中图分类号：TP391

单位代码：11414
学　　号：2020215941

中国石油大学
CHINA UNIVERSITY OF PETROLEUM

硕士专业学位论文

题　　目	复杂环境下基于点线特征的视觉SLAM算法研究
专业或领域	计算机技术
研究方向	人工智能及应用
专业硕士生	孙浩
指导教师	连远锋副教授
现场导师	张春晓高级工程师

二〇二三年五月

摘　要

随着油气系统开始向智能化方向发展，智能机器人已经成为了代替传统人工巡检的首要选择。而即时定位与地图构建（Simultaneous Localization and Mapping，SLAM）是在未知环境中进行感知以及定位的基本方法，也是通过机器人进行自主导航而实现油气场站无人值守的关键技术。近些年 SLAM 技术受到越来越多的研究人员关注，在弱纹理的人工室外环境下传统的视觉 SLAM 由于难以提取足够的特征，容易导致其出现较大的误差，从而对系统的定位精度产生较大影响。针对此问题，本工作在传统的点特征的基础上加入线特征，通过点线特征各自的特性来提高系统的稳定性。由于传统的线特征提取方法大幅度增加计算量，故本工作利用深度学习的方法解决点线特征的提取和匹配问题，提出一种面向点线感知的异构图注意力视觉 SLAM。进一步针对动态环境下 SLAM 系统的鲁棒性和定位精度下降问题，提出了一种面向动态环境的实时语义 SLAM，主要研究内容如下：

（1）针对复杂工业场景下由于纹理单一、几何结构复杂等原因导致的位姿估计精度低、鲁棒性差等问题，论文设计了一种面向点线感知的异构图注意力视觉 SLAM 系统。首先，利用点与点、点与直线的几何关系构建了同步点线特征提取网络，分别利用点线感知注意模块引导网络获取关键区域几何关联特征；并通过感知迁移的知识蒸馏策略对网络模型优化，进一步提高系统实时性。其次，为了提升图像点线匹配精度，提出了一种点线异构图注意力网络。利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间学习，提高几何特征匹配结果的准确率。最后，将点线匹配问题转换为一个最优传输问题，构建结合贪婪策略的最邻近点迭代方法（Greedy Inexact Proximal Point Method for Optimal Transport，GIPOT）对该优化问题进行求解。

（2）由于在动态环境中现有的视觉 SLAM 方法容易失败，因此引入深度学习生成的语义信息来消除动态对象的影响。为了降低计算量，这里只对关键帧进行语义分割以去除动态对象，并保持一个静态环境下进行相机跟踪。此外，为了检测未知的动态目标，提出了一个轻量级的分割模块用以获得动态目标的像素级分割结果，并通过运动自估计模块计算相机姿态，进而通过两种不同策略来剔除动态特征点，实现在低功耗嵌入式平台上实时运行的语义 SLAM 系统。

（3）为了验证所提 SLAM 系统在油气场站的适用性，利用 Unity3D 虚拟引擎开发了油气场站仿真系统。通过虚拟场景四足机器人嵌入所提出的 SLAM 系统来

运行和测试在油气场站环境中的实验效果。并与主流的视觉 SLAM 系统进行比较，通过实验结果对所提出系统的定位精度、实时性等方面进行了分析，验证了所提出的视觉 SLAM 算法的准确性和有效性，具有广泛的适用性和应用价值。

关键词：即时定位与地图构建；视觉里程计；特征匹配；图神经网络；语义分割

Research on Visual SLAM Algorithm with Point and Line Features in Complex Environment

ABSTRACT

As oil and gas systems move toward greater intelligence, intelligent robots have become the preferred choice to replace traditional manual inspections. Simultaneous Localization and Mapping (SLAM) is a basic method for perception and localization in unknown environments, and is also a key technology for achieving unattended oil and gas stations through autonomous robot navigation. In recent years, SLAM technology has received increasing attention from researchers. However, traditional visual SLAM methods struggle to extract sufficient features in weakly textured, artificial outdoor environments. This can result in large errors and significantly impact the positioning accuracy of the system. To address this challenge, this study adds line features on the basis of traditional point features to improve the stability of the system. As traditional line feature extraction methods significantly increase computational complexity, this paper uses deep learning methods to solve the extraction and matching of point-line features, proposing a point-line aware-based heterogeneous graph attention visual SLAM. Further, a real-time semantic SLAM for dynamic environments is proposed to address the robustness and accuracy degradation problems of visual SLAM system localization in dynamic environments. The contents of this paper are as follows:

(1) To address the problems of low accuracy and poor robustness in bit-pose estimation caused by single texture and complex geometric structures in complex industrial scenes, this paper proposes a heterogeneous graph attention visual SLAM system for point-line aware. Firstly, a synchronous point and line feature extraction network is constructed based on the geometric relationships between points-points and points-lines. The point-line aware attention module is utilized to guide the network to obtain geometrically associated features in key regions. Moreover, the network model is optimized using a knowledge distillation strategy of aware transfer to further enhancing

the system's real-time performance. Secondly, a point-line heterogeneous graph attention network is proposed to enhance the accuracy of image point-line matching. By utilizing the edge-aggregated graph attention module and cross-heterogeneous graph iterative module, the intra-graph and inter-graph learning are respectively achieved to improve the accuracy of geometric feature matching. Finally, the point-line matching problem is transformed into an optimization problem, and a greedy inexact proximal point method is developed to solve this optimal transport problem.

(2) Due to the fact that existing visual SLAM methods are prone to failure in dynamic environments, deep learning-generated semantic information is introduced to eliminate the influence of dynamic objects. To reduce computational complexity, only keyframes are semantically segmented to remove dynamic objects, while a static map is maintained for camera tracking. Furthermore, a lightweight segmentation module is proposed to detect unknown dynamic objects and obtain pixel-level segmentation results, which are then used by a motion self-estimation module to compute the camera pose. Two different strategies are employed to remove dynamic feature points, resulting in a real-time semantic SLAM system that can run on low-power embedded platforms.

(3) To verify the applicability of the SLAM system proposed in this paper in oil and gas stations, we developed a simulation system using the Unity3D virtual engine. By embedding the SLAM system into a virtual scene of a quadruped robot, we were able to run experiments and test its performance in the oil and gas field station environment. We also compared our system with mainstream visual SLAM systems and analyzed the accuracy and effectiveness of our proposed visual SLAM algorithm based on experimental results in terms of positioning accuracy, real-time performance, and other relevant factors. Our findings confirm the accuracy and effectiveness of the proposed algorithm, and demonstrate its wide applicability and significant practical value.

Key Words：SLAM；Visual odometry；Feature matching；Graph neural network；Semantic segmentation

目　录

第 1 章　绪论

1.1　研究背景及意义

近年来，随着机器人控制等领域的快速进步以及人工智能、深度学习等技术的日益成熟，自主定位技术在军事、农业、医疗、服务等方面发挥着越来越重要的作用[1]。特别是在工业工程领域，自主定位技术具有重要的实用价值。尽管北斗、GPS 等全球定位系统技术已经得到了广泛应用[2]，但在室内、高楼耸立的街道和遮挡物较多的场景下无法正常工作，并且定位精度无法满足自主导航任务需求。因此，即时定位与地图构建（Simultaneous Localization and Mapping，SLAM）算法应运而生。即在未知环境中，SLAM 技术可以在进行对机器人实时自主定位的同时完成周围环境地图的构建[3]。通过利用机器人上的传感器信息，例如相机、激光雷达等，SLAM 技术可以对环境进行建模和定位，从而实现对机器人或车辆的自主导航和控制。如图 1.1 所示，SLAM 在机器人[4]、辅助驾驶[5]、三维重建[6]以及智能物流车[7]等各个领域都得到了广泛应用。

