 *Sigma* 点集 $\boldsymbol{\chi}_i$ 近似状态的分布

$$\begin{aligned}
\boldsymbol{\chi}_{k-1}^{0} &= \bar{\boldsymbol{x}}_{k-1} \\
\boldsymbol{\chi}_{k-1}^{i} &= \bar{\boldsymbol{x}}_{k-1} + \left(\sqrt{(n+\lambda)\boldsymbol{P}_{k-1}}\right)_{i}, \quad i = 1,\ldots,n \\
\boldsymbol{\chi}_{k-1}^{i} &= \bar{\boldsymbol{x}}_{k-1} - \left(\sqrt{(n+\lambda)\boldsymbol{P}_{k-1}}\right)_{i-n}, \quad i = n+1,\ldots,2n
\end{aligned} \tag{2-15}$$

式中 $\left(\sqrt{(n+\lambda)\boldsymbol{P}_{k-1}}\right)_{i-n}$ 表示矩阵的均方根的第 i 行。

均值的权重系数 $w_0^{(m)}$ 和协方差的权重系数 $w_0^{(c)}$ 分别为：

$$\begin{aligned}
w_0^{(m)} &= \lambda/(n+\lambda) \\
w_0^{(c)} &= \lambda/(n+\lambda) + \left(1-\alpha^2+\beta\right) \\
w_i^{(m)} &= w_i^{(c)} = 1/\left(2n+2\lambda\right), i = 1,\ldots,n
\end{aligned} \tag{2-16}$$

再将采样点经过非线性变换，得到：

$$\hat{\boldsymbol{\chi}}_{k|k-1}^{i} = f(\boldsymbol{\chi}_{k-1}^{i}), i = 0,\ldots,2n \tag{2-17}$$

(2) 预测

状态预测均值 $\hat{\boldsymbol{x}}_{k|k-1}$ 和预测协方差 $\boldsymbol{P}_{k|k-1}$ 分别为

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1} &= \sum_{i=0}^{2n} w_i^{(m)} \hat{\boldsymbol{\chi}}_{k|k-1}^{i} \\
\boldsymbol{P}_{k|k-1} &= \sum_{i=0}^{2n} w_i^{(c)} \left(\hat{\boldsymbol{\chi}}_{k|k-1}^{i} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\hat{\boldsymbol{\chi}}_{k|k-1}^{i} - \hat{\boldsymbol{x}}_{k|k-1}\right)^{\top} + \boldsymbol{Q}_{k-1}
\end{aligned} \tag{2-18}$$

(3) 更新

根据观测模型进行 *Sigma* 点变换, 即

$$\hat{\mathcal{Z}}_{k|k-1}^{i} = \boldsymbol{h}\left(\mathcal{X}_{k|k-1}^{i}\right), i = 0,\ldots,2n \tag{2-19}$$

观测值和观测协方差可表示为：

$$\begin{aligned}\hat{\boldsymbol{z}}_{k|k-1} &= \sum_{i=0}^{2n} w_i^{(m)} \hat{\mathcal{Z}}_{k|k-1}^i \\ \boldsymbol{S}_k &= \sum_{i=0}^{2n} w_i^{(c)} \left(\hat{\mathcal{Z}}_{k|k-1}^i - \hat{\boldsymbol{z}}_{k|k-1}\right)\left(\hat{\mathcal{Z}}_{k|k-1}^i - \hat{\boldsymbol{z}}_{k|k-1}\right)^\top + \boldsymbol{R}_k\end{aligned} \tag{2-20}$$

状态和观测的互协方差 $\boldsymbol{C}_k$ 和滤波增益 $\boldsymbol{K}_k$ 为：

$$\begin{aligned}\boldsymbol{C}_k &= \sum_{i=0}^{2n} w_i^{(c)} \left(\mathcal{X}_{k|k-1}^i - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\hat{\mathcal{Z}}_{k|k-1}^i - \hat{\boldsymbol{z}}_{k|k-1}\right)^\top \\ \boldsymbol{K}_k &= \boldsymbol{C}_k \boldsymbol{S}_k^{-1}\end{aligned} \tag{2-21}$$

最后，k 时刻估计的均值和协方差为：

$$\begin{aligned}\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}\right) \\ \boldsymbol{P}_k &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{S}_k \boldsymbol{K}_k^\top\end{aligned} \tag{2-22}$$

多目标跟踪的一个核心问题是数据关联，即如何将有噪声的检测结果准确分配给先前的轨迹。往往通过找到检测结果和轨迹之间的度量函数，如基于外观、运动等信息。数据关联将新检测到的物体分配到之前的轨迹中，或者基于物体的运动产生新的轨迹，并为每个轨迹分配唯一 ID。数据关联滤波器可分为两类，一类是确定性滤波器，一类是概率滤波器。确定性滤波器如最近邻关联，根据检测和轨迹之间的距离进行关联，但当多个检测接近时，容易出现误分配。概率类滤波器如 PDA、JPDA，是对检测来自不同目标的概率进行加权并计算联合概率，避免了确定性滤波器带来的关联错误。

本文根据不同情况，选择了不同的数据关联算法。如在4.3.3节和5.5节使用基于确定性关联的匈牙利匹配和全局最近邻关联，在5.3和5.4节使用基于概率关联的联合概率数据关联 JPDA，因此在本节对所使用的数据关联方法进行介绍。

以下将历史检测形成的轨迹称为目标，将当前检测结果称为观测。通过式 (2-23) 计算 $\boldsymbol{x}$ 的状态，核心在于对关联概率 $\beta_j(k)$ 的求解。

$$\begin{aligned}\hat{\boldsymbol{x}}^t(k \mid k) &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{v}(k)) \\ \boldsymbol{v}_k &= \sum_{j=1}^{m_k} \beta_j^t(k)(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k)\end{aligned} \tag{2-23}$$

式中关联概率 $\beta_j^t(k)$ 表示观测 j 来自目标 t 的概率。

如果用 $\omega_j^t(\theta(k))$ 表示观测是否来自目标，则 $\beta_j^t(k)$ 可表示为：

$$\beta_j^t(k)=\sum_{\theta(k)} P\left(\theta(k)\mid Z^k\right)\omega_j^t(\theta(k)) \tag{2-24}$$

k 时刻的 $\theta(k)$ 的后验概率 $P\left(\theta(k)\mid Z^k\right)$ 为：

$$P\left(\theta(k)\mid Z^k\right)=\frac{1}{c}\frac{\Phi!}{V^{\Phi}}\prod_{j=1}^{m_k}\left\{N_{t_j}\left[Z_j(k)\right]\right\}^{\tau_j}\prod_{t=1}^{T}\left(P_D^t\right)^{\delta_t}\left(1-P_D^t\right)^{1-\delta_t} \tag{2-25}$$

式中，c 是归一化常数；Φ 为虚假观测事件数 (误检数)；P_D^t 是目标 t 的检测概率；δ_t 是目标指示器，如果目标 t 在事件 $\theta(k)$ 中与某观测相连接，则 $\delta_t=1$，反之为 0；τ_j 是观测指示器，若如果观测 j 在事件 $\theta(k)$ 中与某目标相连接，则 $\tau_j=1$，反之为 0；$N_{t_j}\left[Z_j(k)\right]$ 表示 $Z_j(k)$ 服从高斯分布，如式 (2-26) 所示；V 是航迹有效门体积，如对于 q 维观测，验证区域的体积如式 (2-27) 所示：

$$N_{t_j}\left[Z_j(k)\right]=\frac{1}{\sqrt{\det\left(2\pi\boldsymbol{S}_{t,j}\right)}}\exp\left(-\frac{1}{2}\left(\boldsymbol{z}_{t,j|k}-\hat{\boldsymbol{z}}_{t,j|k-1}\right)^{\top}\boldsymbol{S}_{\boldsymbol{t,j}}^{-1}\left(\boldsymbol{z}_{t,j|k}-\hat{\boldsymbol{z}}_{t,j|k-1}\right)\right) \tag{2-26}$$

$$V_k=\frac{\boldsymbol{\pi}^{\frac{q}{2}}}{\Gamma\left(\frac{q}{2}+1\right)}\sqrt{|\gamma S_k|} \tag{2-27}$$

式 (2-27) 中 S_k 为真实观测的新息协方差，γ 是门限阈值。

2.3 因子图算法

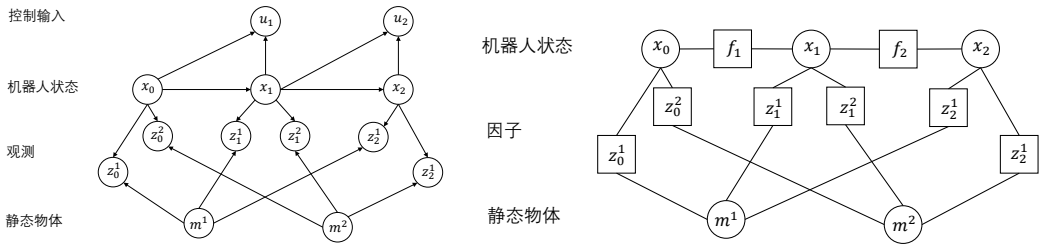
本节在2.1.1节的基础上进行推导。因子图是一种二分图，由两种节点组成，一种是变量节点，一种是因子节点。图 2-3显示了将基于动态贝叶斯网络的的 SLAM 转换为对应因子图的形式。用集合 $\boldsymbol{G}=(\boldsymbol{X},\boldsymbol{F},\boldsymbol{E})$ 表示因子图模型，因子图 $\boldsymbol{G}$ 可以表示对函数 $f(\boldsymbol{X})$ 的因式分解：

$$\boldsymbol{F}(\boldsymbol{X})=\prod_i f_i\left(\boldsymbol{X}_i\right) \tag{2-28}$$

因子图的最大后验估计为：

$$X^{MAP}=\arg\min_X f(X)=\arg\min_X\prod_i f_i\left(\boldsymbol{X}_i\right) \tag{2-29}$$

统一以 $\boldsymbol{y}_i$ 表示观测量，$\boldsymbol{x}_i$ 表示待估计状态，f 表示观测方程，则每项因子可建



(a) (b)

图 2-3 SLAM 的不同表示。(a) 动态贝叶斯网络 SLAM；(b) 因子图形式的 SLAM。

模为高斯噪声的形式：

$$f_i(\boldsymbol{X}_i) \propto \exp\{-\frac{1}{2}\|\boldsymbol{y}_i - f(\boldsymbol{x}_i)\|^2_{\Sigma_i}\} \tag{2-30}$$

将式 (2-30) 带入式 (2-29)，取负对数并进行化简，则最大后验估计问题可以转换为一个非线性最小二乘优化问题：

$$X^{MAP} = \arg\min_X \sum_i \|\boldsymbol{y}_i - f(\boldsymbol{x}_i)\|^2_{\Sigma_i} \tag{2-31}$$

在高斯白噪声的条件下，对于因子图而言，其最大后验估计的求解等价一个非线性最小二乘估计。对于非线性最小二乘的求解，可以使用直接法，即做线性化后转为线性最小二乘，然后进行直接求解。也可以使用迭代的非线性优化方法求解。遵循 SLAM 中通常使用的优化方法，因此问题又被转换为非线性优化问题。本文的因子图使用 LM 法和狗腿法（Dog-Leg）进行求解，以下对这些方法进行介绍。

取 $\psi(\boldsymbol{x})$ 为其中误差项的和。令 $e(\boldsymbol{x}) = [e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), \cdots, e_m(\boldsymbol{x})]$，其中的元素 $e_i(\boldsymbol{x}) = \Sigma_i^{-\frac{1}{2}} \cdot (\boldsymbol{y}_i - f(\boldsymbol{x}_i))$，再扩展 $\psi(\boldsymbol{x})$ 为误差函数 $e(\boldsymbol{x})$ 的形式，则 $\psi(\boldsymbol{x})$ 可写为：

$$\psi(\boldsymbol{x}) = \sum_i \|\boldsymbol{y}_i - f(\boldsymbol{x}_i)\|^2_{\Sigma_i} = \sum_i \|e_i(\boldsymbol{x})\|^2 = \|e(\boldsymbol{x})\|^2 = e(\boldsymbol{x})^\top \cdot e(\boldsymbol{x}) \tag{2-32}$$

记 $\psi(\boldsymbol{x})$ 导数为梯度 $\mathbf{g}_\psi$，$\psi(\boldsymbol{x})$ 的二阶导数为海森矩阵 $\mathbf{H}_\psi$，$e(\boldsymbol{x})$ 的雅可比矩阵为 $\mathbf{J}_e$。则 $\mathbf{g}_\psi = \nabla\psi(\boldsymbol{x})$，$\mathbf{H}_\psi = \nabla^2\psi(\boldsymbol{x})$。对于 $\mathbf{g}_\psi$ 和 $\mathbf{H}_\psi$，分别写出 $\mathbf{g}_\psi$ 中第 j 行元素和 $\mathbf{H}_\psi$ 中第 k 行 j 列的元素，如下所示：

$$\begin{aligned} \frac{\partial\psi}{\partial\boldsymbol{x}_j} &= \sum_{i=1}^m 2 \cdot e_i(\boldsymbol{x}) \cdot \frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}_j} \\ \frac{\partial\psi}{\partial\boldsymbol{x}_k\boldsymbol{x}_j} &= 2\sum_{i=1}^m (\frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}_k} \cdot \frac{\partial e_i(\boldsymbol{x})}{\partial\boldsymbol{x}_j} + e_i(\boldsymbol{x}) \cdot \frac{\partial^2 e_i(\boldsymbol{x})}{\partial\boldsymbol{x}_k\partial\boldsymbol{x}_j}) \end{aligned} \tag{2-33}$$

$\mathbf{g}_\psi$ 可直接用 $\mathbf{J}_e$ 进行表示。当忽略高阶项时，$\mathbf{H}_\psi$ 也可近似用 $\mathbf{J}_e$ 表示，即：

$$\begin{aligned}\mathbf{g}_\psi &= \nabla\psi(\boldsymbol{x}) = 2 * \mathbf{J}_e^\top \cdot e(\boldsymbol{x}) \\ \mathbf{H}_\psi &= \nabla^2\psi(\boldsymbol{x}) \approx 2 * \mathbf{J}_e^\top * \mathbf{J}_e\end{aligned} \tag{2-34}$$

记 $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{(k)}$，而误差 $e(\boldsymbol{x})$ 的一阶导数是雅可比矩阵 $\mathbf{J}_e$。那么对 $e(\boldsymbol{x})$ 做一阶泰勒展开，得：

$$e(\boldsymbol{x}) \approx e(\boldsymbol{x}^{(k)}) + \nabla e(\boldsymbol{x}^{(k)}) \cdot \Delta\boldsymbol{x} = e(\boldsymbol{x}^{(k)}) + \mathbf{J}_e \cdot \Delta\boldsymbol{x} \tag{2-35}$$

将式 (2-35) 带入式 (2-32) 中，则对应非线性最小二乘优化函数 $\psi(\boldsymbol{x})$ 为：

$$\begin{aligned}\psi(\boldsymbol{x}) &\approx \left(e(\boldsymbol{x}^{(k)}) + \mathbf{J}_e\Delta\boldsymbol{x}\right)^\top \cdot \left(e(\boldsymbol{x}^{(k)}) + \mathbf{J}_e\Delta\boldsymbol{x}\right) \\ &= e^\top e + 2\Delta\boldsymbol{x}^\top\mathbf{J}_e^\top + \Delta\boldsymbol{x}^\top\mathbf{J}_e^\top\mathbf{J}_e\Delta\boldsymbol{x}\end{aligned} \tag{2-36}$$

令其导数等于 0，可计算为：

$$\begin{aligned}\nabla\psi(\boldsymbol{x}) &= \nabla\left(e^\top e + 2\Delta\boldsymbol{x}^\top\mathbf{J}_e^\top + \Delta\boldsymbol{x}^\top\mathbf{J}_e^\top\mathbf{J}_e\Delta\boldsymbol{x}\right) \\ &= 2\mathbf{J}_e^\top e + 2\mathbf{J}_e^\top\mathbf{J}_e\Delta\boldsymbol{x} = 0\end{aligned} \tag{2-37}$$

则可得到迭代更新量 $\Delta\boldsymbol{x}$ 的公式：

$$\mathbf{J}_e^\top \cdot \mathbf{J}_e \cdot \Delta\boldsymbol{x} = -\mathbf{J}_e^\top \cdot e \tag{2-38}$$

通常不会通过直接求逆计算 $\Delta\boldsymbol{x}$，而是使用 QR 分解等方法。总之，可以得到下一步的迭代点：

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x} \tag{2-39}$$

SLAM 系统若仅利用相邻两帧估计自身位姿，随着时间推移不可避免地会出现累计误差，因此在大规模场景或复杂环境中，定位结果不准确。为了减少误差，可以通过在姿态之间添加约束关系，构建因子图优化模型来进一步校正姿态，利用全局信息优化轨迹。本文将在第三章和第五章中使用因子图模型优化建立的 SLAM 系统。

2.4 本章小结

本章对 SLAMMOT 涉及的基础理论进行简单介绍。首先介绍了 Wang 建立在动态贝叶斯网络 SLAM 的同时定位建图和目标跟踪理论。然后介绍了多目标跟踪中数据关联和状态估计的基本方法，在后文将利用 MOT 多目标跟踪理论对物体进行跟踪和估计，将结果用于 SLAMMOT。最后，介绍了因子图模型的建立和优化求解过程，后文建立的因子图都在此方法基础上进行求解。

第三章 基于深度补全的视觉激光融合 SLAM

3.1 视觉激光融合 SLAM 算法流程

相机可以提供丰富的特征，因此在机器人导航领域发挥重要作用，但因相机本身的缺陷，各类视觉导航方案或多或少存在一些问题。如单目相机存在尺度漂移，难以直接应用于导航。双目和 RGB-D 相机可以实时提供 3D 信息，但 RGB-D 相机主要用于室内导航，因为深度探测范围较小，而且对光线非常敏感，很难直接应用于室外环境。双目相机需要通过立体匹配进行深度估计，但深度的范围会受到双目基线的限制。而视觉激光融合可以在室外提供类似 RGB-D 的效果，深度范围远，不受光线干扰，但激光雷达获取的点云在图像平面是稀疏的，点云无法与图像逐像素匹配，导致部分特征点处无深度值，这增大了视觉激光融合的难度。

本节提出一种视觉激光融合的方案，分为前端视觉激光融合里程计和后端因子图全局优化，如图 3-1所示。主要思路是通过深度补全技术解决图像和点云的数据融合问题，使得补全后的点云可以与图像逐像素对应，从而方便获取所有视觉特征点对应的深度值，避免了复杂的深度估计。因此视觉激光融合 SLAM 被拆分为视觉 SLAM 和激光 SLAM 两个系统。此外，为解决单一传感器估计不准确的问题，在后端利用因子图分别对两个传感器的估计结果进行融合，获得更为稳定准确的定位结果。

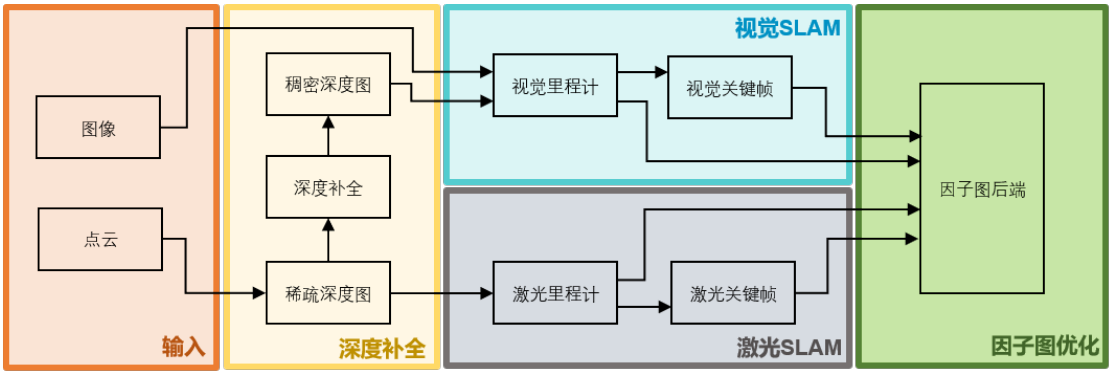


图 3-1 视觉激光 SLAM 融合流程图

首先对稀疏激光点云进行深度补全，获取稠密深度后与图像使用 RGBD-SLAM 进行视觉定位和建图。激光获取的深度在图像范围内非常稀疏，但通过深度补全技术实现稠密深度后，相比传统视觉 SLAM 使用的双目相机或 RGB-D 相机，则可以实现更远更精确的距离测量。同时使用传统激光 SLAM 方案对原始激光点云做激光定位和局部建图。因此前端视觉激光融合里程计可以输出视觉和激

光里程计的各类约束因子。最后，在系统的后端优化模块，利用视觉和激光提供的不同约束建立因子图进行优化。

3.2 稀疏深度补全

3.2.1 点云投影变换

激光雷达的 FoV(Field of view) 和相机不同，在单相机和单激光雷达融合的场景，首先需考虑其融合所使用的视图。而如果有多个相机可以和机械激光雷达进行融合，或使用固态激光雷达与单个相机融合，可以避免视图选择问题。出于简便考虑，本节选择在 SLAM 和目标检测领域都有广泛用途的激光雷达前视图和图像进行融合。

使用前视图和图像融合，只需将点云投影至图像平面，并过滤掉范围之外的点。将激光雷达坐标系下的点云转换至图像平面，涉及到激光雷达到相机的外参变换和相机到图像的投影变换。由于实验在 KITTI 数据集上进行，故以其标定关系进行介绍。若激光雷达绕 z 轴旋转 γ，则旋转矩阵 $\boldsymbol{R}_z$ 为：

$$\boldsymbol{R}_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-1}$$

同理，若分别绕 x 轴和 y 轴旋转 α 和 β，则对应的 $\boldsymbol{R}_x$ 和 $\boldsymbol{R}_y$ 为：

$$\boldsymbol{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \boldsymbol{R}_y = \begin{bmatrix} \cos\omega & 0 & \sin\omega \\ 0 & 1 & 0 \\ -\sin\omega & 0 & \cos\omega \end{bmatrix} \tag{3-2}$$

$\boldsymbol{T}$ 表示激光雷达到相机的欧式变换矩阵，为相机到激光雷达的外参。若发生的平移为 $\boldsymbol{t}$，则变换矩阵 $\boldsymbol{T}$ 为：

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z & \boldsymbol{t} \\ 0^\top & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0^\top & 1 \end{bmatrix} \tag{3-3}$$

$\boldsymbol{P}$ 为相机到图像平面的变换，为相机的内参，如式 (3-4) 所示。

$$\boldsymbol{P}=\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3-4}$$

式中，f_x 和 f_y 是相机在 uv 方向上的尺度因子，u_0 和 v_0 表示相机光心对应的像素坐标。由于数据集中的图像已去畸变，故内参中的畸变参数为 0。

当已知相机和激光雷达的内外参关系时，对于空间中一齐次坐标 $[x,y,z,1]^\top$，有如下变换：

$$Z_C\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=\boldsymbol{PT}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3-5}$$

在数据集中，已完成各传感器的内外参标定，可以直接从标定文件中读出对应的矩阵进行计算。由于激光雷达和相机的视角范围不同，因此当点云投影到图像平面时，需滤除相机视野范围外的投影点。同时为保证投影有意义，计算时取 $x>0$ 的部分，即仅用前方的点进行变换。变换和投影的结果如图 3-2中 (b) 所示。

(a) (b)

图 3-2 点云投影结果。(a) 原始点云；(b) 投影后点云。

3.2.2 深度补全

当激光点云投影到图像上时，虽然对应的深度准确，但非常稀疏，不利于图像和点云融合的后续任务。当点云投影到图像坐标系，并去除图像范围以外的点时，剩余的点云数量约为之前的 15%。对于稀疏点云导致的视觉和激光融合问题，文献 [39–41] 使用基于直接法的视觉 SLAM，避免了稀疏深度导致的特征点对应深度缺失。特别的是，韩国科学技术院的研究者在他们的 DVL SLAM [39,40] 中, 提出一

个观点：对于低线数雷达 (如 16 线) 和低分辨率相机，使用直接法是更好的形式。

然而，直接法对光照和纹理要求较高，且易受到噪声干扰。实际中，基于特征点的 SLAM 相比基于直接法的 SLAM 表现更稳定更准确。另外，随着激光雷达技术的进步和成本的下降[42]，高线数激光雷达逐渐出现在越来越多的场合。固态激光雷达的出现[43]，可以获得相比机械激光雷达更加稠密点云，还能显著降低成本，因为固态激光雷达无需机械旋转部件，不易受损，也不需要经常性的维护和更换。

如 Durlar 数据集[44] 使用 128 线的 Ouster 雷达；Dair-v2x 数据集[45] 在路端使用 300 线的激光雷达，车端使用 Velodyne128 线的雷达；本文实验所使用的 KITTI 数据集[46]，使用 Velodyne 64 线的激光雷达。因此，对具有高线数雷达的场合，不再适合使用视觉直接法和稀疏深度的形式。

一些研究尝试利用激光点云为视觉特征点提供深度。这种方法的挑战在于，尽管激光获取的点云已经在空间中已经较稠密，但点云对于的深度无法和图像中的像素点一一对应，还需要通过其他模块辅助以确定所有特征点处的深度值，这成为视觉激光融合 SLAM 算法的一个难点。如 VLOAM[47] 通过激光、三角测量等多种方式建立起深度图以进行基于特征的 SLAM，LIMO[48] 使用多帧点云建立相对稠密的深度，避免了从运动中估计深度的问题。但以上方法均未完全解决稀疏深度带来的影响，都依赖相对复杂的后续判断以保证深度估计的准确性。

深度补全是从稀疏深度值中估计出稠密的深度信息，并与图像像素建立一一对应的关系。在视觉激光融合的场景下，稀疏深度是由激光雷达点云投影到图像平面产生。目前主要有基于深度学习的方法[49] 和基于点云插值等[50] 的几何方法。基于深度学习的点云补全算法尽管相比几何方法可以取得更低的误差，但这种补全却改变了点云的原始结构，且不具有几何意义，有可能对后续任务产生影响。慕尼黑工业大学在 2022 年提出一种 RGB-L 算法[51]，他们表示基于几何方法的深度补全相比基于深度学习的算法更适合视觉 SLAM 场景，并且计算速度相比双目和深度学习补全都有较大提升。

基于以上考虑，本文尝试利用基于几何方法的深度补全算法 IPbasic[50] 对图像范围内的点云进行补全，用于后续的 SLAM 任务。IP basic 是一种快速的点云补全方法，可以将来自激光雷达的稀疏深度转为和像素对应的稠密深度，因此可以使单目视觉 SLAM 获得绝对尺度从而产生真实的位姿估计

由于靠近激光雷达附近的点的深度接近 0m，而无深度的像素点深度也为 0，为区分二者，首先对输入深度进行反转，方便在后续去除无效深度。统计数据集中最远处的深度约为 80m，若保留约 20m 的距离缓冲，则按 100m 为最大距离反

转深度，如式 (3-6) 所示。

$$D_{inverted} = 100.0 - D_{input} \tag{3-6}$$

然后将输入点云按其深度划分为三种，分别为近距离点，中距离点和远距离点。其对应的距离范围分别为 (0.1m,15m]，(15m-30m]，(30m-80m]。对三种点分别使用 3×3，5×5，7×7 的核进行膨胀运算。如对于中距离点，使用 5×5 的内核进行膨胀，形式如式 (3-7) 所示。

$$\begin{aligned} D \oplus C &= \left\{ z \mid (\hat{B})_z \cap D \neq \varnothing \right\} \\ C &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3-7}$$

式中 D 表示深度图，C 表示使用的内核。

深度图在膨胀后仍然存在较多未填充的小孔，首先使用完整的 5×5 核进行形态学中的闭运算进行填充，如式 (3-8) 所示。对于较大孔洞，使用 9×9 核进行膨胀，运算同式 (3-7)，仅改变所使用的内核 C。

$$\begin{aligned} D \bullet C &= (D \oplus C) \ominus C \\ C &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{aligned} \tag{3-8}$$

膨胀后需要对深度图进行滤波，去除由膨胀产生的大量噪声。可以使用中值滤波或双边滤波。文献 [52] 对深度补全后的点云进行高斯滤波，用产生的稠密点云做 3D 目标检测。本节选择了双边滤波，因为使用双边滤波可以保留点云的局部结构，这对于 SLAM 任务较为重要。

再将运算后的逆深度恢复为真实的深度，如式 (3-9) 所示。

$$D_{output} = 100.0 - D_{inverted} \tag{3-9}$$

图 3-3显示了深度补全的过程，其中图 3-3(b) 是由激光点云投影得到的稀疏深度，图 3-3(c) 是由深度补全获得的稠密深度图。

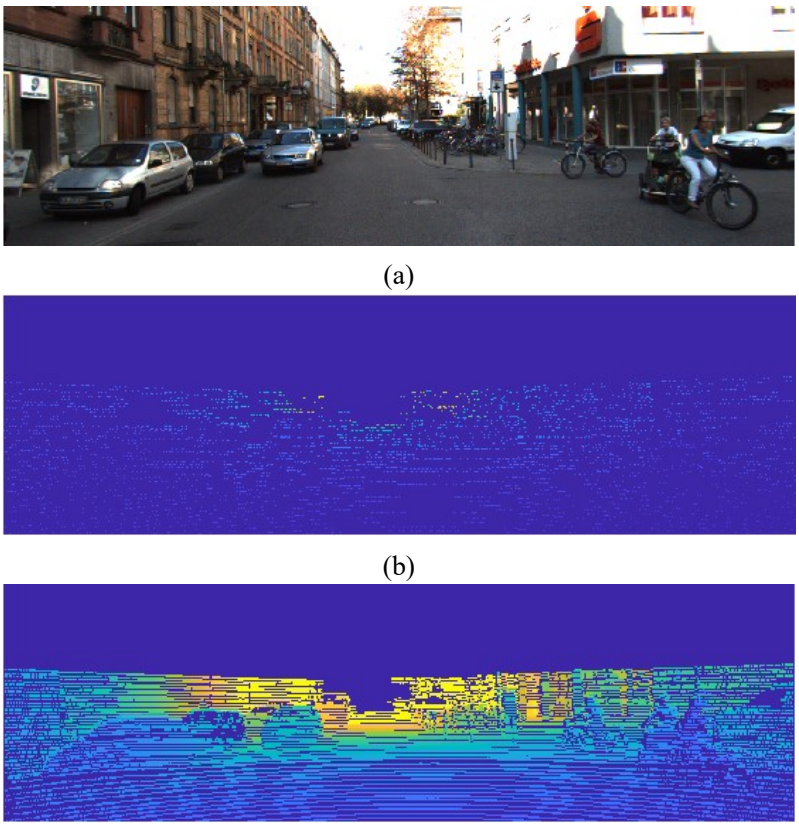


图 3-3 深度补全结果。(a) 输入图像；(b) 稀疏深度；(c) 补全后深度。

3.2.3 深度图逆投影

在获得补全后的稠密深度图时，可以将深度图逆投影至 3D 空间中，获得稠密的 3D 点云。如果稀疏深度已经经过补全，则稠密的点云对应的逆投影公式为：

$$\begin{aligned} x &= Z_c \cdot (u - u_0) \cdot /f_x \\ y &= Z_c \cdot (v - v_0) \cdot /f_y \\ z &= Z_c \end{aligned} \tag{3-10}$$

其中的参数见式 (3-4), 式 (3-5)。此时的 3D 点 x, y, z 位于相机坐标系下，若要将其转换到激光坐标系下，使用式 (3-3) 对相机坐标系下的点做变换即可。

3.3 视觉激光融合的里程计

3.3.1 稠密深度的视觉里程计

稠密深度和输入图像组成 RGB-D 格式的数据，可以在室外使用多种视觉 SLAM 算法。这里选择在 ORB SLAM2 [13] 的基础上进行设计。为了评估视觉激光融合算法的效果，因此不使用 ORB SLAM2 提供闭环检测和全局优化，仅利用帧间里程计和局部光束平差法 (Bundle Adjustment,BA) 作为视觉里程计输入。同时，激光 SLAM 也不做闭环检测，以消除闭环对结果带来的提升。

视觉里程计的流程大体遵循 ORB SLAM2 的 RGB-D 模式，但为方便融合与激光融合做了部分改动，去除了不必要的模块。再利用 BA 进行位姿调整。

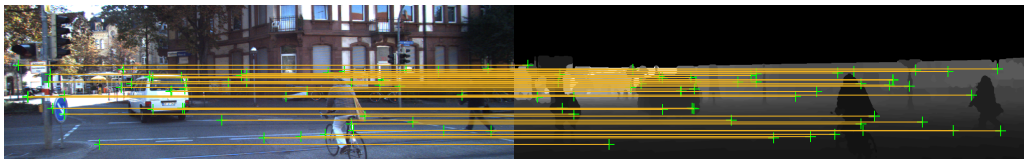


图 3-4 ORB 特征点匹配

视觉里程计按如下如下步骤进行：

(1) 提取当前图像的 ORB 特征，并与上一帧的特征进行匹配。

(2) 使用 PnP 算法 [53] 对相机位姿进行估计。

(3) 使用 BA 算法 [54] 对 PnP 的结果进行优化。

(4) 将局部地图点投影到当前图像以匹配更多特征，再次使用 BA 进行优化。

图 3-4显示了 RGBD 模式下图像与深度图成功匹配的特征点。为便于可视化，已降低特征点的数量。

3.3.2 激光里程计

激光里程计建立在 LOAM 算法 [14] 的基础上。LOAM 是一种激光里程计算法，不具有闭环检测和全局优化。本节以 LOAM 作为激光里程计，并添加局部建图模块，获取关键帧位姿。激光里程计仅利用相邻两帧估计自身位姿，随着时间推移不可避免地会出现累计误差，因此在大场景或复杂环境中，定位结果不准确。为了减少误差，可以通过在姿态之间添加约束关系来进一步校正姿态，利用全局信息优化轨迹。为保证 SLAM 在不发生闭环时依然可以对位姿进行优化，参考文献 [55] 提出的局部位姿图轨迹优化方法，使用关键帧位姿约束对普通帧进行矫正。

LOAM 按照式 (3-11) 计算曲率 [56]，并根据曲率 c 的大小将特征点分为尖锐边缘点，平滑的边缘点，尖锐的平面点和平滑的平面点。

$$c = \frac{1}{|\mathcal{S}| \cdot \left\| X^L_{(k,i)} \right\|} \left\| \sum_{j \in \mathcal{S}, j \neq i} \left(X^L_{(k,i)} - X^L_{(k,j)} \right) \right\|. \tag{3-11}$$

选择方法为，将雷达的每个线束分为六段，分别在每段中提取特征点。曲率最大的两个点为尖锐边缘点，曲率较大的二十个点为平滑的边缘点，曲率最小的四个点为平滑的平面点，剩余的点为尖锐的平面点。

由于检测算法依赖于每个点的近邻点来分类边缘点和平面点，以及识别遮挡区域边界上的不可靠点，因此不可在特征点检测之前进行降采样、去噪和去除地面等预处理步骤。但可以在提取特征点后进行处理。例如，由于尖锐平面点数量较多且不可靠，还可以对他们进行降采样样以加速运算。结果如图 3-5所示，紫色是原始 LOAM 提取的特征点，绿色是经降采样后的特征点。

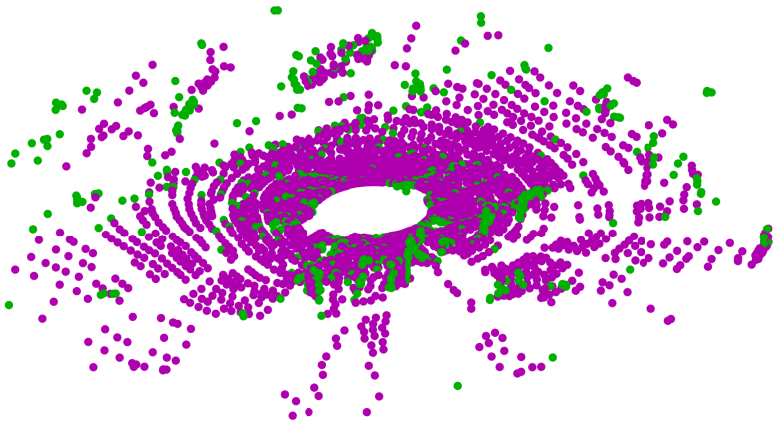


图 3-5 降采样后的 LOAM 特征点

对以上提取到的特征点进行点云配准，获得帧间位姿估计，并计算最近邻匹配后均方根误差。将配准后的均方根误差以信息矩阵的形式构建因子图中的帧间约束。

3.3.3 局部建图和关键帧约束

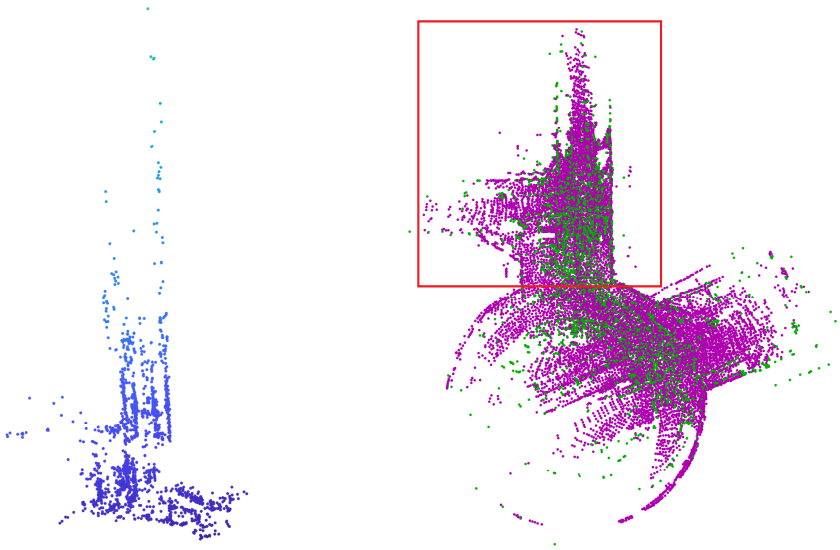
对于视觉帧，参考 ORB SLAM2 中的原方案，即当前帧如同时满足以下两个条件，则判断为关键帧：

- 当前帧距离上一关键帧至少 20 帧，或当前帧被跟踪的地图点少于 100 个
- 当前帧被跟踪的地图点少于被上一关键帧跟踪的点的 90%

获取关键帧后，对关键帧执行局部 BA 优化，从而获取关键帧位姿，进而更新局部地图。

对于激光帧，并未使用 LOAM 中的原始方案，而是重新设计了一种的关键帧选择和局部建图算法。这种方法更易于对多种传感器的多种量测进行统一融合，获得全局优化的位姿。

首先将第一帧定义为关键帧，然后把第一帧的特征加入特征地图。对随后的帧进行判断，若当前帧与上一关键帧的旋转角度大于 10° 或位移大于 2m，即认为是新的关键帧。利用当前帧的特征和特征地图进行匹配，获得激光关键帧位姿。再利用关键帧位姿将关键帧的特征加入特征地图中。随着地图增大，匹配速度也会随之降低。因此对提取当前帧附近局部特征地图用于匹配，而非使用完整的特征地图。