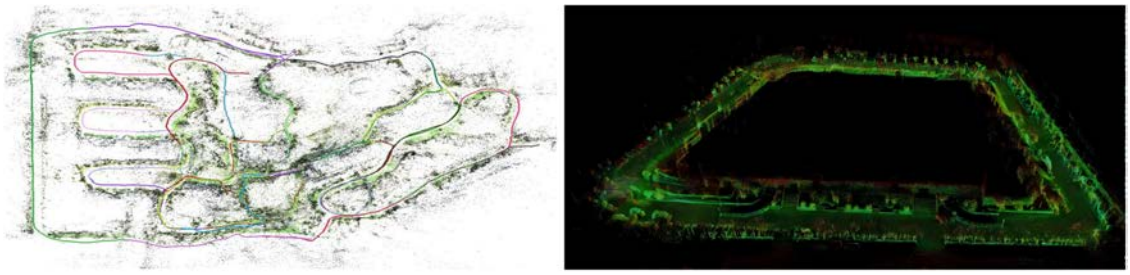
（a）场站巡检机器人　（b）自动驾驶

（c）三维重建　（d）智能物流车

图 1.1 SLAM 的相关应用

Fig. 1.1 Related applications of SLAM

在多种 SLAM 自主定位技术方案中，根据获取环境信息传感器的类型可将 SLAM 划分为视觉 SLAM 和激光 SLAM，其构建的地图如图 1.2 所示。激光 SLAM 的发展较为成熟，能够精确地获得机器人周围环境的距离和方位信息，但成本价格较高，一般在室内进行使用。相比之下，视觉 SLAM 根据相机作为传感器在室内室外都有一定的适用性，对环境的约束较少。同时，相机具有功耗低、成本低、小型化等优点，更关键的是能够获取周围环境中的场景、纹理和颜色等丰富信息。此外，深度学习在计算机视觉中的广泛应用为视觉 SLAM 技术带来了发展机遇，但该技术在复杂的工业环境（如油气场站）中应用受限。因此，实现高鲁棒性和高精度的视觉 SLAM 系统是当前需要解决的技术难题。



（a）视觉地图　　（b）激光地图

图 1.2 构建地图

Fig. 1.2 Mapping

随着深度学习技术的发展，结合深度学习的视觉 SLAM 算法已经取得了一定的进展，但也面临一些挑战。对于依赖相邻图像进行相对位姿估计的方法，将导致运动目标的位姿估计会根据时间累积大量误差进而发生轨迹漂移。为了解决这个问题，一些研究者开始探索如何在局部范围内对相机预测位姿进行约束与优化，例如采用滑动窗口的方法[8]，将相邻帧的信息进行联合优化以降低累积误差。此外，在视觉 SLAM 中，深度学习模型常常用于解决图像配准、特征提取、场景重建和位姿估计等问题。然而，深度学习模型的参数较多，容易导致过拟合现象。过拟合是指模型在训练集上表现出色，但在测试集上表现较差。视觉 SLAM 系统在未知环境下进行自主定位和场景重建，由于环境的复杂性和不确定性，算法的鲁棒性和泛化能力会受到挑战，从而影响系统的定位精度和稳定性。但使用深度学习方法来替换传统 SLAM 中的一个或几个模块可以取得不错的效果。

综上所述，在未知复杂环境下结合深度学习方法提取更丰富的特征，从而提升算法的稳定性和定位精度已经成为视觉 SLAM 前沿研究方向。

1.2 国内外研究现状

1.2.1 视觉 SLAM 技术

传统的视觉 SLAM 是一种利用相机捕获的图像信息来推测相机在空间中的位置和姿态的方法。根据图像特征提取和匹配的方式，视觉 SLAM 被划分为直接法和间接法。直接法是一种根据图像像素光度值的方法来估计相机运动的技术，通过最小化图像像素光度值的误差来推断相机在移动过程中的位姿。Newcombe 等人[9]提出的 DTAM 系统采用直接法通过连续的图像帧之间的稠密匹配来跟踪相机运动，但在大规模场景中的运行效率较低，同时也对计算能力有一定要求，因此并不适用于所有的实时应用场景。相比之下，文献[10]提出了半稠密 LSD−SLAM 系统，通过利用图像中高像素梯度来估计相机的位姿并构建大规模场景地图。Forster 等人[11]提出了经典的半直接 SVO，该方法使用亚像素特征对应来提高精度，并且只对选定的关键帧进行特征提取，由于估计的像素数较少并且不需要计算描述子，因此其提取效率非常高。直接法 SLAM 系统不受特征点数量的限制，适用于各种场景，但其计算量较大，并且对于运动模糊和相机畸变等问题不太稳健，处理遮挡问题较为困难。

间接法是通过测量环境中特征点的位置信息确定自己的位置并构建环境地图。文献[12]提出的 MonoSLAM 是一种实时的视觉 SLAM，将 SLAM 问题建模为一个状态估计问题，并采取扩展卡尔曼滤波算法（Extended Kalman Filter，EKF）对问题求解。但由于需要不断地更新状态和协方差矩阵会导致特征点跟踪容易受到噪声干扰、对于一些复杂场景不易建模等问题。此外，Klein 等人[13]提出的 PTAM 是一种依据单目相机的视觉 SLAM 算法，通过将相机的位姿估计和地图构建分别处理，并行运行，实现了高效的场景重建和相机位姿估计。该算法采用了依据过滤关键帧的策略来提高性能，虽然结合过滤关键帧的视觉 SLAM 方法在小规模环境中表现良好，但在大规模场景中遇到尺度模糊等挑战。Mur−Artal 等人[8,14]提出的 ORB−SLAM 是一种利用特征点法的经典视觉 SLAM 系统，该系统采用 FAST 算法[15,16]进行关键点检测，并使用二进制向量[17]作为特征点描述符，能够实时地提取和匹配关键点。其特点在于创建三个并行线程，分别用于特征点跟踪、局部建图优化和全局位姿优化，并具有跟踪故障后的重新定位以及回环检测功能。利用特征点的 SLAM 方法依赖于高鲁棒性的关键点和描述子，对光照和旋转等因素表现出不敏感性，但在处理存在运动模糊、纹理缺失、环境遮挡等情况的图像时表现脆弱。

1.2.2　融合深度学习的视觉 SLAM

近些年，基于深度学习的视觉 SLAM 研究引起了广泛关注。CNN-SLAM[18]是一种典型的方法，它利用卷积神经网络进行图像特征点的检测，并通过自适应地学习特征权重来提高性能。DeTone 等人[19]设计了一种使用两个卷积神经网络的方法。其中，第一网络用于检测图像中的角点，第二网络用于计算两幅图像之间的单应矩阵。Sarlin 等人[20]针对复杂的室内定位环境提出了一种基于注意力聚合的图神经网络 SuperGlue，并利用最优传输模型进行匹配优化，实现了在室内和室外两种环境下的位姿估计。Tang 等人[21,22]将卷积网络与递归网络结合，提出了几何对应网络（Geometric Correspondence Network，GCN）由特征提取器和对应预测器组成，分别用于从输入图像中提取特征和预测不同图像之间对应点的位置，从而提高位姿估计的准确性。此外，语义分割网络也经常应用于视觉 SLAM 系统中。Bowman 等人[23]使用极大似然估计将语义 SLAM 转换为概率问题，并利用目标检测将尺度信息和语义信息融合，从而在相机进行位姿估计和构建环境地图时提高精度。Kaneko 等人[24]提出了利用语义分割网络生成的掩码的框架，以消除不稳定特征点，利用剩余的特征点进行稳定的位姿估计。Chao 等人[25]使用 SegNet[26]网络进行图像的像素级语义分割，并结合运动一致性检查以消除掉场景的动态部分。类似地，Bescos 等人[27]提出了 DynaSLAM，它通过多视图几何和 Mask R-CNN[28]在复杂场景中更好地检测和跟踪动态对象，但计算量较高，不能在资源受限的设备上实时运行。Masoud 等人[29]提出了将依据特征的 SLAM 与多目标跟踪（Multi-Target Tracking，MTT）结合，并使用 Fast R-CNN[30]来检测并区分移动的和静止物体。但该算法只能检测两类目标，并不能很好地处理复杂的动态场景。