(a) (b)

图 3-6 用于局部建图的特征地图。(a)ORB 特征地图；(b)LOAM 特征地图。

图 3-6是用于局部建图和关键帧位姿获取的视觉和激光特征点地图。图 3-6 (b) 表示 LOAM 特征的地图，紫色表示面点，绿色表示边点。由于相机和激光雷达视角不同，导致建立的特征地图范围不同。图中已标注共同区域，如 LOAM 特征地图上红框 (图 3-6 (b)) 表示对应图 3-6 (a) 中视觉特征点的区域。

3.4 因子图优化

视觉激光里程计利用图像和稠密深度、稀疏点云进行帧间的视觉和激光里程计估计，并分别利用其建立的局部地图进行局部定位和优化。本节在后端优化中，接收前端各种约束，融合视觉和激光里程计的多种因子，建立因子图进行全局优化同时建立全局地图。所建立的因子图模型示意图如图 3-7所示，包括视觉里程计、激光里程计、视觉关键帧、激光关键帧四种因子，其中待优化变量为每时刻

自身位姿。

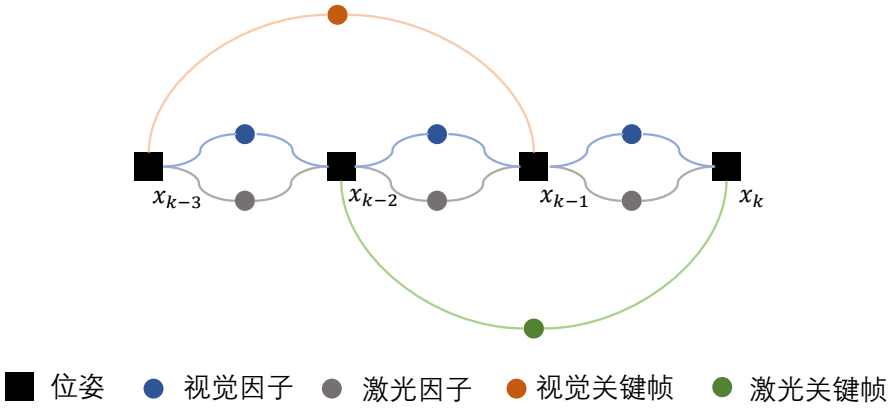


图 3-7 因子图优化视觉激光 SLAM 融合

各因子可统一写为：

$$\boldsymbol{e}\left(\boldsymbol{X}_i, \boldsymbol{X}_j, \Delta \boldsymbol{T}_{ij}\right)=\left(\left(\boldsymbol{X}_i\right)^{-1} \cdot \boldsymbol{X}_j\right)^{-1} \cdot \Delta \boldsymbol{T}_{ij} \tag{3-12}$$

对于视觉和激光的里程计因子，$i=j-1, \Delta \boldsymbol{T}_{ij}$ 表示里程计结果；对于视觉和激光的关键帧因子，$i=1, j=t$，t 为当前关键帧的时刻，$\Delta \boldsymbol{T}_{ij}$ 表示由局部建图获得的位姿，并转换到世界坐标系下，是相对与第一帧的位姿。

所建立的因子图模型：

$$\boldsymbol{X}^*=\arg \min _{\boldsymbol{X}}\left\{\sum_i^T\left\|\boldsymbol{e}_v\right\|_{\Sigma_v}^2+\sum_i^T\left\|\boldsymbol{e}_l\right\|_{\Sigma_l}^2+\sum_i^{K_v}\left\|\boldsymbol{e}_{vk}\right\|_{\Sigma_{vk}}^2+\sum_i^{K_l}\left\|\boldsymbol{e}_{vk}\right\|_{\Sigma_{lk}}^2+\right\} \tag{3-13}$$

式中，$\|\boldsymbol{e}\|_{\Sigma}^2$ 表示 $\boldsymbol{e} \cdot \Sigma^{-1} \cdot \boldsymbol{e}$；下标 v, l, vk, lk 分别表示误差来自视觉帧，激光帧，视觉关键帧和激光关键帧；e 的形式见式 (5-2)。

3.5 实验结果

在 KITTI 的里程计数据集[46] 上评估实验结果。该数据集由德国卡尔斯鲁厄理工学院和丰田科研实验室联合创建，提供了多个传感器的数据，包括激光雷达、相机和惯性测量单元（IMU）等。数据集包含了多个场景，如城市街道、高速公路和乡村道路，具有不同的光照、天气和道路状况。使用测试数据中从 00 到 10 的 10 个序列进行实验。

首先对于使用的指标进行介绍。绝对轨迹误差 (Absolute Trajectory Error, ATE) 是估计位姿和真值的差，反应了轨迹估计的精度和全局一致性，如式 (3-14) 所示。

$$\mathrm{ATE}_{\mathrm{trans}}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathrm{trans}\left(T_{\mathrm{gt},i}^{-1}T_{\mathrm{est},i}\right)\right\|_2^2} \tag{3-14}$$

相对位姿误差 (Relative Pose Error, RPE) 评价了固定时间间隔内相对两帧之间的定位误差，为简便起见，只考虑 RPE 的平移部分，RPE 的公式如式 (3-15) 所示。

$$\mathrm{RPE}_{\mathrm{trans}}=\sqrt{\frac{1}{N-\Delta t}\sum_{i=1}^{N-\Delta t}\left\|\mathrm{trans}\left(T_{\mathrm{gt},i}^{-1}T_{\mathrm{gt},i+\Delta t}\right)^{-1}\left(T_{\mathrm{esti},i}^{-1}T_{\mathrm{esti},i+\Delta t}\right)\right\|_2^2} \tag{3-15}$$

可以使用均值、中位数、最大值等方式评价 ATE 和 RPE。本文使用常用的均方根误差 (RMSE) 进行描述。

视觉激光融合 SLAM 的结果如表 3-1所示。在 01 和 02 序列上 LOAM 出现较大偏差，导致最终误差过大，因此去除了这两段特殊序列。和激光 SLAM 中的 LOAM 以及进行比较。相比激光 SLAM，本节提出的 SLAM 在多个序列上的 ATE 和 RPE 上有所降低。特别是在 z 轴方向上的垂直误差。说明视觉提供的观测降低了激光 SLAM 在高度上的估计误差。由于激光 SLAM 在高度上分辨率不足 (取决于雷达线束)，因此提出的算法改进了这一问题。

表 3-1 KITTI 里程计数据集上和激光 SLAM 的对比结果

序列	ATE(m)		RPE(m/f)		垂直误差 (m)	
	ALOAM	Ours	ALOAM	Ours	ALOAM	Ours
00	10.131	**6.190**	0.031	**0.030**	9.790	**7.690**
03	4.636	**1.360**	0.037	**0.020**	4.970	**3.110**
04	3.673	**1.320**	0.072	**0.034**	3.180	**2.570**
05	5.193	**4.470**	0.019	**0.018**	4.900	**3.230**
06	3.356	**3.480**	0.048	**0.024**	2.830	**2.460**
07	**0.920**	2.270	0.020	**0.018**	0.780	**0.650**
08	18.778	**5.670**	0.042	**0.040**	15.470	**14.420**
09	8.369	**2.950**	0.025	**0.024**	7.380	**5.960**
10	11.972	**2.660**	**0.022**	0.023	12.100	**6.550**
平均	7.448	**3.374**	0.035	**0.026**	6.822	**5.182**

还使用常用的 ATE 指标和其他 SLAM 算法进行了对比，实验结果如表 3-2所示，包括 LeGO-LOAM [57]，hdl-graph-slam [58]，ORB-SLAM2 [13]，DynaSLAM [2]，VISO2-LOAM [59]。其中，LeGO-LOAM，hdl-graph-slam 是纯激光的 SLAM 算法，ORB-SLAM2 是纯视觉 SLAM，DynaSLAM 是在动态环境下视觉 SLAM 算法，

表 3-2 和多种算法的 ATE(m) 指标对比的结果

序列	LeGO LOAM	hdl-graph-slam	ORB-SLAM2	DynaSLAM	VISO2-LOAM	Ours
00	**3.46**	28.10	5.33	7.55	13.30	6.19
01	21.70	76.08	-	-	25.39	19.06
02	6.84	2.67	21.28	26.29	17.12	7.31
03	4.01	1.53	1.51	1.81	-	**1.36**
04	2.84	16.42	1.62	0.97	0.79	**1.32**
05	2.68	**1.42**	4.85	4.60	2.69	4.47
06	3.21	3.07	12.34	14.74	**0.83**	3.48
07	3.27	1.33	2.26	2.36	**1.00**	2.27
08	**2.80**	59.88	46.68	40.28	4.70	5.67
09	7.85	46.14	6.62	3.32	**1.37**	2.95
10	8.16	31.93	8.80	6.78	**1.89**	2.66
平均	6.07	24.42	11.13	10.87	6.91	**5.16**

VISO2-LOAM 是一个视觉激光松耦合融合的算法。

从实验结果可以看出，本节提出的视觉激光融合方法的整体表现更稳定，平均结果更优。由于系统由 RGB-D 模式的视觉 SLAM 和激光 SLAM 两个子系统组成，并在后端利用因子图对两种传感器量测进行融合，因此结果更鲁棒、稳定。同时相比其他 SLAM 算法，轨迹误差得以降低，最终绝对轨迹误差为 5.16m。视觉 SLAM 通常拥有较准确的旋转估计，而平移估计的性能受限于深度估计，激光 SLAM 则对于平移的估计更准确，这为二者的结合提供了基础。

表 3-2中，如 ORB SLAM2 和 DynaSLAM 在 01 序列上经常定位失败，是因为可用于平移估计的特征点较少。本节提出的融合算法可在视觉特征提取过少的场景下利用激光雷达定位，依赖较准确的激光 SLAM 实现定位，提升了 SLAM 整体的稳定性。VISO2-LOAM 是视觉激光融合的 SLAM，但 VISO2-LOAM 使用双目相机和激光雷达，本节的方法避免引入双目标定，因此可以在室外获得更准确的深度估计。使用深度补全也可更快完成深度估计，并且运行时间仅为 10ms，约为立体匹配的一半。

需要注意的是，对于激光 SLAM 而言，本节的视觉激光融合方法只是在激光特征较少的场景补充视觉特征，因此对轨迹的精度提升有限。但在垂直方向上的定位准确性表现明显，以序列 09 为例进行分析，如图 3-8所示。其中蓝线表示本文提出的方法，红线表示真值，绿线和黑线分别是 LegoLOAM 和 LOAM 的结果。

由于 KITTI 数据集的真值保存在相机坐标系中，因此 y 轴表示垂直方向，即激光坐标系中的 z 轴方向。如在 x 和 z 方向轨迹几乎无变化，但在垂直方向上漂移大幅降低。因为图像在高度上的分辨率较激光雷达线数更高，可以对 z 轴平移

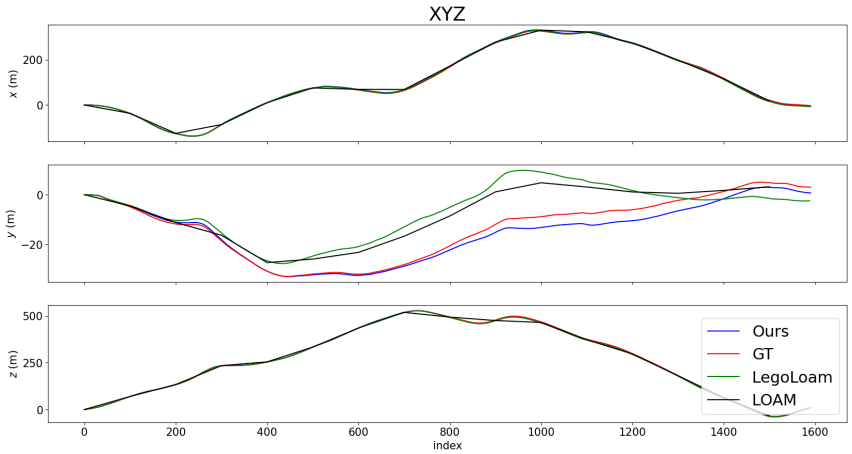


图 3-8 KITTI 里程计数据集序列 09 各方向的轨迹结果

估计产生稳定的观测。同时，视觉观测的加入也稳定降低了激光 SLAM 的相对位姿误差。

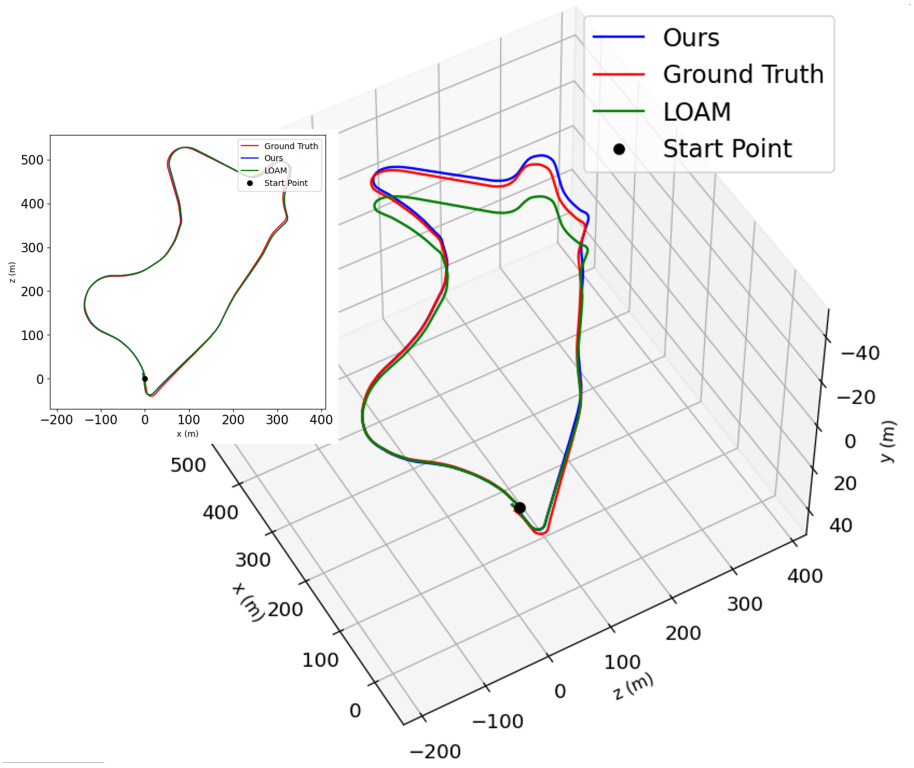
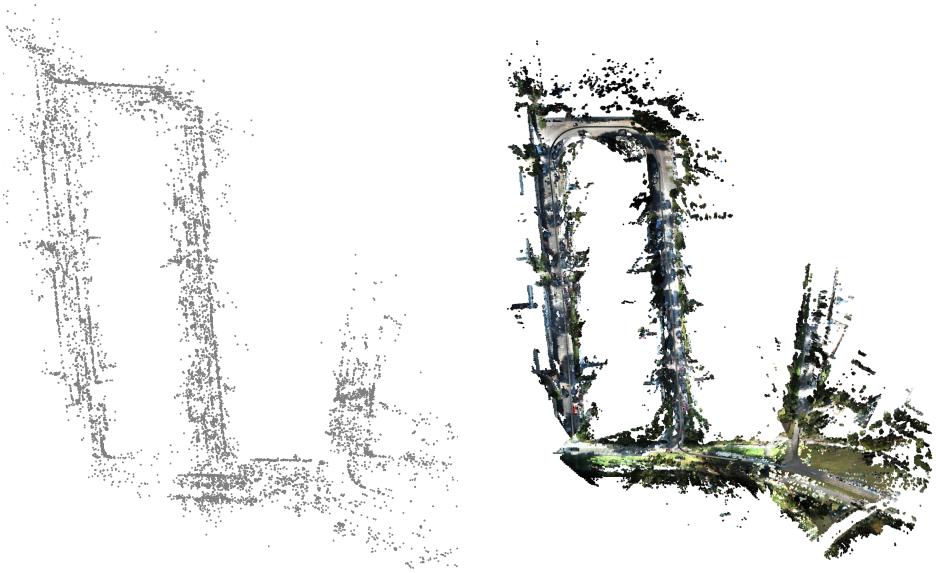


图 3-9 KITTI 里程计数据集序列 09 的整体轨迹结果

图 3-9显示了在序列 09 上的轨迹的整体效果结果，其中左上角小图是 xy 平面上的二维轨迹。可以看出，LOAM 在水平面上定位较准确，但在垂直方向上发生了较大偏差，也会导致建立的地图出现偏移的现象。而加入了视觉激光融合的 SLAM，利用视觉 SLAM 在垂直方向上估计较准确的特点，使得垂直漂移现象得

到缓解。由于闭环检测可以降低累计误差，构建全局一致地图，因此为避免闭环检测对整体结果的影响，整体实验中关闭了闭环模块，以体现视觉激光融合对轨迹带来的提升效果。

在得到自身轨迹后，可以根据特征点及其空间位置建立地图。在 ORB SLAM2 中，仅有特征点处的深度，因此只能建立稀疏特征点地图，这种稀疏地图可用于定位，但难以辅助后续任务。而本节提出的方法使用激光雷达提供的稀疏深度并进行深度补全，获得了稠密的深度图，故而可以重建出稠密的 RGB 地图，示意图如图 3-10所示。



(a) (b)

图 3-10 稠密建图结果。(a) 视觉 SLAM 建立的稀疏特征点地图；(b) 使用深度补全后建立的稠密 RGB 地图。

因此，本方案的核心贡献在于提供了一种快速通用的视觉激光融合的 SLAM 算法，仅需对激光雷达点云做投影和补全，就可以稳定降低垂直误差和相位位姿误差。但本方案的缺点在于未做传感器异常值的处理，如果其中一个 SLAM 子系统位姿偏差过大，则会导致因子图向错误的方向优化。可行的解决方案如对当前测量值做卡方检验，若卡方检验值大于统计显著性阈值，则将测量视为异常值抛弃。

3.6 本章小节

本节提出一种视觉激光融合 SLAM 的方法，主要通过深度补全实现。首先利用激光雷达获得稀疏深度并进行补全，视觉 SLAM 使用稠密深度进行视觉估计，

分别产生激光和视觉的普通帧和关键帧和约束。然后在后端使用建立因子图优化模型进行优化，结果显示定位精度相比激光 SLAM 有提升，提升主要反映在激光 SLAM 的垂直方向上。

第四章 联合自运动估计和 3D 运动目标检测的迭代动态配准算法

4.1 迭代动态配准算法流程

大多数 SLAM 算法假设环境处于静止状态，但实际中场景中物体的运动不可避免，如机器人在有人的室内环境中移动，拥挤的城市交通驾驶场景，运动的物体会造成位姿估计过程中特征的误匹配，导致定位的偏差，最终影响定位和建图的效果。由于目标检测技术只处理单张图像或雷达数据，物体的运动对目标检测影响较小，所以可以在单帧结果中直接获取目标的感知结果。但仅凭借单张检测结果又无法判断物体是否处于运动，因此本节提出一种动态环境下联合自运动估计方法以及三维运动目标检测方法（以下称为动态配准），通过自运动判断检测到的物体是否处于运动，并将动态物体移除，最终分割出运动物体并获得准确的自定位信息。提出方法的整体流程图如图 4-1所示。以下将介绍算法中的各部分。