1.2.3　动态 SLAM 算法

许多不同的方法被提出来解决视觉 SLAM 中的动态问题[31]。Sun 等人[32]通过自我运动补偿粗略地过滤掉运动物体的影响，然后对矢量量化深度图像应用最大后验估计来精确定位前景。Wang 等人[33]利用点轨迹对图像进行聚类，并利用能量函数最小化来排除动态目标，该方法是稳健的但不能实时执行。Du 等人[34]提出一种基于条件随机场和长期观测一致性的算法，实现了动态场景下的精确位姿估计。即通过图分割的方法筛选动态和静态特征点，并在静态特征点上进行初始位姿估计，然后利用条件随机场在三维环境中检测动态路标点，并进一步在静态路标点上实现精确的位姿估计。Li 等人[35]提出的 DP-SLAM 是一种通过贝叶斯概率估计来

跟踪动态匹配点，克服了几何约束和语义分割框架的偏差。Ferrera 等人[36]提出了针对动态水下环境的实时单目视觉 SLAM，利用 RANSAC 来计算基本匹配矩阵，并消除与外极几何不一致的异常值，但该系统缺乏环闭合机制，不能被认为是一个完整的视觉 SLAM 方法。Liu 等人[37]使用滑动窗口模型在两个不连续帧之间进行特征匹配，并使用网格的运动统计（Grid-based Motion Statistics，GMS）[38]过滤异常值。其它方法利用外部传感器融合，如惯性测量单元（Inertial Measurement Unit，IMU），通过估计相机的运动来解决这个问题[39,40]。Song 等人[41]将 RGB-D 传感器与 IMU 结合起来，以估计高度动态环境下的相机姿态。将 IMU 信息作为一种先验信息来约束相机的姿态估计，消除视觉信息误差。除了这些处理刚性场景的方法外，Agudo 等人[42,43]还提供了一种处理结合卡尔曼滤波的非刚性对象的解决方案，用 Navier-Stokes 方程来表示物体的表面力学，并用有限元法（Finite Element Method，FEM）进行求解。然后，该算法将方程嵌入到一个扩展的卡尔曼滤波器中，从而得到一个序列贝叶斯优化框架。

1.3 研究内容

论文围绕融合深度学习的视觉 SLAM 关键技术展开研究工作，针对在复杂环境下人工设计的特征无法有效地提取和匹配问题，尝试融合不同特征并引入深度学习技术进行替换，从而提高视觉 SLAM 系统的性能。论文在点线特征融合、深度学习进行特征提取匹配和动态环境三个视觉 SLAM 子问题进行了研究，并取得了一定的成果。具体内容主要包括：

针对复杂工业场景下由于纹理单一、几何结构复杂等原因导致的位姿估计精度低、鲁棒性差等问题，提出了面向点线感知的异构图注意力视觉 SLAM 系统。首先，构建了同步点线特征提取网络，并通过感知迁移的知识蒸馏策略对网络模型优化，进一步提高系统实时性。其次，为了提升图像中点线匹配精度，提出了一种点线异构图注意力网络，利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间学习，提高几何特征匹配结果的准确率。最后，将点线匹配问题转换为最优传输问题，构建依据贪婪策略的最邻近点迭代方法对该优化问题进行求解。

由于在动态环境中现有的视觉 SLAM 方法容易失败，论文提出了一种面向动态环境中的实时语义 SLAM 系统。提出通过两种不同策略来剔除动态特征点，实现了在低功耗嵌入式平台上实时运行的语义 SLAM 系统。

最后，为了验证所提 SLAM 系统在油气场站的有效性，利用 Unity3D 虚拟引擎开发了油气场站仿真系统。

1.4　论文结构与研究路线

1.4.1　论文结构

论文的具体结构安排如下：

第一章为绪论，主要阐述了视觉 SLAM 的研究背景及重要意义，对国内外视觉 SLAM 算法的研究进展进行了梳理和总结，并对论文主要内容、文章结构及研究路线进行了简要描述。

第二章旨在介绍视觉 SLAM 的理论基础，涵盖视觉里程计、闭环检测、后端优化和建图等相关技术。文章首先解释了视觉 SLAM 系统的基本概念，随后简要概述了主流视觉 SLAM 算法的流程和内容，同时对 SLAM 中常用的数学进行了概述。最后，该章节引出了论文需要解决的问题。

第三章介绍了面向点线感知的异构图注意力视觉 SLAM 系统。首先，利用点与点、点与直线的几何关系构建了同步点线特征提取网络；并通过感知迁移的知识蒸馏策略对网络模型优化，进一步提高系统实时性。其次，为了提升图像点线匹配精度，提出了一种点线异构图注意力网络，利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间学习，提高几何特征匹配结果的准确率。最后，将点线匹配问题转换为一个最优传输问题，构建结合贪婪策略的最邻近点迭代方法对该优化问题进行求解。

第四章介绍了融入深度学习的实时动态 SLAM 算法。针对在动态环境中的 SLAM 精度大幅下降的问题，设计了一个结合轻量级目标检测网络的并行语义线程，使算法能够更快地获得场景中的语义信息。与其它融合深度学习的视觉 SLAM 算法相比，该算法的实时性和性能得到了显著的提高。

第五章介绍了利用 Unity3D 虚拟引擎开发了一款油气场站仿真系统来解决油气场站难以进行实验的问题。该系统模拟了油气场站的真实环境，提供了高度逼真的模型和光照条件，可以实现天气情况设置和机器人的自动巡检。在本章中，描述了该系统的框架和功能设计，并对系统的主要功能进行了测试。

第六章的主要目的是对本文的研究的发现和内容进行综述，进而探讨未来研究的方向。

1.4.2 研究路线

本工作采用异构图注意力网络技术设计了视觉 SLAM 系统，研究内容包括相关资料的整理与分析、SLAM 系统设计、实验对比及总结分析。通过对国内外视觉 SLAM 算法及可视分析技术的概括与分析，完成了研究资料的收集与整理，为论文 SLAM 框架的设计提供了理论基础。在模型设计阶段，将视觉 SLAM 与图注意力网络相结合，以此来提高特征提取和匹配的精度；图 1.3 展示了本文的研究路线。