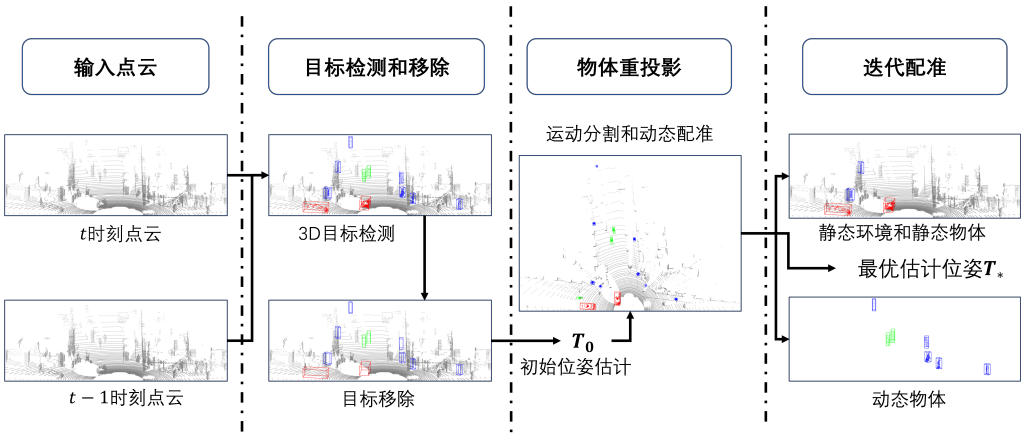


图 4-1 动态配准整体流程

4.2 目标检测和移除

4.2.1 3D 目标检测

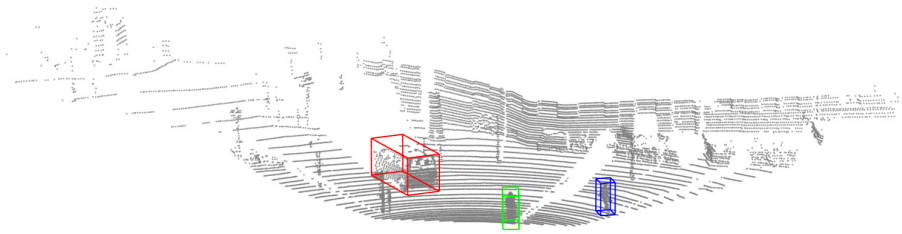
在动态场景下的激光 SLAM 研究中，已有一些方法尝试利用语义分割[19]进行目标的检测，但分割方法的固有局限限制了其应用。点云的语义分割和图像中的语义分割类似，仅能估计点或像素的语义类别，均无法产生物体实例的概念，进而无法判断运动物体。有些研究者为了解决这一问题，在语义分割的基础上使用聚类、最小包围框生成等后处理方法间接实现物体的概念，这些间接方法会导致物体检测效果较差[23,60]。

这些后处理方法衍生出了点云实例分割[61,62]和 3D 目标检测[63,64]的相关研究。点云实例分割可以获得完整的物体分割效果，单从分割效果上看，点云实例分割是更好的方法。但网络设计复杂，推理泛化效果差，所以对点云实例分割的研究仅局限在理论阶段，很难投入实际场景。

由于空间中点云是稀疏的，获取 3D 目标检测包围框内的点可实现点云实例分割同样的效果[62]，而计算效率却可以极大提升。考虑到实时性和检测精度，3D 目标检测是更好的方法。因此，本文使用 3D 目标检测算法 PointPillars[6] 作为检测器，可以在自动驾驶等场景实现实时的检测效果。若检测器在 t 时刻检测到 n 个物体，则将输出结果定义如下：

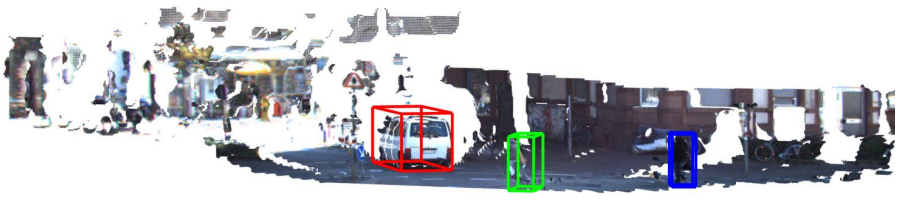
$$\begin{aligned} D_t &\triangleq \{d_0, d_1, \ldots, d_n\} \\ d_n &\triangleq \{x, y, z, l, w, h, \theta, label, score\} \end{aligned} \tag{4-1}$$

式中，x, y, z 表示包围框的中心点坐标；l, w, h 表示包围框的长宽高；θ 表示偏航角；*label* 表示物体的标签，包括行人，汽车和骑自行车的人三类；*score* 表示检测结果的置信度。



(a)

(b)



(c)

图 4-2 场景中物体的检测。(a) 融合前的 3D 目标检测；(b) 融合后的 3D 目标检测；(c) 补全后的 3D 目标检测。

将图像和点云进行融合并进行可视化，得到 PointPillars 的检测结果，如图 4-2所示。图 4-2 (a) 是纯激光的检测结果，图 4-2 (b) 中点云已用图像像素进行着色。

3D 目标检测的结果需做一定转换才能进行后续计算，以下在计算中仅取其位姿，即位置和姿态角。假设物体仅在水平面运动，不产生滚转和俯仰变化，仅有偏航角 θ。第 n 个物体 t 时刻的检测结果 D_t^n，其位姿形式 $\boldsymbol{D}_t^n$ 为：

$$\begin{aligned}\boldsymbol{D}_t^n &\triangleq \{x, y, z, 0, 0, \theta\} \\ &\triangleq \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{4-2}$$

对于 3D 目标检测模型，使用 MMDetection3D 提供的版本。因为希望产生相对精确而真实的检测的结果，因此不在跟踪数据集上做微调，直接使用原模型。模型在 KITTI 3D 检测数据集上进行了预训练，因此对跟踪数据集有一定的推理能力。KITTI 数据集上的 3D 对象检测 x, y, z 轴的检测范围分别为 $[(0, 70), (-40, 40), (-3, 1)]$。因此，根据 SLAM 和点云配准的场景，将 x, y, z 轴上将点云范围分别限制为 $[(0, 40), (-30, 30), (-3, 1)]$。随后滤除范围以外的检测。同时，选择检测后置信度高于 0.5 的结果，去掉一些低质量的检测结果。

4.2.2 3D 目标移除

三维目标检测仅给出了整个输入点云中物体的位置及大小，并未包括其中具体的点。使用以下方法可判断并筛选出位于每个包围框内的点：

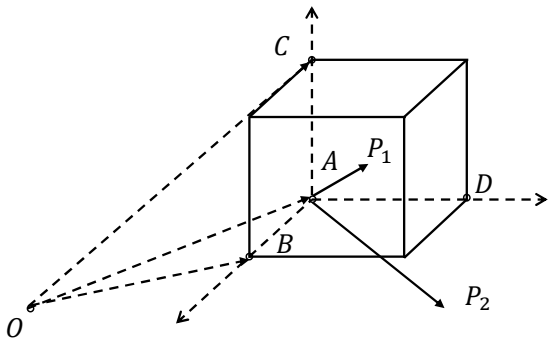
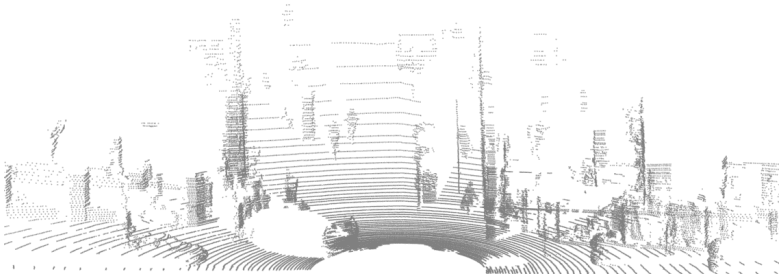


图 4-3 空间中任意一点和三维包围框的关系

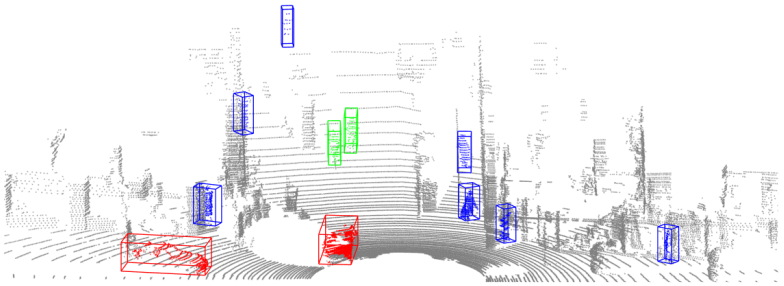
假设空间中一个包围框可以用三个向量 AB, AC, AD 进行表示，如图 4-3所示。

$ABCD$ 表示空间中的三维包围框，O 是坐标系原点，P_1 位于包围框内，P_2 位于包围框外。可以利用点积关系判断点是否位于包围框内：设 P 是三维空间中一点，若满足：

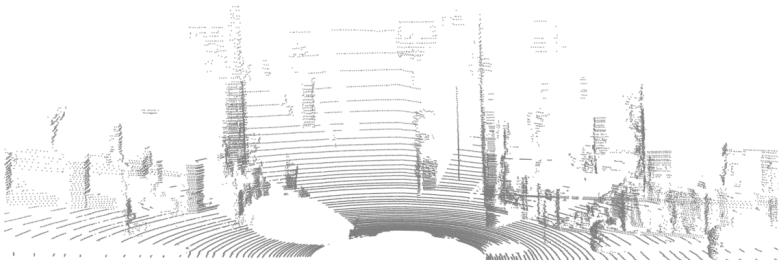
$$0 \leqslant \overrightarrow{AP} \cdot \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \leqslant \overrightarrow{AB} \cdot \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \tag{4-3}$$



(a)



(b)



(c)

图 4-4 场景中物体的检测和移除。(a) 物体移除前；(b) 检测到物体；(c) 物体移除后。

则认为 P 位于 AB 轴 $\pm 90°$ 之间，其长度不超过 $\|\overrightarrow{AB}\|$。同理可在其他轴上进

行判断，即可确定 P 的位置。具体如式 (4-4) 所示。

$$\begin{aligned} 0 \leqslant \overrightarrow{AP} \cdot \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \leqslant \overrightarrow{AB} \cdot \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} \\ 0 \leqslant \overrightarrow{AP} \cdot \frac{\overrightarrow{AD}}{\|\overrightarrow{AD}\|} \leqslant \overrightarrow{AD} \cdot \frac{\overrightarrow{AD}}{\|\overrightarrow{AD}\|} \\ 0 \leqslant \overrightarrow{AP} \cdot \frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} \leqslant \overrightarrow{AC} \cdot \frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} \end{aligned} \tag{4-4}$$

若点 P 满足式 (4-4)，则判断位于包围框内。无论点 P 是否是运动物体上的点，都暂时将其移除。如果在后续过程中物体被分割为静止，则会将其中的点再添加到静态环境中。图 4-4显示了物体检测和移除前和后的效果。

4.3 联合自运动估计与 3D 运动目标分割

4.3.1 自运动估计

在暂时移除全部物体之后，可以通过经典点云配准方法获得一个初始的帧间运动估计。以下对使用的两种配准方法，迭代最近点 (Iterative Closest Point, ICP) [65] 和正态分布变换 (Normal Distributions Transform, NDT) [66] 进行简要介绍。规定参考帧点云为 **M**，输入帧点云为 **N**。位姿估计结果为变换矩阵 **T**，通过旋转矩阵 **R** 和平移向量 $\boldsymbol{t}$ 定义为：

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{4-5}$$

ICP 算法计算两帧点云之间点的距离，并通过最小化此距离获得旋转矩阵 **R** 和平移向量 $\boldsymbol{t}$，如式 (4-6) 所示。

$$\boldsymbol{E}(\mathbf{R}, \boldsymbol{t}) = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{i,j} \left\| \boldsymbol{m}_i - (\mathbf{R}\boldsymbol{n}_j + \boldsymbol{t}) \right\|^2 \tag{4-6}$$

式中 $\boldsymbol{m}_i$ 和 $\boldsymbol{n}_j$ 表示点云 **M** 和 **N** 内的一点，$w_{i,j}$ 表示权重，如果 $\boldsymbol{m}_i$ 是到 $\boldsymbol{n}_j$ 的最近点，则 $w_{i,j}$ 为 1，否则为 0。

NDT 算法首先需将参考帧点云 **M** 划分为不同体素，再计算每个体素 i 内包含点云的均值 p_i 和协方差 Σ_i。将输入帧中的点 n_i 变换到通过式 (4-7) 变换到参考帧

下，再利用式 (4-8) 进行求解。

$$\boldsymbol{n}_i' = \mathbf{R}\boldsymbol{n}_i + \boldsymbol{t}' \tag{4-7}$$

$$\boldsymbol{E}(\mathbf{N}', \mathbf{T}) = -\sum_{i}^{N-1} \exp \frac{-(\boldsymbol{n}_i' - \boldsymbol{p}_i) \sum_i^{-1} (\boldsymbol{n}_i' - \boldsymbol{p}_i)}{2} \tag{4-8}$$

通过以上优化方法求出的位姿变换矩阵 $\mathbf{T}$，是通过将物体全部去除获得的，并未完全利用场景的静态信息，故将其定义为初始变换矩阵 $\mathbf{T}_0$。为统一起见，不区分具体的配准方法，将其规定为：

$$\boldsymbol{T}_0 = \text{ registration } (\boldsymbol{M}, \boldsymbol{N}) \tag{4-9}$$

4.3.2 物体重投影

物体里程计获取的是物体相对于自身起始点的真实运动，便于获取物体的轨迹，进行后续分析跟踪预测等步骤。而在实时定位过程中，需要对于相邻帧获取的点云数据进行配准，因此将物体投影至上一时刻是更简单有效的方式，以下称为物体重投影。

借鉴视觉 SLAM 中重投影误差的概念，利用初始变换矩阵 $\mathbf{T}_0$ 将 t 时刻的检测结果投影至上一时刻 $t-1$，得到两帧的检测结果：

$$\boldsymbol{d}_{t-1}' = \mathbf{T}_0 \cdot \boldsymbol{d}_t \tag{4-10}$$

重投影后的结果已经消除了自运动，仅包含其他物体自身的运动，如图 4-5所示。

4.3.3 数据关联

多目标跟踪可以实时估计环境中目标的状态，在自动驾驶感知中至关重要。通过时间约束，可以过滤掉目标检测的错误结果，并且对遮挡具有一定的鲁棒性。多目标跟踪可以获得不同类别的多个物体的运动轨迹，将有助于在动态环境下实现准确的行为决策和路径规划。

3D 目标检测的输出是包含物体尺寸、偏航和中心位置的包围框。之后对每个物体需保持检测结果的唯一性，这就要求将每时刻的检测结果准确地分配给对应的物体，此问题被称为多目标跟踪中的数据关联问题。此处需要对两个时刻的结果进行匹配，确定不同检测结果属于同一物体不同时刻的检测，才能在两帧点云中将其正确地去除。

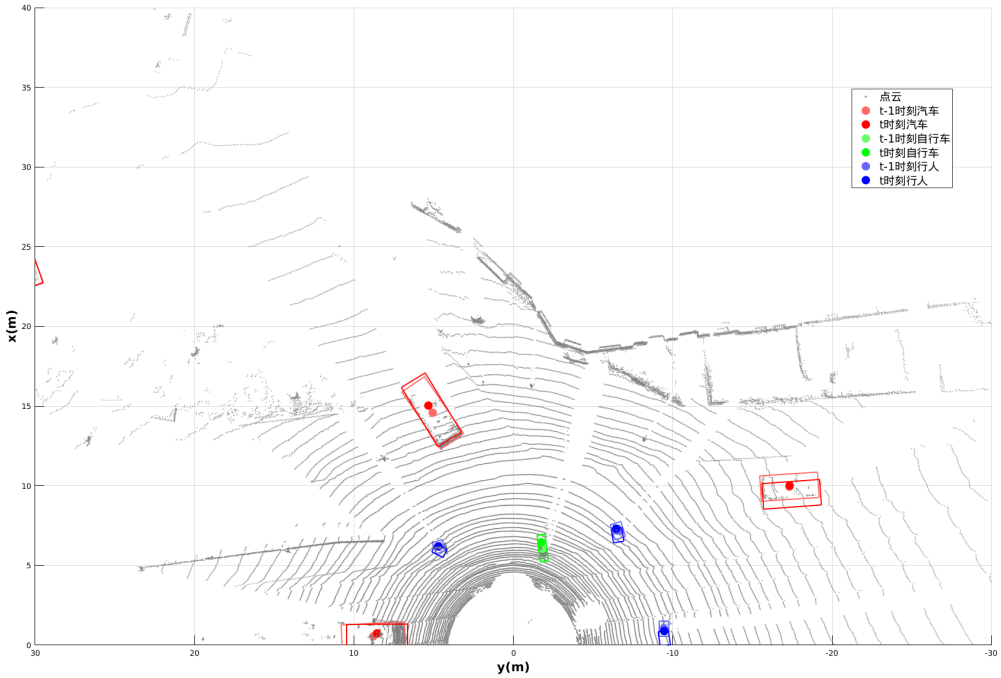


图 4-5 物体重投影和数据关联结果

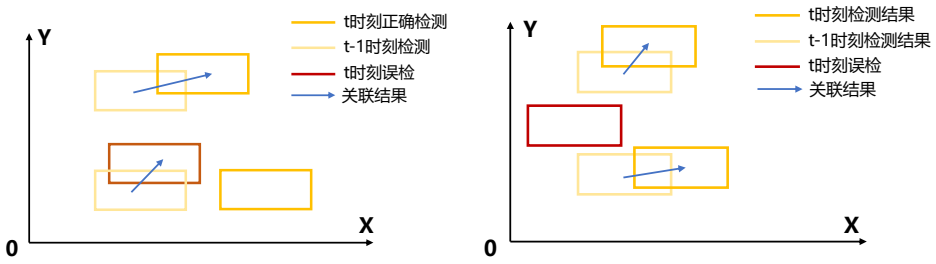
目前已有基于贝叶斯理论和深度学习的关联方法，但在这种场景下会极大提高算法的复杂度。由于系统仅需两帧中的对应物体，参考 SORT [67] 和 AB3DMOT [68] 两个计算机视觉领域经典的多目标跟踪方法，此处使用匈牙利匹配 [69] 处理关联问题。

对于多目标跟踪，由于物体运动和自运动是同时存在的，复杂运动的情况导致难以进行数据关联，各类文献提出的方法集中在如何确定距离度量，而动态配准场景下，自运动已被消除，关联的不确定性仅来自三维目标检测器，不同时刻的检测结果具有较大的重合度。基于这一特点，可以用投影前后的中心点距离为度量进行匹配和关联，无需计算 AB3DMOT 中提出繁琐的 3D IoU。直观上看，二者中心点距离越小，就有更大概率被认为是同一物体，如图 4-5所示。基于中心点距离的关联也可以消除由于自运动导致的错误关联，这一问题在多目标跟踪中往往被忽略。

数据关联的示意图见图 4-6, 除了正确匹配的物体外，还包含未匹配的结果，来自于检测器的误检或漏检。自运动的存在会导致进行错误的数据关联。

4.3.4 运动分割

当物体被正确关联后，就可以计算物体重投影的误差。此时物体的重投影误差就表示物体在两帧之间真实的运动，以此为阈值就能实现运动分割，即分割出



(a) (b)

图 4-6 场景中物体的检测和移除。(a) 消除自运动前的错误数据关联；(b) 消除自运动后的正确数据关联。

分别处于运动和静止的物体。经本节实验发现，以表 4-1中阈值进行分割，可以在大多数场景中的得到正确的分割结果。随后，静态物体与之前的静态环境相融合，至此，动态环境已被分割为动态物体和包括静态物体的静态环境。