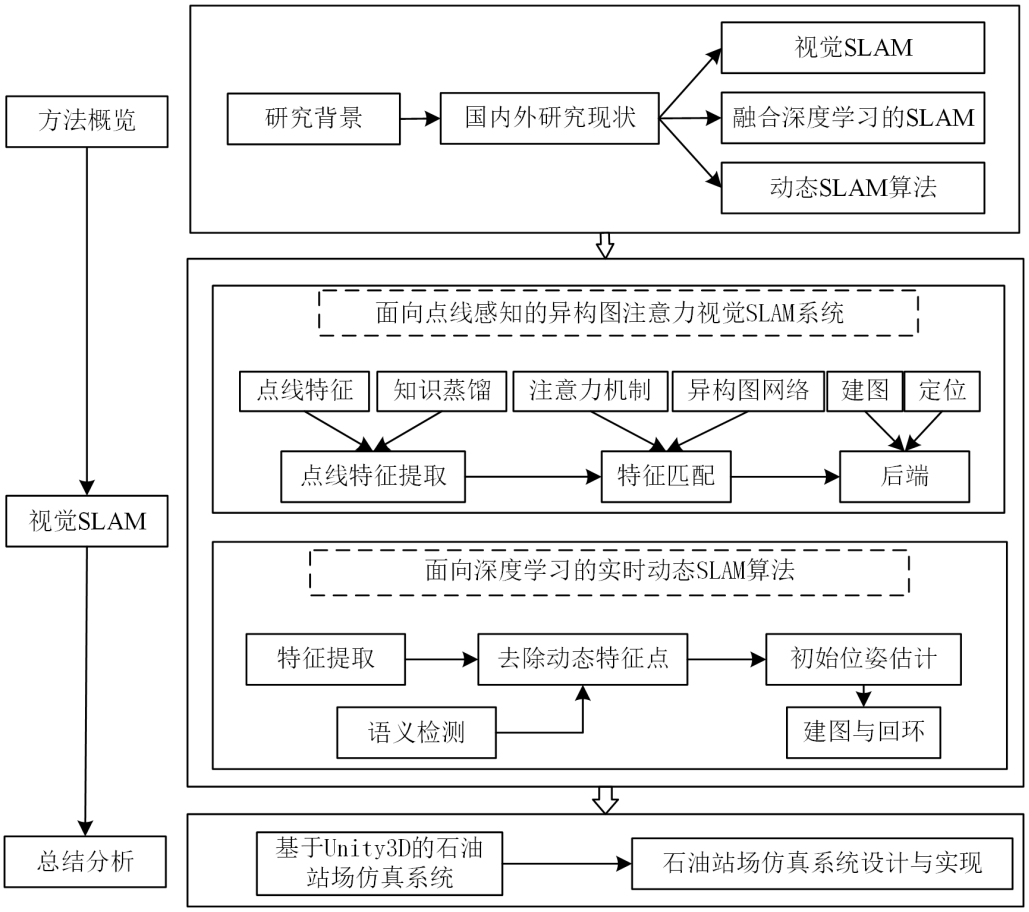


图 1.3 研究路线

Fig. 1.3 Research route

1.5 本章小结

本章首先概述了视觉 SLAM 的背景、意义和在工业场景中所面临的挑战。接着对国内外视觉 SLAM 技术、融合深度学习的视觉 SLAM 算法以及动态 SLAM 算法进行了整理和分析。最后，简要介绍了主要研究内容、论文结构和研究路线。

第 2 章　相关理论与技术

本章主要介绍视觉 SLAM 所涉及的相关技术。首先，介绍了相机投影模型；接着，概述了 SLAM 系统中的特征点提取和匹配方法，以及光流法；然后，对 SLAM 系统中的回环检测和地图构建方法进行了详细的阐述，包括如何检测回环以及利用回环信息更新地图；最后，本章讨论了 SLAM 系统中的前端和后端优化，分别介绍了基于滤波器的方法和基于非线性优化的方法。

2.1　相机模型

视觉 SLAM 系统利用搭载相机的机器人采集信息，实现对机器人位姿的估计和周围环境地图的构建。机器人在运动过程中，其运动轨迹可以使用运动方程描述，而相机采集到的图像像素数据与对应的三维环境中物体之间的关系可以使用观测方程表示。如图 2.1 所示，相机采样的时间戳为 $t_1, t_2, \cdots t_k$，对应机器人位姿用 $x_1, x_2, \cdots x_k$ 表示，$y_1, y_2, \cdots y_{\mathbf{N}}$ 代表当前场景附近的 **N** 个路标点。在 t_k 时刻，机器人的运动方程可以表示为：

$$x_k = f(x_{k-1}, u_k, w_k) \tag{2.1}$$

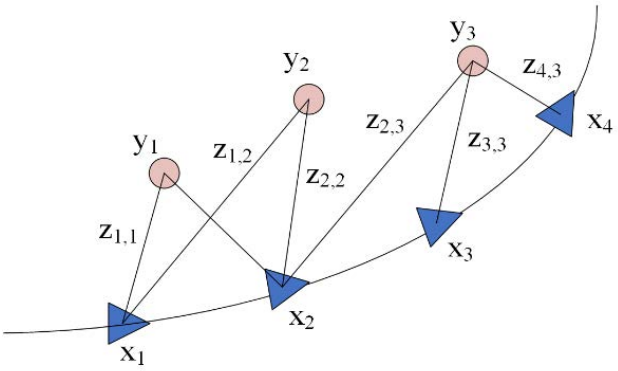


图 2.1 视觉 SLAM 原理图

Fig. 2.1 Schematic diagram of visual SLAM

观测方程表示为：

$$z_{k,j} = h(y_i, x_k, v_{k,j}) \tag{2.2}$$

其中，w_k和v_k为均值为 0 高斯分布的噪声项，定义在t_{k-1}时刻位于x_{k-1}处的机器人观测到的路标点y_i，u_k对应t_k时刻传感器采集的数据，$z_{k,j}$为时刻t_k对第i号路标点的观测数据。

相机位姿用于表示相机的位置和朝向，用(R,t)表示；其中，R为机器人的旋转矩阵，t为机器人的平移向量，目的是将相机坐标系中的信息转换到世界坐标系中。定义Y为环境中的一处路标，该点在平面图像上所投影的像素坐标为$z=[u,v]^T$。接下来介绍投影流程：首先，利用相机位姿(R,t)和三维环境中的路标点Y得到该点的平面像素坐标，其计算过程定义如下：

$$y' = RY + t = \left[X', Y', Z' \right]^T \tag{2.3}$$

由此该观察点在两个坐标系之间进行了配准，然后将路标点y'投影至相机的二维平面上来简化计算，对应的像素坐标可以定义为：

$$y_c = \left[u_c, v_c, 1 \right]^T = \left[X'/Z', Y'/Z', 1 \right]^T \tag{2.4}$$

最后，利用相机的参数和针孔相机成像原理，得到对应二维平面上的像素坐标：

$$u_s = f_x u_c + c_x \tag{2.5}$$

$$v_s = f_y v_c + c_y \tag{2.6}$$

其中，f_x和f_y分别表示图像横轴和纵轴的缩放因子，c_x和c_y表示图像横轴和纵轴中心所对应的偏移值。因此，根据环境路标点和平面图像像素之间的相似三角形原理，计算观测方程的矩阵可以获得机器人的位移结果。

2.2 视觉里程计

前端视觉里程计（Visual Odometry，VO）是视觉 SLAM 系统中实现位姿估计的关键组成部分。VO 利用相机采集到的真实世界图像进行相机位姿的计算并进行定位。VO 的计算过程主要包括特征提取、特征匹配、运动估计和局部优化几个步骤。相邻图像帧之间的位置变化关系是计算位姿的主要参考，因此 VO 利用像素点的匹配来计算相机位姿和运动轨迹。与人类的环境感知能力类似，VO 依据相邻图

像帧之间的像素点匹配来获取机器人的运动轨迹路线。通过计算相邻两帧的相机位姿，VO 能够进一步得到机器人相对于静止物体的运动位移，并实现对机器人的定位。在静态环境当中时，视觉里程计能够进行更精确的位姿估计。相对而言，在动态环境当中存在大量的动态物体，这将令视觉里程计相对运动的计算产生强烈干扰，导致定位精度急剧下降。

2.2.1　光流法

光流法是一种直接法的视觉里程计方法，不需要提取特征点，而是直接利用像素级别的信息来计算相邻图像帧间的运动。其基本假设是，相邻帧之间的像素亮度不变，即一个像素在两个相邻帧中的灰度值保持不变。具体来说，光流法会对图像中的每个像素点进行追踪，并计算它在相邻帧中的像素位置的变化量。这些位置变化量构成了像素的“光流向量”，即表示像素运动方向和速度的向量。假设相邻的两帧图像中像素的亮度相同，并且帧之间的时间差很短，则灰度是恒定的，可得如下式：

$$P\left(i,\ v,\ z\right)=P\left(i+di,\ v+dv,\ z+dz\right) \tag{2.7}$$

其中，z 和 $z+dz$ 分别代表相邻两帧的时刻，$P\left(i,\ v,\ z\right)$ 和 $P\left(i+di,\ v+dv,\ z+dz\right)$ 表示对应时刻特征点在二维平面图像的像素坐标，如果相邻图像帧之间只有很小的时间间隔时，可以利用泰勒级数展开得到如下公式：

$$P\left(i+di,\ v+dv,\ z+dz\right)\approx P\left(i,\ v,\ z\right)+\frac{\partial P}{\partial x}\partial x+\frac{\partial P}{\partial y}\partial y+\frac{\partial P}{\partial t}\partial t \tag{2.8}$$

将式 2.7 代入式 2.8 并除以 ∂t 得到：

$$\frac{\partial P}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial P}{\partial y}\frac{\partial y}{\partial t}=-\frac{\partial P}{\partial t} \tag{2.9}$$

其中，$\frac{\partial x}{\partial t}$ 和 $\frac{\partial y}{\partial t}$ 分别表示 x 轴和 y 轴上的速度，定义为 v 和 z。将 $\frac{\partial P}{\partial x}$ 和 $\frac{\partial P}{\partial y}$ 用 I_i 和 I_j 表示，并定义灰度随时间的变换为 I_f，公式 2.9 可以写成更简洁的矩阵形式：

$$\left[I_i,I_j\right]\begin{bmatrix}u\\v\end{bmatrix}=-I_f \tag{2.10}$$

最后，通过计算相邻两帧图像中同一个像素点在 x 和 y 方向上的灰度值变化，可以得到像素点在图像平面内的运动速度 u，v。这个计算过程使用最小二乘法来求解，即通过最小化灰度误差的平方和来估计像素的运动速度。以上过程定义为：

$$\mathbf{A}=\begin{bmatrix}\left[I_x,I_y\right]_1\\ \vdots\\ \left[I_x,I_y\right]_m\end{bmatrix},\quad \mathbf{b}=\begin{bmatrix}I_{t1}\\ \vdots\\ I_{tm}\end{bmatrix},\quad m=1,\ldots,\omega^2 \tag{2.11}$$

$$[u]_v^*=-\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{2.12}$$

2.2.2 特征点法

图像特征点是在图像中具有独特性、可重复性且易于提取的像素点，通常包括角点、边缘和斑点等。其中，角点是图像中像素灰度变化显著的点，如图像中的拐角处，可以通过计算像素点周围的梯度信息来检测角点。边缘是图像中像素灰度变化明显的区域，可以通过计算像素点周围的梯度信息来检测边缘。而斑点是图像中局部区域内像素灰度值变化显著的区域，例如图像中的纹理区域。选择哪种特征点作为图像特征点取决于具体的应用场景和要求。为了提高特征点的鲁棒性和准确性，研究人员提出了一系列的图像特征点算法。

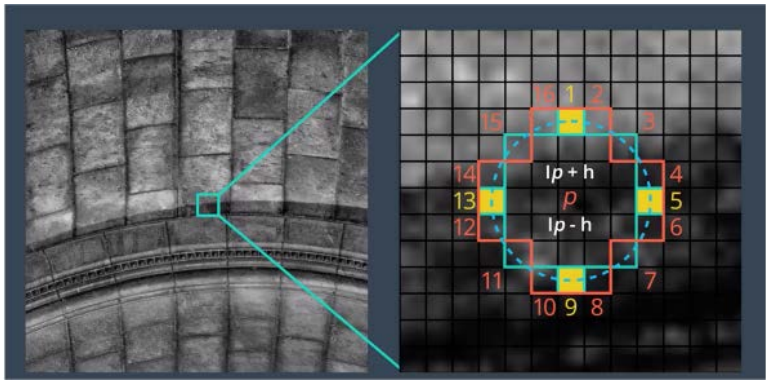


图 2.2 ORB 特征原理图

Fig. 2.2 Schematic diagram of ORB characteristics

SIFT（Scale Invariant Feature Transform）[44]是一种能够适应尺度、光照、旋转等条件改变的特征点算法。该算法通过对图像进行高斯模糊处理，提取图像的尺度

空间特征，并利用 DoG（Difference of Gaussian）算子在不同尺度空间上进行极值点检测。最后，通过方向直方图计算特征点的方向，构建特征描述子。FAST（Features from Accelerated Segment Test）[45]是一种采用像素点的灰度阈值快速检测的算法。该算法通过比较像素点与周围像素的灰度值差异，检测出具有明显变化的像素点作为特征点。由于该算法在计算中采用了加速策略，因此可以快速地检测出图像中的特征点。如图 2.2 所示，ORB（Oriented FAST and Rotated BRIEF）[17]特征结合了 FAST 特征点的快速计算能力和 BRIEF 特征描述子的旋转和尺度不变性。该算法构建具有旋转不变性的描述子，适用于在各种条件下的目标检测和跟踪。特征点的描述子是一种用于表示图像中特定位置的局部特征的向量或矩阵。描述子通常基于特征点周围的像素值计算，以捕获该点周围的纹理和形状信息，可以通过计算两个向量之间的距离来判断它们的相似程度。

2.2.3　特征匹配

特征匹配是指在两幅图像中找到一些具有唯一性、可重复性和区分性的特征点，并将它们在两幅图像中进行对应。在特征点匹配的过程中，需要计算两个图像中每个特征点的描述子，并比较它们之间的距离，从而找到最佳匹配。当特征点数量过多时，使用暴力匹配算法进行配对将会消耗大量的运算时间，因此需要采用更加适应的算法来提高匹配的速度和效率。一种常用的算法是快速近似最近邻算法（Fast Library for Approximate Nearest Neighbors，FLANN）[46]，它可以在保证精度的前提下，大幅度降低匹配过程的计算成本。FLANN 算法利用空间索引结构来加速最近邻搜索，将原本需要遍历所有特征点的匹配过程，转化为在空间索引结构上进行快速搜索的过程，大大提高了匹配的效率。但仍需要调整匹配参数以获得更好的匹配效果。其中，外点过滤是一种常用的技术，用于排除在特征匹配过程中产生的不正确匹配的特征点对，从而提高匹配的精度。外点过滤的实现方式包括计算最优次优比[47]、随机一致性采样（Random Sample Consensus，RANSAC）[48]、交叉验证以及启发式方法[49]，可以根据具体应用场景选择不同的方法。

深度学习在特征匹配上的进展集中于对特征匹配进行过滤，通过将匹配结果划分为内点或者外点来进行操作[50-55]。这些工作都是对通过最近邻搜索获得的匹配集进行操作，因此丢失了匹配过程中的结构信息。此外，特征匹配可以被视为一个图匹配的问题，Sun 等人[56]使用浅层的神经网络将图匹配由一个二次分配的 NP−难问题近似为一个线性指派问题。

图 2.3 特征点匹配图

Fig. 2.3 Feature point matching map

2.2.4 运动估计

通过特征点的匹配结果，运动估计则可以由相邻图像帧之间对应关键点的位移计算出。根据匹配特征的不同维度类型，可以将常用的算法分为三种：2D-to-2D、3D-to-2D 和 3D-to-3D。

2D-to-2D 的对极几何算法通过对两幅图像中对应的特征点进行匹配，利用极线约束来计算相邻帧之间的基础矩阵或本质矩阵，进而恢复相机的相对位姿。该方法的优点在于只需要像素坐标作为先验信息，适用于单目相机或者双目相机中的其中一枚相机的情况，但是由于只利用了图像中的二维信息，可能会出现深度歧义问题，需要结合其它算法进行优化。

3D-to-2D 的多点透视成像（Perspective-n-Point，PnP）[57]算法，利用三维路标点和对应的二维像素点作为先验信息，求解相机的位姿，从而实现运动估计。该方法需要先对三维点进行建模，需要已知三维路标点信息和对应的二维像素点信息，适用于稠密 3D 点云或者已知空间中有特定路标点的情况。

3D-to-3D 的迭代最近点（Iterative Closest Point，ICP）[58]算法，通过将两个三维点云进行配准，得到它们之间的相对位姿，从而实现运动估计。该方法需要先获取空间中的 3D 点云信息，并且需要相机和物体之间存在较明显的运动，适用于相机与物体之间的配准和运动估计。