表 4-1 不同类别物体在重投影后的分割阈值

类别	运动阈值 (m)
汽车	0.3
自行车	0.1
行人	0.05

4.4 迭代动态配准

4.4.1 动态配准

整个动态配准的过程为：在输入点云 M_{t-1}, M_t 中进行 3D 目标检测，获得检测结果 D_{t-1}, D_t。从输入点云 M_{t-1}, M_t 中移除所有检测结果 D_{t-1}, D_t 中的点云，获得移除所有物体后的点云 M_{t-1}^{RA}, M_t^{RA} 。利用移除所有物体后的点云 M_{t-1}^{RA}, M_t^{RA} 进行初始配准，获得初始估计位姿 T_t^0。利用初始估计位姿 T_t^0 将 t 时刻检测结果 D_t 重投影 $t-1$ 时刻，得到物体重投影后的结果 D_t^{t-1}。在重投影后的鸟瞰图上对 D_{t-1}, D_t^{t-1} 进行运动分割，大于运动阈值的物体为动态物体，小于运动阈值的为静态物体。分别得到 $t-1$ 和 t 时刻的动静态物体，分别为 $D_{t-1}^d, D_{t-1}^s, D_t^d, D_t^s$。将两时刻的静态物体 D_{t-1}^s, D_t^s 和静态环境 M_{t-1}^{RA}, M_t^{RA} 进行融合，得到仅移除动态物体的环境点云 M_{t-1}^{RD}, M_t^{RD}。利用静态环境点云 M_{t-1}^{RD}, M_t^{RD} 再次配准，得到最终位姿 T_t^{DR} 。算法流程如算法 4-1所示。

表中出现的上下标字母，DR 表示动态配准获得的结果 (*Dynamic Registration*)，

算法 4-1 动态配准

Input: 输入点云: M_{t-1}, M_t
Output: 最终估计位姿: T_t^{DR}
被分割出的动态物体: D_{t-1}^d, D_t^d
被分割出的静态物体: D_{t-1}^s, D_t^s

$(D_{t-1}, D_t) \leftarrow$ 3D 目标检测 (M_{t-1}, M_t)
$(M_{t-1}^{RA}, M_t^{RA}) \leftarrow$ 移除所有物体 $(M_{t-1}, D_{t-1}, M_t, D_t)$
$T_t^0 \leftarrow$ 初始点云配准 (M_{t-1}^{RA}, M_t^{RA})
$D_t^{t-1} \leftarrow$ 物体重投影 (T_t^0, D_{t-1}, D_t)
$(D_{t-1}^d, D_t^d, D_{t-1}^s, D_t^s) \leftarrow$ 运动分割 (D_{t-1}, D_t^{t-1})
$(M_{t-1}^{RD}, M_t^{RD}) \leftarrow$ 环境点云融合和静态 $(D_{t-1}^s, D_t^s, M_{t-1}^{RA}, M_t^{RA})$
$T_t^{DR} \leftarrow$ 点云配准 (M_{t-1}^{RD}, M_t^{RD})

RA 表示结果由移除全部物体获得 (*Remove All*)，*RD* 表示结果由移除动态物体获得 (*Remove Dynamic*)，*d* 表示物体处于运动，*s* 表示物体处于静止。在动态配准中，利用移除仅动态物体的点云进行配准，就认为是动态配准的最终输出。

4.4.2 迭代过程

大多数场景下，仅简单地移除全部检测到的物体，就可以提高定位精度。但即使是在动态环境中也包含静态物体。静止的物体符合 SLAM 的静态环境假设。这些静态物体可以提供用于定位的特征和信息，所以不应当简单全部移除。

当静态物体被分割后，可以融入提取的静态环境中，并更新静态环境。再次利用添加静态物体之后的环境点云进行点云配准，就可以获得更为准确的自运动估计。而新的自运动估计可以分割出更准确的结果，因此这个过程可以迭代进行，直到没有新的静态或动态物体产生。这时静态环境，动态物体和自运动三者被完全分割，系统利用其他建图、定位或目标跟踪方法对这三者进行后续处理。迭代动态配准流程如算法 4-2所示。

迭代动态配准是动态配准的迭代过程。即已经通过动态配准获得了物体检测结果，初始位姿，动态配准后的位姿等结果，在此基础上迭代计算得到的。点云配准使用的输入点云不同，则最终估计出的位姿也不同。因此每获得新的静态点云，便利用其进行一次点云配准，最终获得动态环境下稳定的位姿。

算法 4-2 迭代动态配准

Input: 输入点云: M_{t-1}^{RA}, M_t^{RA}
检测结果: D_{t-1}, D_t
初始配准位姿: T_t^0
动态配准位姿: T_t^{DR}
Output: 最终估计位姿: T_t^*
分割出的动态物体: D_{t-1}^{d*}, D_t^{d*}
分割出静态物体: D_{t-1}^{s*}, D_t^{s*}

```
${}^0D_t^{t-1} \leftarrow$ 物体重投影 $(T_t^0, D_{t-1}, D_t)$
$({}^0D_{t-1}^d,\ {}^0D_t^d,\ {}^0D_{t-1}^s,\ {}^0D_t^s) \leftarrow$ 运动分割 $(D_{t-1},\ {}^0D_t^{t-1})$
${}^*D_t^{t-1} \leftarrow$ 物体重投影 $(T_t^{DR}, D_{t-1}, D_t)$
$({}^*D_{t-1}^d,\ {}^*D_t^d,\ {}^*D_{t-1}^s,\ {}^*D_t^s) \leftarrow$ 运动分割 $(D_{t-1},{}^*D_t^{t-1})$
while ${}^*D_{t-1}^s \neq D_{t-1}^s$ and ${}^*D_t^s \neq D_t^s$ do
    $({}^*M_{t-1}^{RD},{}^*M_t^{RD}) \leftarrow$ 点云融合 $({}^*D_{t-1}^s,{}^*D_t^s, M_{t-1}^{RA}, M_t^{RA})$
    $T_t^* \leftarrow$ 点云配准 $({}^*M_{t-1}^{RD},{}^*M_t^{RD})$
    $D_{t-1}^s = {}^*D_{t-1}^s$ 和 $D_t^s = {}^*D_t^s$
    ${}^*D_t^{t-1} \leftarrow$ 物体重投影 $(T_t^*, D_{t-1}, D_t)$
    $({}^*D_{t-1}^d,\ {}^*D_t^d,\ {}^*D_{t-1}^s,\ {}^*D_t^s) \leftarrow$ 运动分割 $(D_{t-1},{}^*D_t^{t-1})$
end
```

4.5 实验结果

4.5.1 配准结果

相对位姿误差 (Relative Pose Error, RPE) 公式如式 (3-15) 所示，RPE 评价了相对两帧之间的定位误差，较为符合本节点云配准的适用场景。此处 RPE 同时考虑了旋转和平移部分，为 $\mathrm{RPE}_{\mathrm{full}}$，如式 (4-11) 所示。此外，为了更好地评估本节提出的动态配准算法的性能，本节采用 KITTI 跟踪数据集 [46] 进行实验，因为数据集同时提供自我和对象信息，方便进行调试。KITTI 跟踪数据集与里程计数据集的基本设置类似，但场景中具有多个运动物体，主要用于目标跟踪实验，但也可用于 SLAM 领域。

$$\mathrm{RPE}_{\mathrm{full}} = \sqrt{\frac{1}{N-\Delta t}\sum_{i=1}^{N-\Delta t} \left\| \left(T_{\mathrm{gt},i}^{-1} T_{\mathrm{gt},i+\Delta t}\right)^{-1} \left(T_{\mathrm{esti},i}^{-1} T_{\mathrm{esti},i+\Delta t}\right) - I_{4\times 4} \right\|_2^2} \tag{4-11}$$

实验结果如表 4-2所示。对比了动态配准和传统的点云配准方法。分别在 NDT 和 ICP 两种配准方法的基础上构建了动态配准，总体来看结果符合预期，即去除动态物体降低了帧间估计误差。表中的 RMA 是通过移除全部物体实现的，RMD 是通过仅移除动态物体实现的，即本文提出的动态配准方法。

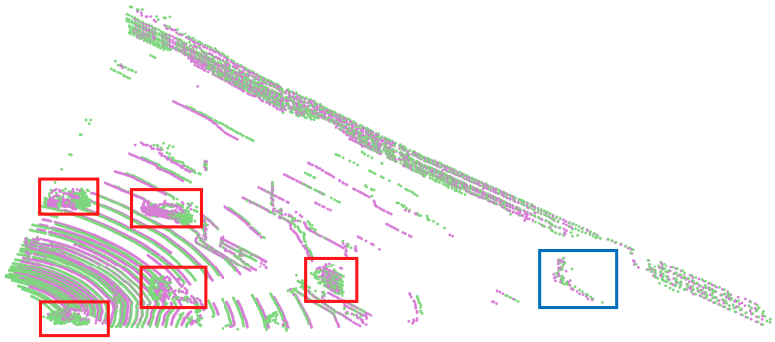
效果最好的方法是仅移除动态物体的动态配准方法，因为移除检测到的动态

物体是较好符合静态环境假设的。移除动态物体的效果比移除全部物体要好。因为简单移除所有物体会导致参与配准的点变少，而将静态物体添加至环境则会提供更多信息，使估计更稳定可靠，总体体现出较低的定位误差。

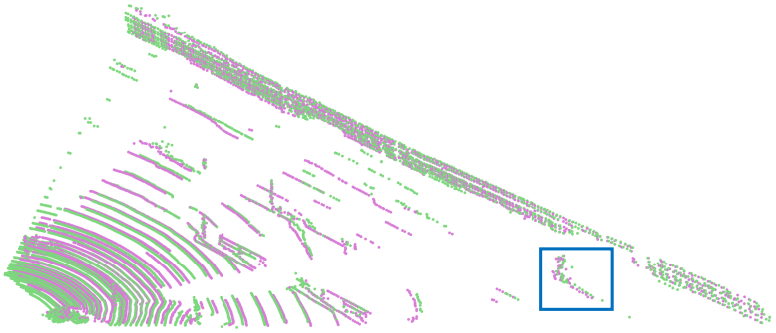
表 4-2 KITTI 跟踪数据集上动态配准算法 RPE 的对比结果。

seq	Registration		Dynamic Registration			
	NDT	ICP	NDT-RMA	ICP-RMA	NDT-RMD	ICP-RMD
0000	0.2583	0.3641	0.2646	0.3887	**0.2101**	**0.3444**
0001	0.3453	0.4550	0.3506	0.4793	**0.2689**	**0.4050**
0002	0.4560	0.5723	0.4718	0.5861	**0.4479**	**0.5696**
0003	0.7364	0.7952	0.7523	0.7869	**0.7151**	**0.7769**
0004	0.8349	0.7200	0.8509	0.7353	**0.8127**	**0.6874**
0005	0.8709	0.8537	0.8773	0.8643	**0.8545**	**0.8301**
0006	0.3247	0.4664	0.3190	0.4651	**0.2912**	**0.4496**
0007	0.3650	0.4428	0.3566	0.4601	**0.3195**	**0.4095**
0008	0.7415	0.7842	0.7361	0.7859	**0.7102**	**0.7589**
0009	0.4218	0.5042	0.4432	0.5230	**0.3979**	**0.4613**
0010	0.9602	0.8938	**0.9185**	0.8978	0.9407	**0.8667**
0011	0.3978	0.5209	0.4137	0.5629	**0.3722**	**0.5111**
0012	0.1161	**0.1432**	**0.0856**	0.1490	0.0955	0.1473
0013	0.3404	0.4283	0.3488	0.4282	**0.2861**	**0.3735**
0014	**0.1874**	**0.4625**	0.1939	0.5032	0.1876	0.4653
0015	0.3067	0.3625	0.3161	0.3705	**0.2894**	**0.3571**
0016	0.1917	**0.1136**	0.1341	0.1145	**0.1102**	0.1159
0017	0.1259	**0.1598**	0.1514	0.1658	**0.0801**	0.1617
0018	0.2791	0.3746	**0.2759**	0.3857	0.2788	**0.3074**
0019	0.2154	0.2977	0.2162	0.3054	**0.1954**	**0.2797**
0020	0.4879	0.7300	0.5098	0.7938	**0.4526**	**0.7138**
平均	0.4268	0.4974	0.4279	0.5120	**0.3960**	**0.4758**

图 4-7定性展示了点云配准的结果，绿色表示上一帧的点云，紫色表示当前帧的点云，利用估计出的帧间位姿将当前帧点云变换至上一帧并一同显示。在理想情况下，变换后的点云应尽可能重合。图 4-7 (a) 是未经处理的 NDT 配准算法，图 4-7 (b) 使用动态配准移除动态物体，保留静态物体并再次估计。图 4-7 (a) 红框是判断运动的物体，因为其不满足静态环境假设，因此框内点云出现较大移动，无法重合，而在图 4-7 (b) 中保留了静态环境和静态物体 (如蓝框所示)，因此点云配准效果较好。



(a)



(b)

图 4-7 点云配准结果对比。(a)NDT 配准的结果；(b) 动态配准的结果。

4.5.2 建图和里程计结果

使用连续帧间点云配准的结果，可以将动态配准算法简单扩展为激光里程计。由于其结构简单，仅用以可视化效果和定性评估。

图 4-8为在跟踪数据集中的 0013 序列上的估计的位姿和真值的对比结果。虚线表示真值，实线表示估计的位姿，其中蓝实线表示原始的 NDT 里程计，绿实线表示基于 NDT 的动态配准算法，对应表 4-2中的 NDT-RMD。相比 NDT 里程计，NDT-RMD 里程计拥有更小的轨迹偏移，更接近真值。

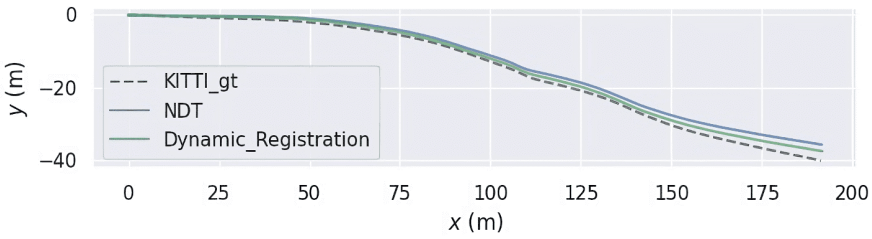
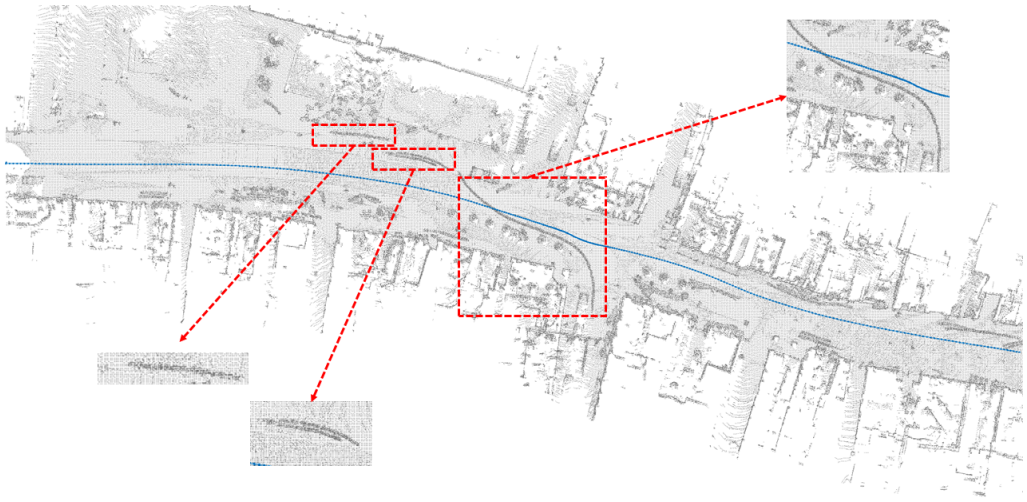


图 4-8 xy 平面上的轨迹对比图

图 4-9是在同一序列上的建图结果，其中图 4-9(a) 是 NDT 的位姿构建的点云地图，图 4-9(b) 是由 NDT-RMD 构建的点云地图。从建图结果上看，本文提出的

动态配准方法成功移除了动态物体。



(a)

(b)

图 4-9 移除动态物体前后的点云地图对比结果。(a)NDT 建图结果；(b) 移除动态物体后的 NDT 建图结果。

图 4-9中 (a) 和 (b) 中红框表示运动物体产生的轨迹，如果不对动态物体进行处理，则物体会留在地图中并产生鬼影，影响地图质量和后续的路径规划等任务。图 4-9(a) 中较为明显的轨影在图 4-9(b) 中已经被消除，但依然存在一些轨迹片段，如图 4-9(b) 中紫框所示。这是由于检测器的漏检造成的。因此，大部分运动目标可以被正确分割。

需要注意到运动目标检测的结果和三维目标检测，自运动估计和分割阈值相

关。当物体离自身较远时，就会导致误检和漏检。这些错误的检测结果可以通过数据关联进行一定程度的抑制。这是由于检测器性能导致无法每帧都精确地检测到物体。如果可以对检测的动态物体进行跟踪与预测，就可以在发生漏检时依然估计出物体的位置并移除相应的点云。本文将在第五章使用多目标跟踪对检测物体进行处理，增强4.3.3节使用的匈牙利匹配算法。

4.6 本章小节

本文提出了一种针对三维点云的动态配准方法。在点云配准方法的基础上引入三维目标检测技术，实现了动态环境下同时进行自运动估计和三维运动目标检测，并在公开数据集（KITTI 跟踪数据集）上进行实验验证。实验结果表明，所提方法有效降低了帧间的车辆定位误差。定性结果显示仍有部分运动物体轨迹未被删除，同时所提出方法仅在两帧点云间，本文在第五章将提出一个集成动态配准的完整 SLAM，同时利用多目标跟踪技术弥补仅 3D 目标检测的不足。

第五章 紧耦合的同时定位建图与多目标跟踪算法

5.1 SLAMMOT 算法流程

本节将第四章的算法推广至完整的 SLAM，将其定义为同时定位建图和多目标跟踪 (SLAMMOT)，可以实现同时定位建图和对多个目标进行跟踪。第四章提出的动态配准算法已经可以通过任意点云配准算法在两帧之间估计出准确的自运动，并分割出静态和动态物体，但动态配准仅处理相邻两帧之间的位姿估计，没有考虑到整个点云序列。例如，在连续帧的点云中，可以通过卡尔曼滤波等算法估计物体的速度，从而更精确地分割物体，或构建关键帧约束，利用局部地图进行定位等。所提出的 SLAMMOT 算法流程如图 5-1所示：

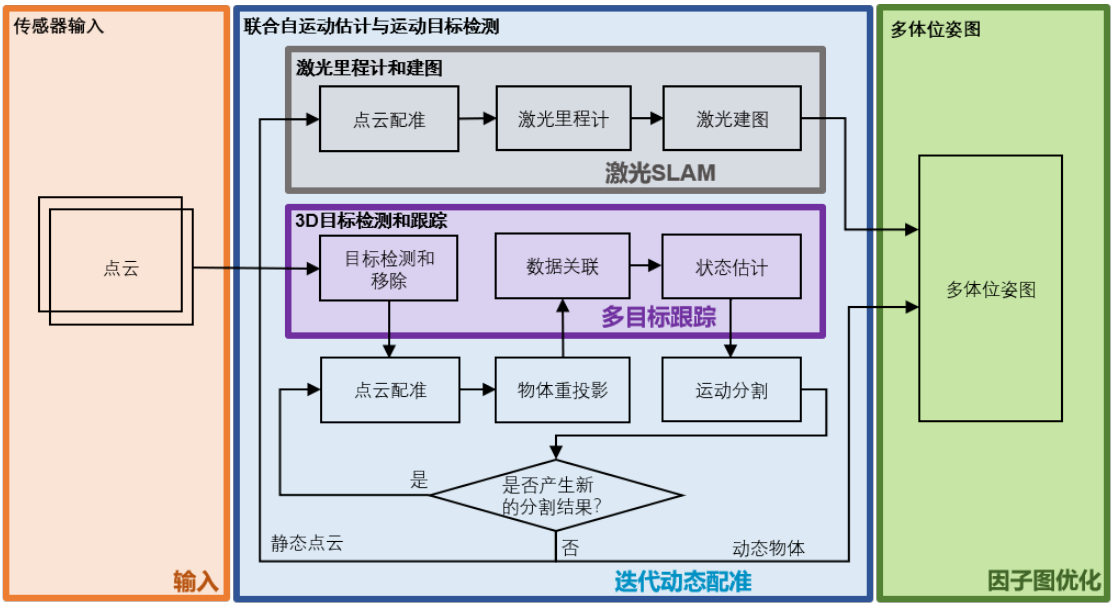


图 5-1 SLAMMOT 算法流程

5.2 激光 SLAM 算法及误差因子构建

本节在4.3.1节所使用的点云配准方法的基础上, 选择其中的 NDT 构建完整的激光 SLAM 算法，由于实际中闭环场景并不总是存在，导致累计误差无法消除，因此本节构建不依赖闭环检测的的激光 SLAM 后端，即通过关键帧和场景中的动静态物体提供约束。

5.2.1 激光里程计

在进行点云配准前，增加移除地面模块。由于激光雷达扫描地面会产生大量冗余的点云，变化较小，导致无法产生有效估计，因此可以在点云配准前移除地面。这里使用文献 [70] 提出的一种快速地面分割方法。首先将原始点云转到距离图像下，计算 α 角，如式 (5-1) 所示。

$$\begin{aligned} \alpha &= \operatorname{atan2}(\|BC\|, \|AC\|) = \operatorname{atan2}(\Delta z, \Delta x) \\ \Delta z &= |R_{r-1,c} \sin \xi_a - R_{r,c} \sin \xi_b| \\ \Delta x &= |R_{r-1,c} \cos \xi_a - R_{r,c} \cos \xi_b| \end{aligned} \tag{5-1}$$

式中，$R_{r,c}$ 表示位于距离图上 r 行 c 列的点，ξ_a 和 ξ_b 表示 $r-1$ 行和 r 行对应的垂直角，如图 5-2所示。

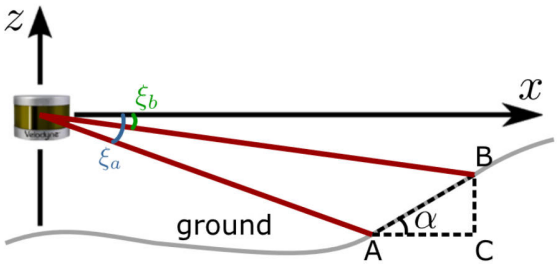


图 5-2 地面分割参数示意图 [70]

如果距离图上每一列的最后一行对应的 α 小于 45°，则假定为地面，如果其邻域内的 α 变化小于 5°，则标记为地面。在移除地面后，按照第四章所述的基于 NDT 的动态配准方法进行帧间运动估计，获得激光里程计。

以 $\boldsymbol{X}_t$ 表示当前自身的位姿，用 $\Delta \boldsymbol{T}_t$ 表示激光里程计在 $t-1$ 时刻和 t 时刻之间的相对位姿，则激光里程计的误差可表述为：

$$\boldsymbol{e}_{ego}\left(\boldsymbol{X}_{t-1}, \boldsymbol{X}_t, \Delta \boldsymbol{T}_t\right) = \left(\left(\boldsymbol{X}_{t-1}\right)^{-1} \cdot \boldsymbol{X}_t\right)^{-1} \cdot \Delta \boldsymbol{T}_t \tag{5-2}$$

5.2.2 局部建图和关键帧

由于本节构建的激光 SLAM 希望在不完全依赖闭环的情况下实现位姿的优化，因此根据 [55] 提出的思路，利用关键帧点云和局部地图进行匹配，获得关键帧位姿并转化为约束加入因子图中。

对于关键帧的选取，依旧按照3.3.3节中激光 SLAM 关键帧选择的方法进行设计。不同的是，获取关键帧后，利用关键帧和点云地图进行 NDT 配准，获得关键

帧位姿，并利用关键帧更新点云地图。

对于点云地图，相应构建 NDT 地图。由于前端使用基于 NDT 的动态配准，所以这种 NDT 地图也适合进行关键帧位姿匹配。NDT 地图将点云转为一组体素框，用三维正态分布表示体素内的分布。在关键帧和地图匹配时，首先是根据当前运动状态，从全局地图中分割出当前关键帧附近的局部子地图，再利用关键帧和地图进行匹配。

最终将获取的位姿统一转换到第一帧下，得到当前关键帧相对于世界坐标系的位姿，并构建关键帧约束的误差，以便后续加入因子图进行优化。若通过关键帧匹配获得的位姿为 $\Delta \boldsymbol{T}_t^{loc}$，则激光关键帧约束因子为：

$$\boldsymbol{e}_{loc}\left(\boldsymbol{X}_1, \boldsymbol{X}_t, \Delta \boldsymbol{T}_t^{loc}\right) = \left(\left(\boldsymbol{X}_1\right)^{-1} \cdot \boldsymbol{X}_t\right)^{-1} \cdot \Delta \boldsymbol{T}_t^{loc} \tag{5-3}$$

5.3 多目标跟踪算法设计

将数据关联由帧间的匈牙利匹配扩展到全局的 JPDA 数据关联算法，处理复杂环境下杂波干扰的数据关联和轨迹管理问题。相比4.3.3中使用的匈牙利匹配，增加了对目标的状态估计，用多种模型对复杂环境下的机动目标进行建模。具体来说，将匈牙利匹配换为联合概率数据关联 JPDA 滤波器，使用交互多模型 IMM 在不同模型间切换，使用 UKF 对非线性运动模型进行估计。三者形成一个 JPDA-IMM-UKF 组合滤波器[23,37]。

本文已在2.2节对基本的多目标跟踪方法进行介绍，因此本节只涉及在 JPDA-IMM-UKF 滤波器的基础上进行设计。定义匀速模型和匀转弯率和速度模型两个运动模型，使用 IMM 滤波器进行估计。由于不同运动模型具有不同的维度的状态，如果直接用零对齐维数，会导致不同运动模型的过程噪声方差奇异，进而导致滤波估计协方差奇异，使得矩阵分解失败、无法产生采样点。因此在针对不同模型的滤波器，只对真实包含的状态进行计算，其余状态保持不变。最后再组合状态进行输出。

用随机离散模型建模被跟踪目标的运动状态，定义为:

$$\boldsymbol{x} = [x, v_x, y, v_y, z, v_z, \theta, \omega]^{\top} \tag{5-4}$$

系统的状态方程和观测方程为：

$$\begin{aligned} \boldsymbol{x}_k &= f_k\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_k \\ \boldsymbol{z}_k &= h\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{v}_k \end{aligned} \tag{5-5}$$

$\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 分别是相互独立且为零均值的高斯白噪声，协方差为 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$。

假设物体处于恒速运动，则可以简单取状态为 $\boldsymbol{x}_{CV} = [x, v_x, y, v_y, z, v_z]$ 进行计算。在完成估计后，将更新后的位置和由检测获得的偏航角进行组合，获得当前物体的状态。此时物体处于恒速模型下，状态转移方程为：

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \begin{bmatrix} v_{x,k-1} \\ 0 \\ v_{y,k-1} \\ 0 \\ v_{z,k-1} \\ 0 \end{bmatrix} + \boldsymbol{w}_k \tag{5-6}$$

恒速模型的噪声 $\boldsymbol{w}_k$ 来自直线加速度 a_x, a_y, a_z，假设 $a_x \sim \mathcal{N}(0, \sigma_x^2), a_y \sim \mathcal{N}(0, \sigma_y^2), a_z \sim \mathcal{N}(0, \sigma_z^2)$，则预测过程噪声 $\boldsymbol{w}_k$ 和其协方差矩阵 $\boldsymbol{Q}$ 为：

$$\boldsymbol{w}_k = \begin{bmatrix} \frac{1}{2}T^2 a_x \\ T a_x \\ \frac{1}{2}T^2 a_y \\ T a_y \\ \frac{1}{2}T^2 a_z \\ T a_z \end{bmatrix} = \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{1}{2}T^2 \\ 0 & 0 & T \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \boldsymbol{G}\boldsymbol{u} \tag{5-7}$$

$$\boldsymbol{Q} = \mathrm{cov}(\boldsymbol{w}) = E\left(\boldsymbol{w}\boldsymbol{w}^\top\right) = \boldsymbol{G}E\left(\boldsymbol{u}\boldsymbol{u}^\top\right)\boldsymbol{G}^\top = \boldsymbol{G}\begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}\boldsymbol{G}^\top \tag{5-8}$$

恒转弯率和速度模型中通常 $\boldsymbol{x}_{CTRV} = [x, y, v, \theta, \omega]$，描述物体在 $x-y$ 平面中的二维运动。其中，v 为物体的速度，θ 为偏航角，是追踪的目标车辆在当前车辆坐标系下与 x 轴的夹角，逆时针方向为正，取值范围是 $[0, 2\pi)$，ω 是偏航角速度。而在三维空间中时，需扩展 $\boldsymbol{x}_{CTRV} = [x, v_x, y, v_y, z, v_z, \theta, \omega]$, 其状态转移函数如下:

$$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + \begin{bmatrix} \frac{v}{\omega}\left[\sin(\theta+\omega T)-\sin(\theta)\right] \\ v(\cos(\theta+\omega T)-\cos(\theta)) \\ \frac{v}{\omega}\left[\cos(\theta)-\cos(\theta+\omega T)\right] \\ v(\sin(\theta+\omega T)-\sin(\theta)) \\ v_{z,k-1} \\ 0 \\ \omega T \\ 0 \end{bmatrix} + \boldsymbol{w}_k \\ &= \boldsymbol{x}_{k-1} + \begin{bmatrix} \frac{1}{\omega}(v_{x,k-1}\sin(\omega T)+v_{y,k-1}\cos(\omega T)-v_{y,k-1}) \\ v_{x,k-1}\cos(\omega T)-v_{y,k-1}\sin(\omega T)-v_{x,k-1} \\ \frac{1}{\omega}(v_{y,k-1}\sin(\omega T)-v_{x,k-1}\cos(\omega T)+v_{x,k-1}) \\ v_{x,k-1}\sin(\omega T)+v_{y,k-1}\cos(\omega T)-v_{y,k-1} \\ v_{z,k-1} \\ 0 \\ \omega T \\ 0 \end{bmatrix} + \boldsymbol{w}_k \end{aligned} \tag{5-9}$$

式 (5-9) 仅描述了偏航角速度 ω 不为 0 时 CTRV 模型的状态转移方程，而对于 $\omega = 0$ 的情况，则应对应匀速模型 CV 的状态转移方程，如式 (5-6) 所示。

在 CTRV 模型中噪声的引入主要来源于两处：直线加速度噪声 a_x, a_y, a_z 和偏航角加速度噪声 a_ω，假定 $a_x \sim \mathcal{N}(0,\sigma_a^2), a_y \sim \mathcal{N}(0,\sigma_y^2), a_z \sim \mathcal{N}(0,\sigma_z^2), a_\omega \sim \mathcal{N}(0,\sigma_\omega^2)$，则预测过程噪声 $\boldsymbol{w}_k$ 和其协方差矩阵 $\boldsymbol{Q}$ 为：

$$\boldsymbol{w}_k = \begin{bmatrix} \frac{1}{2}T^2a_x \\ Ta_x \\ \frac{1}{2}T^2a_y \\ Ta_y \\ \frac{1}{2}T^2a_z \\ Ta_z \\ \frac{1}{2}T^2a_\omega \\ Ta_\omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 & 0 \\ T & 0 & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & \frac{1}{2}T^2 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & \frac{1}{2}T^2 \\ 0 & 0 & 0 & T \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \\ a_\omega \end{bmatrix} = \boldsymbol{G}\boldsymbol{u} \tag{5-10}$$

$$\boldsymbol{Q}=\operatorname{cov}(\boldsymbol{w})=E\left(\boldsymbol{w}\boldsymbol{w}^{\top}\right)=\boldsymbol{G}E\left(\boldsymbol{u}\boldsymbol{u}^{\top}\right)\boldsymbol{G}^{\top}=\boldsymbol{G}\begin{bmatrix}\sigma_x^2 & 0 & 0 & 0\\ 0 & \sigma_y^2 & 0 & 0\\ 0 & 0 & \sigma_z^2 & 0\\ 0 & 0 & 0 & \sigma_\omega^2\end{bmatrix}\boldsymbol{G}^{\top} \tag{5-11}$$

系统的观测模型为：

$$\boldsymbol{z}_{k-1}=\boldsymbol{H}\boldsymbol{x}_{k-1}+\boldsymbol{v}_k=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\end{bmatrix}\cdot\begin{bmatrix}x_{k-1}\\v_{x,k-1}\\y_{k-1}\\v_{y,k-1}\\z_{k-1}\\v_{z,k-1}\\\theta\\\omega\end{bmatrix}+\boldsymbol{v}_k \tag{5-12}$$

观测噪声 $\boldsymbol{v}_k$ 是高斯分布，其协方差矩阵为 $\boldsymbol{R}$。

5.4 基于迭代动态配准的松耦合多体里程计

在多目标跟踪过程中，为了方便计算，需要利用自运动将检测和跟踪的输入和结果转换到统一世界坐标系下算，但在建立物体间的约束时，还需将检测和跟踪结果转换到当前位姿下。对于包含了物体和自身位姿的里程计，以下称为多体里程计。

在世界坐标系下的 t 时刻，以 $\boldsymbol{X}_t$ 表示自身的位姿，$\boldsymbol{O}_t^n$ 表示场景中第 n 个物体的位姿。$\Delta\boldsymbol{T}_t$ 表示激光里程计在 $t-1$ 时刻和 t 时刻之间的相对位姿，$\boldsymbol{T}_t$ 表示激光里程计累积得到的世界坐标系下的绝对位姿。$\boldsymbol{D}_t$ 表示机器人在 $\boldsymbol{X}_t$ 处由 3D 目标检测获得的第 n 个目标相对于自身的位姿，$\Delta\boldsymbol{S}_t^n$ 表示在世界坐标系下由 3D 目标跟踪估计得到的 $t-1$ 时刻和 t 时刻之间目标的位姿变化。

同一个物体在 t 和 $t-1$ 时刻的检测结果为 $\boldsymbol{D}_t^n$ 和 $\boldsymbol{D}_{t-1}^n$(检测结果的转换见式 (4-2))，两帧间的相对位姿 $\Delta\boldsymbol{T}_t$ 由激光里程计估计获得，即第四章中介绍的迭代动态配准算法。

首先利用帧间运动 $\Delta\boldsymbol{T}_t$ 累乘得到当前的位姿 $\boldsymbol{T}_t$，并将 3D 检测结果转换到全

局坐标系下，则被跟踪后的目标位姿为：

$$\boldsymbol{S}_t^n = UKF(\boldsymbol{T}_t \cdot \boldsymbol{D}_t^n) \tag{5-13}$$

则第 n 个物体物体在两帧间的运动为：

$$\Delta\boldsymbol{S}_t^n = (\boldsymbol{S}_{t-1}^n)^{-1} \cdot \boldsymbol{S}_t^n \tag{5-14}$$

$\Delta\boldsymbol{D}_t^n$ 表示了空间中三维物体的真实运动，这主要是通过 3D 目标检测实现的。3D 目标检测可以直接在输入点云中检测到物体在当前坐标系下的位置、姿态和大小，同时可以建立起物体坐标系，由此便可以通过坐标系转换得到物体运动。在 SLAMMOT 的后端，物体里程计将和自身里程计一同进行优化。

此时可得到一个松耦合的 SLAMMOT 算法，SLAM 部分是基于 NDT 的动态配准算法，MOT 部分是基于 JPDA-IMM-UKF 的多目标跟踪算法，二者通过动态配准连接，见图 5-1中灰色和紫色部分。大多情况下此算法可实现动态环境下的定位，但这种方法没有利用动态物体的信息。通过 MOT 估计出物体的速度后，将静态物体加入配准中，动态物体移除 SLAM 流程，由 MOT 单独跟踪。本质上仍属于2.1.2中所提到的 SLAM with DATMO [4]。在下一节中，将建立多体位姿图对自运动、静态物体位置、动态物体状态一同优化，实现利用动态物体定位。

目标跟踪误差是目标位姿之间的约束，为：

$$\boldsymbol{e}_{obj}\left(\boldsymbol{O}_{t-1}^n, \boldsymbol{O}_t^n, \Delta\boldsymbol{S}_t^n\right) = \left((\boldsymbol{O}_{t-1}^n)^{-1} \cdot \boldsymbol{O}_t^n\right)^{-1} \cdot \Delta\boldsymbol{S}_t^n \tag{5-15}$$

3D 目标检测误差是自身位姿和目标位姿之间的约束，计算为：

$$\boldsymbol{e}_{det}\left(\boldsymbol{X}_t^n, \boldsymbol{O}_t, \boldsymbol{D}_t\right) = \left((\boldsymbol{X}_{t-1}^n)^{-1} \cdot \boldsymbol{O}_t^n\right)^{-1} \cdot \boldsymbol{D}_t^n \tag{5-16}$$

5.5 多体位姿图的紧耦合因子图优化

多体里程计是松耦合的 SLAM 和 MOT，本节将多体里程计转为多体位姿图，并进行紧耦合的因子图优化。SLAM 和 MOT 通过动态配准连接起来，实现可以同时对自身运动、环境地图、目标的运动进行估计。利用因子图处理多目标跟踪问题，德国开姆尼茨工业大学 [9] 率先展开了研究，主要利用因子图修改节点之间边约束的方法解决多目标跟踪中不确定数据关联。考虑到对目标不确定的数据关联可能对 SLAM 系统带来负面影响，因此本文仅利用因子图实现目标状态估计，数据关联通过单独的方法实现。

对于多目标跟踪的数据关联，选择基于匈牙利匹配的全局最近邻算法。动态

配准中的数据关联仅在两帧之间考虑匈牙利匹配，未考虑到整个序列的跟踪连续性和轨迹管理问题。因此本节在匈牙利匹配的基础上使用全局最近邻算法对检测结果进行数据关联。

联合概率数据关联 JPDA 由于软关联的形式，导致在因子图中建模较困难，需要考虑轨迹和目标之间的概率关联，因此不适宜使用。将在后期工作中解决这一问题。但在 SLAM 与 MOT 松耦合的系统中，由于 JPDA 在复杂环境下对多目标跟踪的处理优于传统的 GNN，因此仍建议使用 JPDA-IMM-UKF 组合滤波器与 SLAM 进行结合，如5.4节所示方案。

对于未匹配的轨迹，不与当前的检测结果建立约束。但为防止漏检的存在，将轨迹保留一段时间，若连续五帧均未匹配，则将其从跟踪中删除。存在新的轨迹，即出现新目标。对目标进行初始化。但为防止误检的存在，若连续三帧中的两帧都成功匹配，则将其加入跟踪中。若既无未匹配得轨迹，也无未分配的检测，则根据匈牙利匹配结果建立当前检测和被跟踪轨迹的约束。