2.2.5 局部优化

局部优化（Local Optimization）是在全局位姿优化的基础上，对当前状态的一部分进行进一步优化，以提高位姿的精度和鲁棒性。其基本思路是通过优化一个局部窗口内的相机位姿和三维点的位置，从而使得重投影误差最小化。具体来说，在

局部优化过程中，首先根据当前状态的一部分（例如相邻时刻之间的状态）和其它已知的信息（例如相机内参和路标点的位置），计算每个路标点在相机坐标系下的三维位置。然后，根据当前相机位姿对路标点进行投影，得到每个路标点的预测像素坐标；接下来，通过比较预测像素坐标和实际观测到的像素坐标之间的差距，得到重投影误差。最后，使用非线性最小二乘（Nonlinear Least Squares，NLS）优化方法来最小化重投影误差，并更新相机位姿和环境中路标点的观测位置。由于 NLS 优化是一个迭代过程，所以需要指定初始值。在 SLAM 中，通常使用全局优化得到的位姿作为初始值，然后使用局部优化来进一步提高位姿精度和鲁棒性。

总的来说，局部优化是 SLAM 中的一个重要环节，能够提高系统的鲁棒性和位姿精度，从而提高 SLAM 系统的可靠性和适用范围。

2.3　闭环检测

在视觉 SLAM 系统的前端视觉里程计通过估计相邻图像帧间的相机位姿，并积累这些位姿变化来恢复相机的运动轨迹。然而，由于位姿估计误差的存在，随着系统不断运行，绝对位姿的计算会具有更大的累计误差，最终使系统整体失效。为了解决这一问题，可以引入闭环检测算法。闭环检测算法通过检测相机经过的相同位置或靠近的位置，并优化位姿图中的相机位姿来消除漂移误差，进而提高定位系统的精度和鲁棒性。闭环检测模块一般包括以下步骤：首先，对于每个新的图像帧，从之前的帧序列中选择一些候选帧作为可能的闭环帧。其次，对于每个候选帧，通过特征匹配和几何验证来确定是否存在闭环。最后，将确定的闭环帧与之前的位姿图进行优化，以消除漂移误差并提高系统的精度和鲁棒性。图 2.4 展示了闭环检测闭环优化实例图。因此，闭环检测模块在 SLAM 系统中扮演着至关重要的角色。

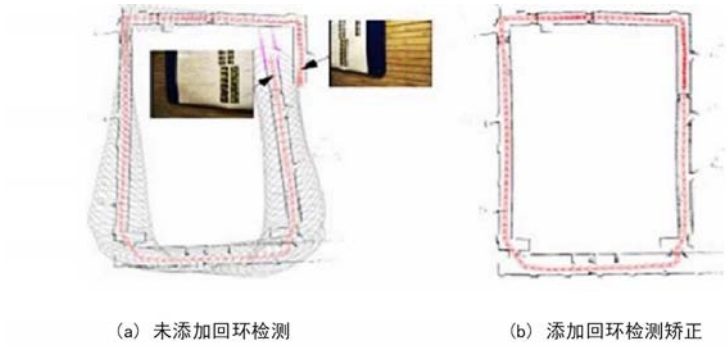


图 2.4 闭环检测对比图

Fig. 2.4 Comparison diagram of closed loop detection

目前，有基于词袋的方法[8,59]和基于特征匹配的方法来进行回环检测。基于词袋的算法是在获得数据后首先对图像进行预处理，并且对图像中的所有特征描述符加以聚合，从而生成字典。接着对于后续读入的图像帧，再通过最近邻算法搜索该帧图像中所对应于词袋中的所有视觉词汇，将其特征描述成适当的特征向量；进而通过对比图像和经过处理后图像中的特征向量的相似程度，确定是否有回环发生。由于词袋模型需要对图像特征进行聚类，因此计算更加复杂和耗时。

依据特征匹配的主要方法是对当前图像和经过处理后的图像的内容进行特征匹配，在匹配后根据特征相似程度来确定是否存在回环。若两帧关键帧图像的匹配特征点数量超过了一定阈值时，则可看作相机已经返回了其曾经到达过的区域，从而有了回环。然而，每产生一帧图像时，都必须把图像的特征点与历史中的所有图像特征进行对应，这将要求高性能的计算机算力。为处理这些问题，系统中引进了词袋模型的概念，将特征当作单词，并通过预训练产生一个涵盖了各种特征类型的词典，再针对图像特征产生相应的单词到字典中得到一个词袋模型[60]。这样，当评估图像的特征相似度时，只需比较它们的单词袋，从而提高了回环检测的速度。

2.4 后端优化

SLAM 的后端优化旨在对前端 VO 采集到的数据进行全局一致性的优化，从而进一步提升系统的精度和稳定性。相对于前端，后端的优化问题更加复杂，主要包括两个方面：优化问题建立和优化问题求解。在优化问题建立方面，需要通过对前端采集到的数据进行建图，构建一个由相机位姿和路标点三维坐标组成的稠密图，同时，将前后两帧之间的相对位姿作为边连接，将路标点与它们被观测到的相机位姿作为节点连接。在建图时，需要对路标点的深度信息进行估计，这通常通过三角化或深度传感器得到。优化算法求解需要将所有的边和节点的误差加权并汇总成一个目标函数，然后通过迭代方式逐步调整节点和边的参数，使得目标函数达到最小值。常用的两种 SLAM 后端优化方法分为滤波器方法和非线性优化方法。

滤波器方法主要包括扩展卡尔曼滤波器（Extended Kalman Filter，EKF）和粒子滤波器（Extended Kalman Filter，PF）。EKF 通过线性化状态转移和测量模型，将 SLAM 问题转化为一个线性的系统，然后采用卡尔曼滤波器进行优化。PF 则采用一组随机样本来近似概率分布，通过重要性采样和重采样来更新样本，最终得到滤波器的估计结果。这些方法通常具有较高的实时性和可扩展性，但对于非线性问题的优化效果有限。

非线性优化方法则是通过最小化误差函数来对估计结果进行优化。这些方法通常包括非线性最小二乘法（NLS）、高斯-牛顿（GN）算法和 Levenberg-Marquardt（LM）算法等，这些方法对于非线性问题有更好的优化效果，但计算复杂度较高，需要较高的计算资源和存储容量。并且对于初始位姿估计的准确性和数据关联的鲁棒性也更为敏感。对应误差估计表示如下：

$$e = z - h(\xi, p) \tag{2.13}$$

其中，h 定义为观察函数，p 代表观测到的真实空间下特征点的集合，李代数 ξ 表示为当前相机位置和方向。$z = [u_s, v_s]^T$ 则代表特征点的集合在二维平面图像下的坐标。整体的光束平差（Bundle Adjustment，BA）代价函数通常是非线性优化的 SLAM 后端中使用的一种方法。该代价函数包括所有时刻的状态信息和它们之间的关系，可以被描述为一个非线性优化问题，目标是最小化观测误差和状态变量之间的矛盾。整体的 BA 代价函数可以用以下形式表示：

$$\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|e_{ij}\right\|^2 = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|z_{ij} - h(\xi_j, p_j)\right\|^2 \tag{2.14}$$

其中，z_{ij} 表示二维平面图像下的真实坐标，$h(\bullet)$ 表示相机的观测坐标。最小化该 BA 代价可以减小相机的位姿估计的误差。后端优化是整个 SLAM 系统的关键环节，直接影响到 SLAM 系统的精度和稳定性。为了在实际应用中保证算法性能，需要根据不同场景选择合适的优化算法。

2.5　章节小结

本章详细介绍了 SLAM 系统的相关技术知识。首先，介绍了 SLAM 系统的相机模型和运动描述，其中涉及到了相机的内外参数。其次，阐述了 SLAM 系统中的特征点提取和匹配方法，以及光流法。接着，介绍了 SLAM 系统中的回环检测和地图构建方法，包括如何检测回环以及如何利用回环信息更新地图。最后，讨论了 SLAM 系统中的前端和后端优化方法，分别介绍了滤波器方法和非线性优化方法，特别是对后端优化中的 BA 算法进行了详细的介绍。这些内容为理解和应用 SLAM 系统提供了基础知识和技术支持。