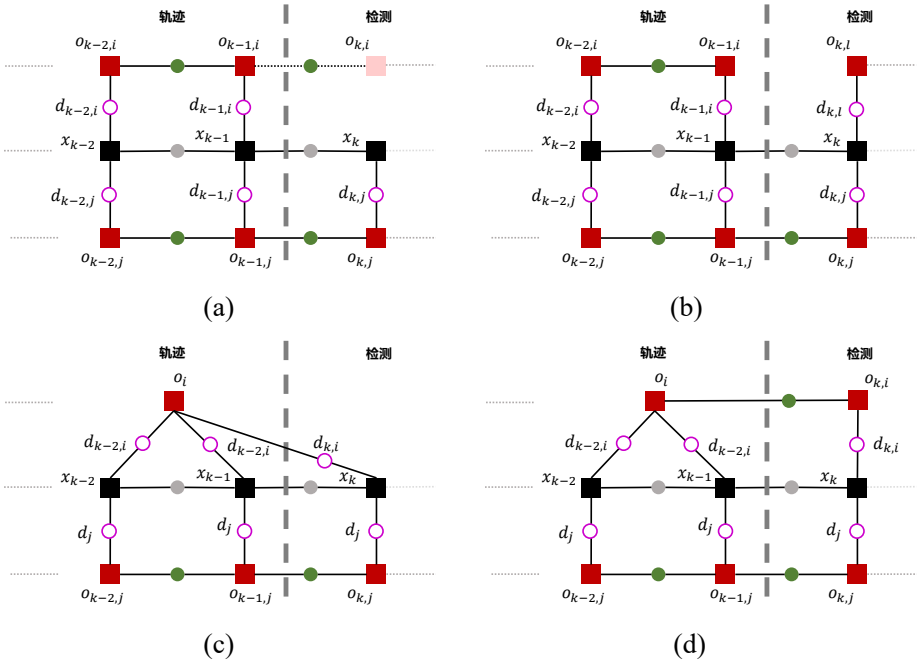


图 5-3 因子图 SLAMMMOT 示意图。(a) 存在未匹配的轨迹，(b) 存在未分配的检测；(c) 静止物体加入因子图；(d) 静止物体发生运动

所建立的因子图示意图如图 5-3所示，其中 (a) (b) (c) (d) 分别表示了不同情况下因子图建立约束的方式。图中黑色方块表示自身位姿节点，灰色圆圈表示帧间运动，红色方块表示目标节点，绿色圆圈表示目标跟踪约束，品红圆圈表示 3D 目标检测约束。

当不存在误检漏检等情况时，理论上的分配结果如图 5-3的 4 个子图下方物体

$o_{k,j}$ 所示，可以直接建立 x_k 和 $o_{k,j}$ 之间的目标检测约束和 $o_{k-1,j}$ 与 $o_{k,j}$ 之间的目标跟踪约束。而实际中检测和跟踪的情况较为复杂，物体可能出现或离开视野，检测器存在漏检和误检，导致检测和轨迹无法完美匹配。以下以两个目标跟踪为例进行介绍。

如在图 5-3中 (a) 的物体轨迹数为 2，当前检测数为 1，检测全部分配到轨迹，存在 1 条未匹配的轨迹，可以建立轨迹和预测结果的约束 (上方物体 $o_{k,i}$，预测结果以浅色方块表示)。

图 5-3中 (b) 的物体轨迹数为 2，当前检测数为 2，1 条检测分配到轨迹，存在 1 条未分配的检测，还需处理新轨迹的出现，建立检测约束（上方物体 $o_{k,l}$）。但也需防止检测为误检，需连续多帧匹配后再加入跟踪，此处的处理在目标跟踪模块完成。

图 5-3中 (c) 的所有检测均分配到轨迹，但物体处于静止，类似于路标的作用。此时将静态物体视为路标，而不是被优化的物体，即不存在目标帧间运动约束。

图 5-3中 (d) 的情况类似于图 (c)，但物体突然发生运动。也对应运动目标突然静止、缓慢运动物体速度估计不准等情况。此时直接建立当前目标和静止物体的约束即可。

本节将这种包含了自身和物体位姿的因子图称为多体位姿图。多体位姿图与基于路标的 EKF-SLAM、基于光束平差法 BA 的视觉 SLAM 类似，都是在位姿图的基础上额外引入优化项，但区别在于除自身外优化的目标是静止或运动的 SE(3) 运动，而上述两者处理只静止的三维点，也无法处理复杂情况下的运动物体。

使用因子图优化 SLAMMOT 问题解决了目标机动的问题，即物体处于“动-静-动”等机动状态。Arya [37] 在其 3D 多目标跟踪中的研究中提出 JPDA-IMM-UKF 滤波器 (如本文5.3节设计的多目标跟踪算法)，是在运动模型的基础上添加静止模型实现的，然而这会导致其他运动模型的退化 [4, 71]。Wang 在提出本文2.1所介绍的贝叶斯 SLAMMOT 的理论基础上，又在 [72] 中提出了一种名为运动-静止假设跟踪的解决方法。

而在本节所描述的紧耦合因子图优化框架下，无需考虑物体的运动模型，避免了因静止模型加入对 IMM 滤波器带来的负面影响。问题的解决主要是利用基于神经网络的 3D 目标检测和简单多目标跟踪器，可以认为是额外引入了语义先验。而传统的多目标跟踪强烈依赖于目标的运动模型。在因子图框架下，如果检测器足够精确，则静止物体退化为路标，只有运动物体和自运动一同以多体里程计的形式进行优化。避免对从运动模型上对物体进行运动分割 (指在 IMM 滤波器中考虑静止模型)，是因子图优化相比贝叶斯估计在广义 SLAMMOT 问题上的优势。

综上，由式 (5-2), 式 (5-3), 式 (5-15), 式 (5-16) 可得，多体位姿图对应的总体误差函数为

$$
\begin{aligned}
\boldsymbol{X}^{*}, \boldsymbol{O}^{*} &= \arg\min_{X,O} \left\{ \sum_{i}^{T} \left\| \boldsymbol{e}_{ego} \right\|_{\Sigma_{ego}}^{2} + \sum_{i}^{K} \left\| \boldsymbol{e}_{loc} \right\|_{\Sigma_{loc}}^{2} + \sum_{i,j}^{T,N} \left\| \boldsymbol{e}_{obj} \right\|_{\Sigma_{obj}}^{2} + \sum_{i,j}^{T,N} \left\| \boldsymbol{e}_{det} \right\|_{\Sigma_{det}}^{2} \right\} \\
&= \arg\min_{X,O} \left\{ \sum_{i}^{T} \boldsymbol{e}_{ego}^{\top} \Sigma_{ego}^{-1} \boldsymbol{e}_{ego} + \sum_{i}^{K} \boldsymbol{e}_{loc}^{\top} \Sigma_{loc}^{-1} \boldsymbol{e}_{loc} + \sum_{i,j}^{T,N} \boldsymbol{e}_{obj}^{\top} \Sigma_{obj}^{-1} \boldsymbol{e}_{obj} + \sum_{i,j}^{T,N} \boldsymbol{e}_{det}^{\top} \Sigma_{det}^{-1} \boldsymbol{e}_{det} \right\} \\
&= \arg\min_{X,O} \left\{ \sum_{i}^{T} \boldsymbol{e}_{ego} \left(\boldsymbol{X}_{t-1}, \boldsymbol{X}_{t}, \Delta \boldsymbol{T}_{t} \right)^{\top} \Sigma_{ego}^{-1} \boldsymbol{e}_{ego} \left(\boldsymbol{X}_{t-1}, \boldsymbol{X}_{t}, \Delta \boldsymbol{T}_{t} \right) \right. \\
&\quad + \sum_{i}^{K} \boldsymbol{e}_{loc} \left(\boldsymbol{X}_{1}, \boldsymbol{X}_{t}, \Delta \boldsymbol{T}_{t}^{loc} \right)^{\top} \Sigma_{loc}^{-1} \boldsymbol{e}_{loc} \left(\boldsymbol{X}_{1}, \boldsymbol{X}_{t}, \Delta \boldsymbol{T}_{t}^{loc} \right) \\
&\quad + \sum_{i,j}^{T,N} \boldsymbol{e}_{obj} \left(\boldsymbol{O}_{t-1}^{n}, \boldsymbol{O}_{t}^{n}, \Delta \boldsymbol{S}_{t}^{n} \right)^{\top} \Sigma_{obj}^{-1} \boldsymbol{e}_{obj} \left(\boldsymbol{O}_{t-1}^{n}, \boldsymbol{O}_{t}^{n}, \Delta \boldsymbol{S}_{t}^{n} \right) \\
&\quad \left. + \sum_{i,j}^{T,N} \boldsymbol{e}_{det} \left(\boldsymbol{X}_{t}^{n}, \boldsymbol{O}_{t}, \boldsymbol{D}_{t} \right)^{\top} \Sigma_{det}^{-1} \boldsymbol{e}_{det} \left(\boldsymbol{X}_{t}^{n}, \boldsymbol{O}_{t}, \boldsymbol{D}_{t} \right) \right\}
\end{aligned}
\tag{5-17}
$$

式中，Σ 为各项对应的协方差矩阵，T，K，N 分别表示时间、关键帧、检测组成的集合。本节发现 DL 法相比 LM 法能取得更优的结果，因此使用使用 DL 法对因子图进行求解。

5.6 实验结果

5.6.1 定位结果

跟踪数据中存在部分完全静止场景，去除了一部分序列。然后选择 KITTI 跟踪数据集中动态物体较多场景下进行测试。部分序列对应的图像如图 5-4所示。

(a) (b)

(c) (d)

图 5-4 动态序列对应的图像。(a) 序列 0000 图像；(b) 序列 0002 图像；(c) 序列 0003 图像；(d) 序列 0004 图像

在具有高动态物体的序列上，对比松耦合和紧耦合两种 SLAMMOT 算法的绝对轨迹误差 ATE，实验结果如表 5-1所示。其中 NDT-MOT-L 是使用松耦合的 SLAMMOT，NDT-MOT-T 是使用紧耦合的 SLAMMOT。

表 5-1 KITTI 跟踪数据集上高动态序列

	ATE(m)	
序列	NDT-MOT-L	NDT-MOT-T
0000	1.65	**1.34**
0002	3.44	**3.12**
0003	**2.87**	2.91
0004	13.65	**12.03**
0005	6.13	**4.03**
0008	17.07	**12.82**
0011	4.29	**3.29**
0020	108.81	**77.47**
平均	19.74	14.63

表中 NDT-MOT-L 使用 JPDA-IMM-UKF 作为多目标跟踪器，直接利用激光 SLAM 提供的自运动信息估计并跟踪多个目标的状态，因此是一种松耦合的算法。NDT-MOT-T 使用 GNN-IMM-UKF 作为多目标跟踪器，并且利用因子图同时优化多个物体的运动和自运动，因此是一种紧耦合的优化方案。

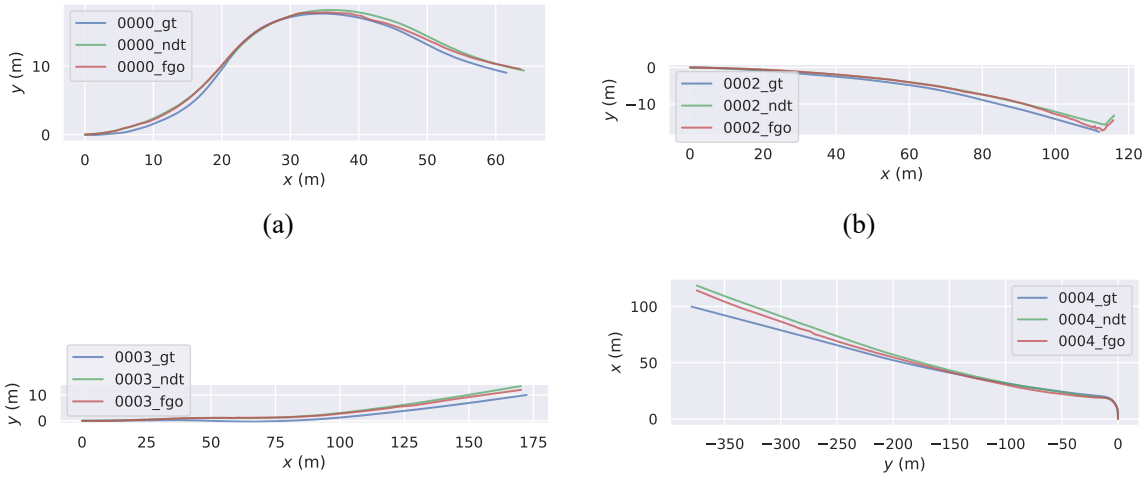


图 5-5 部分序列的轨迹对比图。(a) 序列 0000 轨迹；(b) 序列 0002 轨迹；(c) 序列 0003 轨迹；(d) 序列 0004 轨迹。

NDT-MOT-T 利用目标跟踪提供的物体运动估计和自运动联合构建多体位姿

图并进行联合优化，结果显示在动态物体较多的场景下，使用多体位姿图优化的SLAMMOT 可以取得更好的效果，说明可以利用对目标准确的状态估计辅助定位，从而在高动态场景下利用 MOT 增加 SLAM 定位精度。

图 5-5显示了部分序列上的轨迹情况。表中 gt 为轨迹真值，ndt 为松耦合的 SLAMMOT，对应图 5-5中的 NDT-MOT-L；fgo 为使用因子图优化的紧耦合 SLAMMOT，对应图 5-5中的 NDT-MOT-T。本节提出的紧耦合 SLAMMOT 在动态环境下可以取得更好的结果。从图 5-4中可以看到，本文选择的序列均是在自身运动的同时存在动态物体的序列，并且是单向行驶，没有发生闭环，依赖传统的位姿图等方法无法做出优化，因此本节提出的紧耦合 SLAMMOT 在动态场景下更有效。

在 SLAMMOT 的相关研究中，常使用 RPE 作为评价指标。表 5-2为使用 RPE 的平移部分和多种同类型算法的对比结果。其他算法的结果取自原始论文，- 表示论文中并未提供。

表 5-2 与其他 SLAMMMOT 方法的对比结果

序列	ORB SLAM2 [13]	CubeSLAM [73]	VDO SLAM [10]	DynaSLAM II [11]	Ours
0000	0.04	-	0.05	0.04	0.04
0001	0.05	-	0.12	0.05	0.06
0002	0.04	-	0.04	0.04	0.04
0003	0.07	0.05	0.09	0.06	0.06
0004	0.07	0.07	0.11	0.07	0.07
0005	0.06	0.03	0.10	0.06	0.05
0006	0.02	-	0.02	0.02	0.02
0007	0.05	-	-	0.05	0.05
0008	0.08	-	-	0.10	0.07
0009	0.06	-	-	0.06	0.06
0010	0.07	-	-	0.07	0.07
0018	0.05	0.04	0.07	0.05	0.05
0020	0.11	0.13	0.16	0.07	0.05
平均	0.055	0.064	0.084	0.057	0.053

多目标跟踪增加了 SLAM 的物体感知能力，但联合优化却对自身位姿造成了影响。表中 Cube SLAM, VDO SLAM 和 DynaSLAMII 均构建在 ORB SLAM2 的基础上，都同时估计物体的位置并联合优化，但从 RPE 结果上看，所产生的误差相比改进前更大。这表明当前 SLAMMOT 方案对于如何在 SLAM 算法中利用动态物体和联合优化的方式上存在问题，有待探索更适宜的理论。虽然多种算法在改进后相对位姿误差略有增加，但 SLAM 借以感知周围物体并重建语义场景。

尽管本节提出的算法总体表现更优，但所使用的传感器并不相同，无法直接对结果直接进行对比。如 Cube SLAM 是单目视觉 SLAM，VDO SLAM 是使用了单目深度估计的 RGB-D SLAM，DynaSLAM II 和 ORB SLAM2 是双目视觉 SLAM。本节使用的传感器为激光雷达，但为获取物体检测的信息，仅保留相机视角下的点云，这对激光 SLAM 算法本身也产生了影响，激光里程计更容易在 z 轴上产生错误的估计，激光雷达视角变小导致提取特征变少，部分激光 SLAM 算法不再适用。

使用不同传感器影响了物体检测和 SLAM 本身的精度，如基于图像的 SLAM 更易受到动态物体的干扰，因此提升较为明显，而激光 SLAM 本就对于动态物体更为鲁棒；由于估计原理不同，单目或双目 SLAM 在自身缓慢运动或静止的情况下表现较激光 SLAM 更稳定，激光 SLAM 在此情况下可能会出现对物体位姿的错误估计和优化导致失败。

5.6.2 建图和跟踪结果

图 5-6为跟踪数据集的 0000 序列的建图结果。图中黑线表示自身运动轨迹，以不同颜色标注出了物体的轨迹和 ID，并重建了周围环境。由于使用多目标跟踪器对场景中的物体进行状态估计，因此除物体的位置外，还可以获取物体的速度、运动模型、状态协方差等信息，这些信息的获取将有利于后续的规划控制等任务。

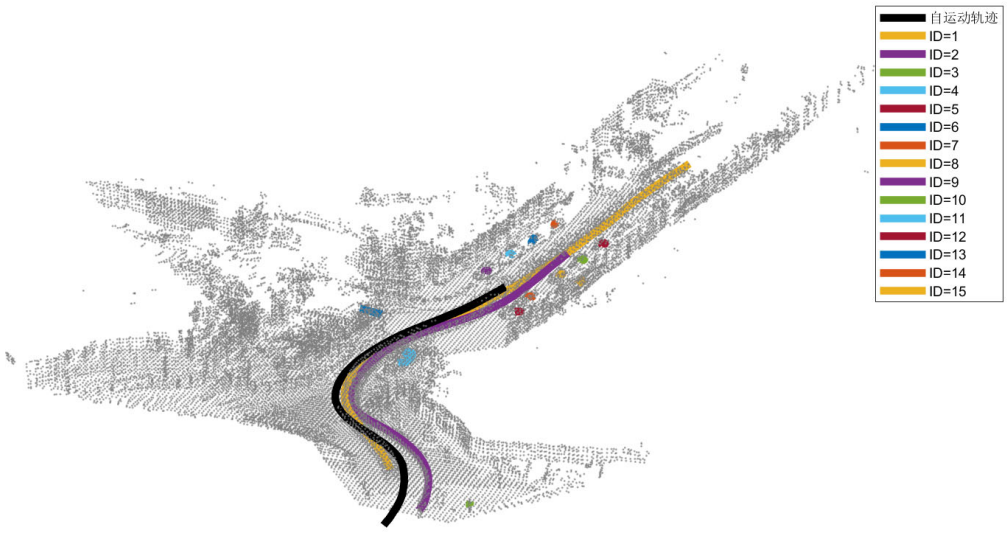


图 5-6 自身轨迹和被跟踪物体的轨迹

SLAM 仅利用静态信息进行定位，这种方式为常见的动态 SLAM 算法，思路为检测并去除动态物体。本节表 5-1中所示的松耦合方法 NDT-MOT-L，在动态 SLAM 的基础上，使用 JPDA-IMM-UKF 滤波器对物体进行跟踪，一定程度上解决

了检测器的误检和漏检问题，但检测到的动态物体仍然被丢弃。因此将这种方法称为 SLAM 和 MOT 的松耦合。

表 5-3 对物体运动分割采用的速度阈值

类别	速度阈值 (m/s)
汽车	1
自行车	1
行人	0.15

本节提出的紧耦合 SLAMMOT 方法 DT-MOT-T，对于检测到的物体，统一估计其运动并加入优化。对于其中的 SLAM 系统，利用静态物体和静态环境组成的完整静态场景信息进行定位。相比表 4-1处运动分割使用的帧间运动距离，此处采用的速度阈值，如表 5-3所示。速度低于阈值的物体被加入环境，再次进行一次点云配准，在动态环境下利用完全的静态信息进行定位。所有的物体在帧间的运动均以 SE(3) 运动的形式加入因子图，静态物体运动将退化为 3D 点，动态物体会形成类似激光里程计的物体里程计，一同组成多体位姿图。