第 3 章　面向点线感知的异构图注意力视觉 SLAM 系统

针对复杂工业场景下由于纹理单一、几何结构复杂等原因导致的位姿估计精度低、鲁棒性差等问题，提出了面向点线感知的异构图注意力视觉同步定位与地图构建（SLAM）系统。首先，利用点与点、点与直线的几何关系构建了同步特征提取网络，分别利用点线感知注意模块引导网络获取关键区域几何关联特征；并通过感知迁移的知识蒸馏策略对网络模型优化，进一步提高系统实时性。其次，为了提升图像点线匹配精度，提出了一种点线异构图注意力网络，利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间学习，提高几何特征匹配结果的准确率。最后，将点线匹配问题转换为一个最优传输问题，提出构建贪婪策略的最邻近点迭代方法（Greedy Inexact Proximal Point Method for Optimal Transport，GIPOT）对该优化问题进行求解。通过在 KITTI 数据集及多种生产场景进行实验，结果验证了所提算法的有效性。

3.1　问题与分析

视觉 SLAM 可以分为依据点特征[8,14,65]、线特征[66-68]、面特征[68-71]以及上述特征的组合[72-74]。若场景中存在遮挡、光照、变形等情况时，现有算法在复杂工业场景下的自主导航任务中难以取得良好效果。

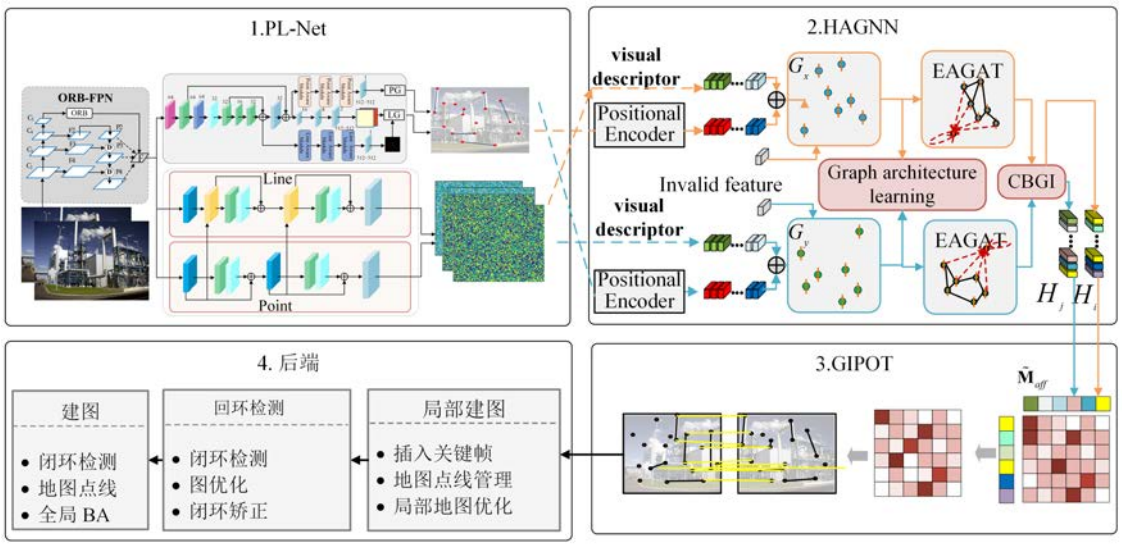


图 3.1 SLAM 系统框图

Fig. 3.1 System overview of SLAM

本章提出 SLAM 系统框架如图 3.1 所示。首先将新图像输入到 PL-Net 网络检测关键点和线段，并得到相应的描述子。然后将两幅图像点和直线特征传递到点线异构图注意力网络（Heterogeneous Attention Graph Neural Network，HAGNN）得到增强后的特征，将其输入到 GIPOT 中得到点线匹配结果，进而计算当前帧的位姿。最后经过 SLAM 后端对相邻帧之间的相对位姿和重投影误差对地图进行优化，并对相机的位姿进行调整，以得到更加准确的地图和相机位姿信息，从而完成建图。

3.2　点线特征提取网络

点线特征提取网络 PL−Net 如图 3.2 所示。首先，利用 ORB−FPN（ORB−Feature Pyramid Network）模块提取图像各层特征，然后，将这些特征通过点感知模块和卷积层处理，并与混合卷积模块桥接后进行上采样生成特征图。接着，利用点生成模块（Point Generate，PG）完成关键点提取。另一方面，通过线段检测模块的分支生成中心点图和位移图。最后，通过卷积与上采样操作生成点线描述子。

3.2.1　ORB-FPN 模块

如图 3.2 所示，这里根据 Nesterov 加速梯度下降算法（Nesterov Acceleration Gradient，NAG）设计了优化残差模块（Optimized Residual Block，ORB），去增强目标特征的表达能力[75]，推导过程如下：

$$y_{k+1} = x_k + \beta\left(x_k - x_{k-1}\right) \tag{3.1}$$

$$x_{k+1} = y_{k+1} - \alpha\nabla f\left(y_{k+1}\right) \tag{3.2}$$

其中，x_k、y_{k+1}分别表示网络第k层的输出和输入，a表示可学习参数，β表示权重，f 函数具有连续可导的性质，$\nabla f\left(y_{k+1}\right)$表示该公式 f 在点y_{k+1}的一阶梯度信息。当动量参数$\beta = 0$时，NAG 算法等价于标准梯度下降算法；当$\beta > 0$时，通过将优化a和β参数相结合，可以实现可以加速优化过程。在网络的前向传播过程中，设输入数据经过输入层与第一个隐藏层的计算过程为：

$$L_{i+1} = \sigma\left(U_i L_i\right) \tag{3.3}$$

其中，L_i表示第i层的特征向量，σ为该层的激活函数。设U为正定矩阵，令$V = \sqrt{U}$，$\mu = VL$，存在函数$g\left(\mu\right)$，当满足$g'\left(\mu\right) = \sigma\left(\mu\right)$时有：

$$\nabla\sum_i g\left(V_j^T\mu\right)=U\sigma\left(U^TL\right)=U\sigma\left(UL\right) \tag{3.4}$$

$f\left(\mu\right)$表示为：

$$f\left(\mu\right)=\frac{\left\|\mu^2\right\|}{2}-\sum_i g\left(V_j^T\mu\right) \tag{3.5}$$