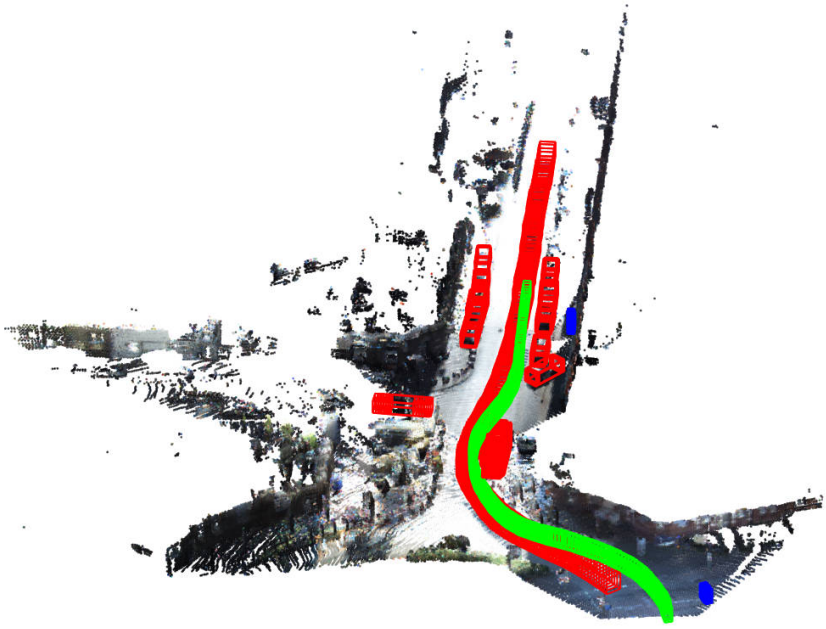


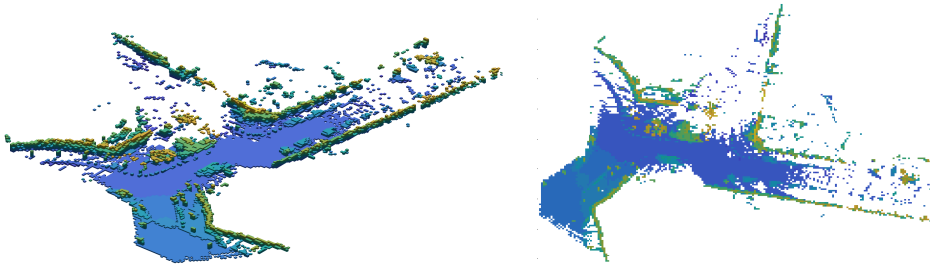
图 5-7 稠密 RGB 地图

本节提出的 SLAMMOT 基于动态配准建立，其中 SLAM 部分仅使用静态环境和静态物体用于定位。SLAMMOT 将 MOT 获得的物体位置等信息转为里程计的形式并加入因子图中，与自运动的激光里程计一同构建多体位姿图并进行优化，因此可以认为，本节提出的算法是在基于静态环境的 SLAM 的基础上额外设计的

模块，是通过提取动态物体中的信息进行定位。不但在动态场景下分割出静态环境用于 SLAM 定位，还利用对物体准确的追踪提升自身定位精度。这是本节工作的核心所在。

在第三章的工作基础上，融合图像建立起稠密 RGB 地图，如图 5-7所示。红色表示汽车，绿色表示自行车，蓝色表示行人，将他们的包围框绘制在点云地图上。多传感器融合的 SLAMMOT 方法可以在动态场景下，准确地重建静态场景，并估计动态物体的运动轨迹和状态，而且 SLAMMOT 可以利用对物体的跟踪结果提升定位精度，增强了传统 SLAM 和动态 SLAM 的感知能力。

重建出的稠密地图包含过多细节，且无法表示可通行区域，故一般不在机器人导航领域中使用，而是使用八叉树地图或占据栅格地图。将获得的静态环境地图转为八叉树地图并输出，如图 5-8 (a) 所示。如果假定无人车运行在水平面上，则 2D 的占据栅格地图使用更多，可以将八叉树地图投影到 x-y 平面得到 2D 导航中使用更广的占据栅格地图，如图 5-8 (b) 所示。



(a) (b)

图 5-8 八叉树占据网格地图。(a) 八叉树地图；(b) 2D 栅格地图。

SLAMMOT 也可以增强八叉树地图和占据栅格地图对动态物体的表示能力，由于使用多目标跟踪获取物体的位置、速度和历史轨迹等信息，如图 5-6所示，因此可以更新或直接表示动态物体，帮助移动机器人在动态环境下实现更安全的导航。

5.7 本章小节

本节将动态配准扩展为 SLAM 系统，首先设计了激光 SLAM 子系统。在点云配准前，使用地面分割去除地面点，然后构建激光里程计获得自身位姿，使用局部地图匹配获得关键帧位姿。设计了完整的多目标跟踪系统，并可以和 SLAM 进行松耦合和紧耦合。对于松耦合的情况，使用 SLAM 与 JPDA-IMM-UKF 通过动态配准连接；对于紧耦合的情况，构建因子图加进入目标检测和目标跟踪因子，然后执行优化并更新状态。

第六章 总结与展望

6.1 全文总结

本文在提出了动态环境下定位的 SLAMMOT 方法。首先介绍了动态环境下的 SLAM 定位问题，以及提出 SLAMMOT 的方案。然后对 SLAMMOT 涉及的基础知识，包括 SLAM 和 MOT 理论，以及本文使用的因子图方案进行简单介绍。然后提出基于稀疏深度补全的视觉激光融合 SLAM，动态环境下基于迭代动态配准的 SLAM，以及基于动态配准的因子图优化 SLAMMOT，并在公开数据集上验证提出方法的精度。具体的工作和创新如下：

(1) 提出一种视觉激光融合的 SLAM 方案。使用深度补全对稀疏补全稀疏深度图，然后分别设计视觉和激光里程计以及各自的局部建图和关键帧模块。最后使用因子图对视觉激光观测进行融合和优化。最终在 KITTI 里程计数据集上多个序列的平均绝对轨迹误差为 5.15m。

(2) 提出一种联合自运动估计和运动目标分割的动态配准算法。使用 3D 目标检测获得所有可能处于运动的物体，然后利用自运动估计和物体距离进行迭代分割，最终输出位姿和环境中的动态物体。最终在 KITTI 跟踪数据集上多个序列的平均相对位姿的旋转平移总误差为 0.40 和 0.48。

(3) 提出一种基于动态配准的同时定位建图和多目标跟踪方法。利用多目标跟踪算法估计目标的姿态和速度，然后将物体加入因子图中和自身位姿联合优化，获得稳定的自身和物体轨迹。最终在 KITTI 跟踪数据集上多个序列的平均相对位姿的平移误差为 0.053m。

6.2 工作展望

本文对同时定位建图和目标跟踪展开研究，提出的方案一定程度上解决了动态环境下 SLAM 定位问题。但仍存在不足之处，将通过未来的工作继续改进:

(1) 在统一框架下实现 SLAM 和目标跟踪问题。目前方案中数据关联是单独实现的，依靠状态估计的误差对目标和自身优化。如何在统一框架下解决该问题有待研究，目前缺乏相关研究和理论支撑。

(2) 同时定位建图与目标跟踪的工程实现。目前已初步完成在 MMCV 开源框架和 ROS 平台下对目标检测和 SLAM 的开发部署，而还面临的核心挑战是多目标跟踪的工程实现。将尝试在如 stonesoup 等开源库的帮助下逐步完成此目标。

致 谢

在清水河的三年即将过去，无忧无虑的研究和学习的时光即将过去。这段旅程是孤独的，但从未有过遗憾。首先我要感谢我的导师汪子君老师，在我进行研究的过程中给予了我耐心的指导和支持，使我能够深入探究各种领域的知识。还要感谢作为本论文的校外导师，清华大学张新钰老师和郭世纯老师对研究做出的指导和帮助。几位老师老师严谨的学术态度、深厚的学术造诣和高度的责任心让我深受启发。在他们的指导下，我逐渐明确了研究的方向并真正开展研究，感谢他们的专业知识、耐心解答和宝贵建议，让我在研究中获得了很多启示和收获。

在这里还要感谢阎啸和李滚两位老师。在研一上过他们的信号检测与估计课程后，我对信息融合和状态估计产生了浓厚的兴趣，这也直接启发了本文的研究。然后要感谢在本科和硕士阶段所有指导过我的老师，我以认真的求学态度对待每一门课，他们也展现出了作为老师对待学生的耐心指导和专业素养。我特别要感谢本科兰州理工大学的刘微容老师，是他的指导和鼓励让我有勇气在电子科技大学进行学习，三年中，每次遇到困难，心中总会浮现他温文尔雅的样子。

在研究生求学的道路上，我还受益于一批出色的同学们的帮助和支持。有幸能在研究生阶段于张老师实验室进行学习和工作，感谢实验室的张世焱博士、李志伟博士、王力博士、朱世凡同学、赵旭东同学、熊一瑾同学等，他们让我感受到了学习和研究的乐趣，他们的合作和协助让我的研究更加顺利。最后要感谢在电子科技大学认识的各位朋友和同学，感谢他们对我的友情帮助和真挚关怀。在与他们相处的过程中，我也学到了很多实际的技能和知识。

此外，我还要感谢所有对我的研究工作做出贡献的人，谢谢你们的支持和帮助。我会继续努力，让我的研究成果更加优秀。

参考文献

[1] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 1168-1174.

[2] Bescos B, Fácil J M, Civera J, et al. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

[3] Wang C C, Thorpe C, Thrun S. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas[C]. 2003 IEEE International Conference on Robotics and Automation (ICRA), 2003: 842-849.

[4] Wang C C, Thorpe C, Thrun S, et al. Simultaneous Localization, Mapping and Moving Object Tracking[J]. The International Journal of Robotics Research, 2007, 26(9): 889-916.

[5] Sabzevari R, Scaramuzza D. Multi-body motion estimation from monocular vehicle-mounted cameras[J]. IEEE Transactions on Robotics, 2016, 32(3): 638-651.

[6] Lang A H, Vora S, Caesar H, et al. PointPillars: Fast Encoders for Object Detection From Point Clouds[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019,6: 12689-12697.

[7] Kaess M, Johannsson H, Roberts R, et al. iSAM2: Incremental smoothing and mapping using the Bayes tree[J]. The International Journal of Robotics Research, 2012, 31(2): 216-235.

[8] Wang H, Sun J, Lu S, et al. Factor graph aided multiple hypothesis tracking[J]. Science China Information Sciences, 2013, 56(10): 1-6.

[9] Pöschmann J, Pfeifer T, Protzel P. Factor graph based 3d multi-object tracking in point clouds[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: 10343-10350.

[10] Zhang J, Henein M, Mahony R, et al. Vdo-slam: A visual dynamic object-aware slam system[J]. arXiv preprint arXiv:2005.11052, 2020.

[11] Bescos B, Campos C, Tardós J D, et al. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 5191-5198.

[12] Qiu Y, Wang C, Wang W, et al. AirDOS: Dynamic SLAM benefits from articulated objects[C]. 2022 International Conference on Robotics and Automation (ICRA), 2022: 8047-8053.

[13] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.

[14] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time.[C]. Robotics: Science and Systems, 2014.

[15] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European Conference on Computer Vision, 2014: 834-849.

[16] Xu W, Cai Y, He D, et al. Fast-lio2: Fast direct lidar-inertial odometry[J]. arXiv preprint arXiv:2107.06829, 2021.

[17] Saputra M R U, Markham A, Trigoni N. Visual SLAM and Structure from Motion in Dynamic Environments: A Survey[J]. ACM Computing Surveys, 2018, 51(2): 1-36.

[18] Vincent J, Labbe M, Lauzon J S, et al. Dynamic Object Tracking and Masking for Visual SLAM[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020: 4974-4979.

[19] Chen X, Milioto A, Palazzolo E, et al. Suma++: Efficient lidar-based semantic slam[C]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019: 4530-4537.

[20] Pfreundschuh P, Hendrikx H F C, Reijgwart V, et al. Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data[C]. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021: 11641-11647.

[21] Chung S Y, Huang H P. SLAMMOT-SP: Simultaneous SLAMMOT and scene prediction[J]. Advanced Robotics, 2010, 24(7): 979-1002.

[22] Choi J, Maurer M. Local Volumetric Hybrid-Map-Based Simultaneous Localization and Mapping With Moving Object Tracking[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(9): 2440-2455.

[23] Wang Z, Li W, Shen Y, et al. 4-D SLAM: An Efficient Dynamic Bayes Network-Based Approach for Dynamic Scene Understanding[J]. IEEE Access, 2020, 8: 219996-220014.

[24] Ma T, Ou Y. MLO: Multi-Object Tracking and Lidar Odometry in Dynamic Envirnoment[J]. arXiv:2204.11621 [cs], 2022.

[25] Tian X, Zhao J, Ye C. DL-SLOT: Dynamic Lidar SLAM and Object Tracking Based On Graph Optimization[J]. arXiv:2202.11431 [cs], 2022.

[26] 周风余, 顾潘龙, 万方 等. 多运动视觉里程计的方法与技术 [J]. 山东大学学报 (工学版), 2021, 51(01): 1-10.

[27] Wang C, Luo B, Zhang Y, et al. DymSLAM: 4D Dynamic Scene Reconstruction Based on Geometrical Motion Segmentation[J]. IEEE Robotics and Automation Letters, 2020, 6(2): 550-557.

[28] Judd K M, Gammell J D, Newman P. Multimotion visual odometry (mvo): Simultaneous estimation of camera and third-party motions[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 3949-3956.

[29] Kundu A, Krishna K M, Jawahar C V. Realtime multibody visual SLAM with a smoothly moving monocular camera[C]. 2011 International Conference on Computer Vision, 2011: 2080-2087.

[30] 王晨捷, 张云, 赵青 等. 分裂合并运动分割的多运动视觉里程计方法 [J]. 中国图象图形学报, 2020, 25(09): 1859-1868.

[31] Henein M, Kennedy G, Mahony R, et al. Exploiting rigid body motion for SLAM in dynamic environments[C]. Proc. IEEE Int. Conf. Robot. Automat., 2018: 19.

[32] Henein M, Zhang J, Mahony R, et al. Dynamic SLAM: The need for speed[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020: 2123-2129.

[33] Zhang J, Henein M, Mahony R, et al. Robust Ego and Object 6-DoF Motion Estimation and Tracking[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: 5017-5023.

[34] 蔡鹤皋, 金明河, 金峰. 卡尔曼滤波与多传感器数据融合技术 [J]. 模式识别与人工智能, 2000, 13(3): 248-253.

[35] Tugnait J K. Detection and estimation for abruptly changing systems[J]. Automatica, 1982, 18(5): 607-615.

[36] Blom H, Bar-Shalom Y. The interacting multiple model algorithm for systems with Markovian switching coefficients[J]. IEEE Transactions on Automatic Control, 1988, 33(8): 780-783.

[37] Arya Senna Abdul Rachman. 3D-LIDAR Multi Object Tracking for Autonomous Driving: Multi-target Detection and Tracking under Urban Road Uncertainties[D]. Delft University of Technology, 2017, 21-66.

[38] Wang C C, Thorpe C. Simultaneous localization and mapping with detection and tracking of moving objects[C]. Proceedings 2002 IEEE International Conference on Robotics and Automation, 2002: 2918-2924.

[39] Shin Y S, Park Y S, Kim A. Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: 5144-5151.

[40] Shin Y S, Park Y S, Kim A. Dvl-slam: Sparse depth enhanced direct visual-lidar slam[J]. Autonomous Robots, 2020, 44(2): 115-130.

[41] Wang W, Liu J, Wang C, et al. DV-LOAM: Direct visual lidar odometry and mapping[J]. Remote Sensing, 2021, 13(16): 3340.

[42] Li N, Ho C P, Xue J, et al. A Progress Review on Solid-State LiDAR and Nanophotonics-Based LiDAR Sensors[J]. Laser & Photonics Reviews, 2022, 16(11): 2100511.

[43] Li K, Li M, Hanebeck U D. Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 5167-5174.

[44] Li L, Ismail K N, Shum H P, et al. Durlar: A high-fidelity 128-channel lidar dataset with panoramic ambient and reflectivity imagery for multi-modal autonomous driving applications[C]. 2021 International Conference on 3D Vision (3DV), 2021: 1227-1237.

[45] Yu H, Luo Y, Shu M, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 21361-21370.

[46] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.

[47] Zhang J, Singh S. Visual-lidar odometry and mapping: Low-drift, robust, and fast[C]. 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015: 2174-2181.

[48] Graeter J, Wilczynski A, Lauer M. Limo: Lidar-monocular visual odometry[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 7872-7879.

[49] Ma F, Cavalheiro G V, Karaman S. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera[C]. 2019 International Conference on Robotics and Automation (ICRA), 2019: 3288-3295.

[50] Ku J, Harakeh A, Waslander S L. In defense of classical image processing: Fast depth completion on the cpu[C]. 2018 15th Conference on Computer and Robot Vision (CRV), 2018: 16-22.

[51] Sauerbeck F, Obermeier B, Rudolph M, et al. RGB-L: Enhancing Indirect Visual SLAM using LiDAR-based Dense Depth Maps[J]. arXiv preprint arXiv:2212.02085, 2022.

[52] 黄漫, 黄勃, 高永彬. 引入深度补全与实例分割的三维目标检测 [J]. 传感器与微系统, 2021, 40(1): 129-132.

[53] Gao X S, Hou X R, Tang J, et al. Complete solution classification for the perspective-three-point problem[J]. IEEE transactions on pattern analysis and machine intelligence, 2003, 25(8): 930-943.

[54] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment—a modern synthesis[C]. Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings, 2000: 298-372.

[55] Chen C, Pei L, Xu C, et al. Trajectory Optimization of LiDAR SLAM Based on Local Pose Graph[C]. China Satellite Navigation Conference (CSNC) 2019 Proceedings, Singapore, 2019: 360-370.

[56] Zhang J, Singh S. Low-drift and real-time lidar odometry and mapping[J]. Autonomous Robots, 2017, 41(2): 401-416.

[57] Shan T, Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 4758-4765.

[58] Koide K, Miura J, Menegatti E. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement[J]. International Journal of Advanced Robotic Systems, 2019, 16(2): 1729881419841532.

[59] Yan M, Wang J, Li J, et al. Loose coupling visual-lidar odometry by combining VISO2 and LOAM[C]. 2017 36th Chinese Control Conference (CCC), 2017: 6841-6846.

[60] Dang X, Rong Z, Liang X. Sensor Fusion-Based Approach to Eliminating Moving Objects for SLAM in Dynamic Environments[J]. Sensors, 2021, 21(1): 230.

[61] Wang W, Yu R, Huang Q, et al. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2569-2578.

[62] Zhou D, Fang J, Song X, et al. Joint 3D Instance Segmentation and Object Detection for Autonomous Driving[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 1836-1846.

[63] Zhou Y, Tuzel O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: 4490-4499.

[64] Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.

[65] Besl P J, McKay N D. Method for registration of 3-D shapes[C]. Sensor Fusion IV: Control Paradigms and Data Structures, 1992: 586-606.

[66] Biber P, Strasser W. The normal distributions transform: A new approach to laser scan matching[C]. Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), 2003: 2743-2748vol.3.

[67] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]. 2016 IEEE International Conference on Image Processing (ICIP), 2016: 3464-3468.

[68] Weng X, Wang J, Held D, et al. 3d multi-object tracking: A baseline and new evaluation metrics[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: 10359-10366.

[69] Munkres J. Algorithms for the assignment and transportation problems[J]. Journal of the society for industrial and applied mathematics, 1957, 5(1): 32-38.

[70] Bogoslavskyi I, Stachniss C. Efficient online segmentation for sparse 3D laser scans[J]. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 2017, 85: 41-52.

[71] Shea P J, Zadra T, Klamer D, et al. Precision tracking of ground targets[C]. 2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484), 2000: 473-482.

[72] Wang C C. Simultaneous localization, mapping and moving object tracking[D]. Pittsburgh: Carnegie Mellon University, 2004, 73-88.

[73] Yang S, Scherer S. Cubeslam: Monocular 3-d object slam[J]. IEEE Transactions on Robotics, 2019, 35(4): 925-938.