其中，V_i为第i层的特征向量V，则$f\left(\mu\right)$进行的一阶导表示为：

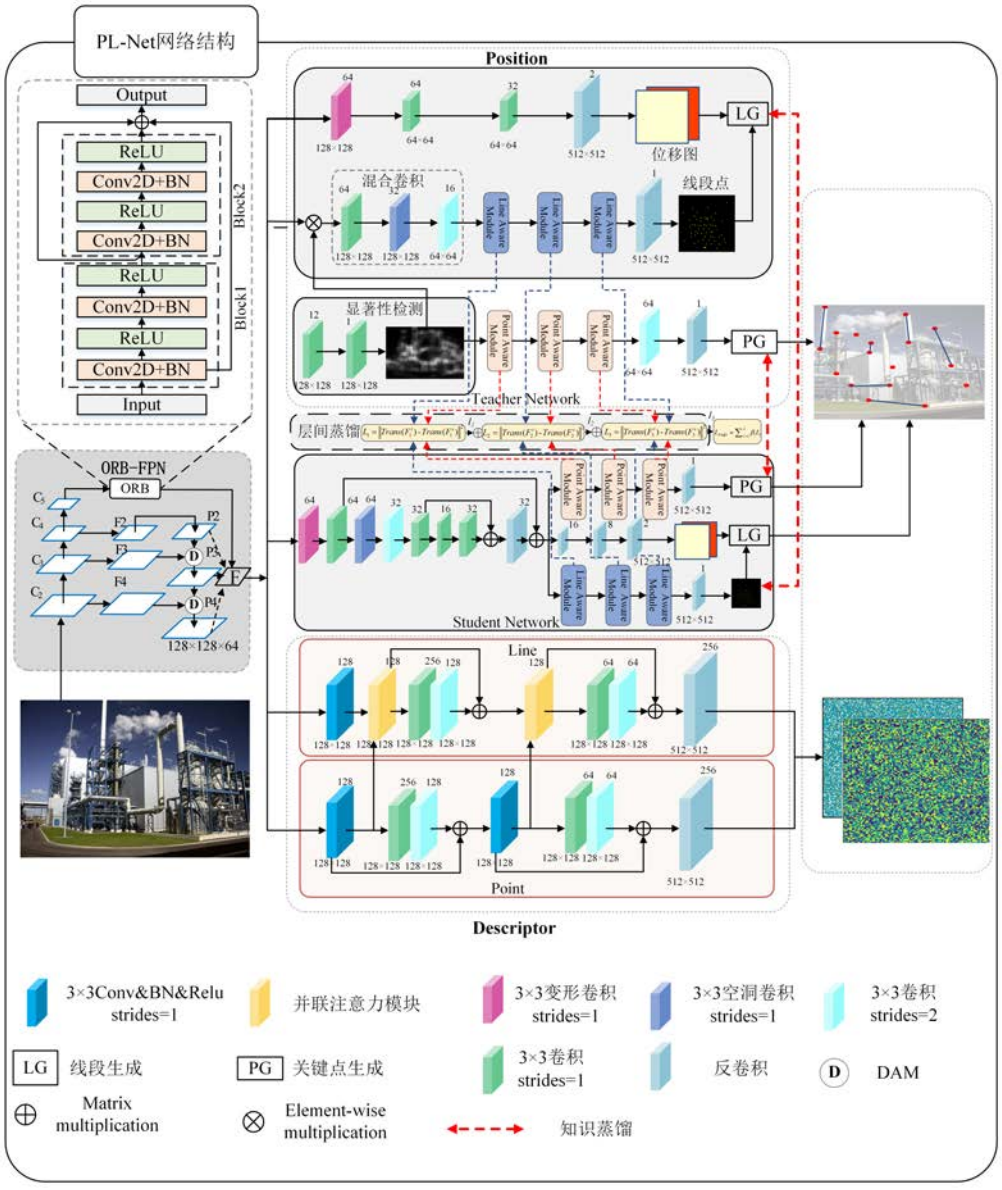


图 3.2 PL-Net 整体结构

Fig. 3.2 The overall structure of PL-Net

$$\nabla f\left(\mu_i\right)=\beta_i-V\sigma\left(V\mu_i\right) \tag{3.6}$$

式 3.2 可以表示为：

$$\mu_{i+1}=\mu_i+\beta\left(\mu_i-\mu_{i-1}\right)-\alpha\left(\left(1+\beta\right)\nabla f\left(\mu_i\right)-\beta\nabla f\left(\mu_{i-1}\right)\right) \tag{3.7}$$

由 $L=V^{-1}\mu$ 可以得到：

$$L_{i+1}=\left(\left(1+\beta\right)\left(1-\alpha\right)-\alpha\ \beta\right)L_i+\beta\left(1-\alpha\right)L_{i-1}+\alpha\left(1+\beta\right)\sigma\left(UL_i\right) \tag{3.8}$$

其中，$\sigma\left(UL_i\right)$ 表示为第i层前馈网络，ORB 模块结构如图 3.2 所示。

输入图像经过 ORB-FPN 模块，由下至上的经过各级编码器后，输出特征为 $\{C_2,C_3,C_4,C_5\}$，相对于输入图像的步幅为$\{4,8,16,32\}$。$\{F_2,F_3,F_4\}$是经过 1×1 卷积后得到具有相同的 128 维通道的特征，最后，通过 ORB 模块在C_5上利用更大的接受域提取更多的上下文信息。将提取出的上下文特征和生成的三个特征映射通过插值和最大池化进行元素级求和聚合得到特征 F 。

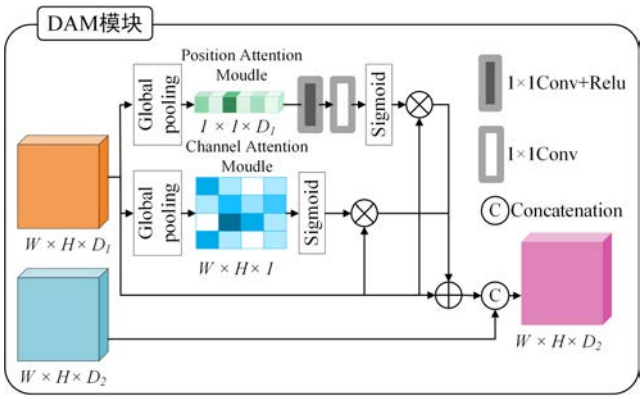


图 3.3 双路注意力模块

Fig. 3.3 Dual attention module

为了聚合 FPN 多尺度特征信息，采用双路注意力模块（Dual Attention Module，DAM）来执行特征的聚合。如图 3.3 所示，首先，为了获取特征的位置和通道信息，浅层特征图 $x\in\mathbb{R}^{W\times H\times D_1}$ 通过全局池化操作分别在空间和通道维度上进行压缩，生成位置向量 $x_p\in\mathbb{R}^{W\times H\times 1}$ 和通道向量 $x_c\in\mathbb{R}^{1\times 1\times D_1}$；然后，对位置向量 x_p 进行 Sigmoid 激活得到每个位置的权重，并与特征图 x 相乘，生成空间位置特征图 $F^p\in\mathbb{R}^{W\times H\times D_1}$，定义如下：

$$F^p = \sigma(x) \otimes x \tag{3.9}$$

其中，x 是浅层特征图，σ 是 Sigmoid 激活函数。

同样，对通道向量 x_c 依次进行 Relu 激活和 Sigmoid 激活的卷积，计算每个通道的权重，并与特征图 x 相乘，生成特征图 $F^c \in \mathbb{R}^{W \times H \times D_1}$，定义如下：

$$F^c = \sigma\left(\delta(W_2\left(W_1\left(gp\left(x\right)\right)\right)\right) \otimes x \tag{3.10}$$

其中，δ 是 Relu 激活函数，W_1 和 W_2 表示大小为 $1\times1\times D_1/16$ 和 $1\times1\times D_1$ 的卷积操作。

最后，为了增强特征的尺度，通过融合特征图 F^p、F^c 和 x'，得到最终的输出特征图 F，即：

$$F = \left[\left(x \oplus F^p \oplus F^c\right), x'\right] \tag{3.11}$$

其中，$\oplus$ 表示两个矩阵对应元素相加，[·] 为拼接操作。

3.2.2 关键点检测模块

如图 3.2 所示，关键点检测包含三个点感知模块和两个 3×3 卷积，步幅长度为 1，两个卷积层中间进行批归一化和 Tanh 激活函数运算，从而减小输出特征图通道数量。其次将其通过 Sigmoid 函数后生成像素值在 0 和 1 之间的显著性图。最后通过点生成模块（PG）利用非极大值抑制（Non−Maximum Suppression，NMS）方法生成关键点。

点感知模块（Point Aware Module）用于捕获关键点之间的关系。如图 3.4 所示，关键点提取分支嵌入了一个上下文增强模块，增强了特征表达能力。将特征 $x^p \in \mathbb{R}^{W \times H \times D}$ 作为输入，通过融合不同尺度的卷积特征，最终生成相同大小和相同通道的特征 $y^p \in \mathbb{R}^{W \times H \times D}$，上述过程的定义如下：

$$y^p = W_1\left[x^p, BN\left(W_1x^p\right), BN\left(W_2x^p\right), BN\left(W_3W_1x^p\right)\right] \tag{3.12}$$

其中，x^p 是输入特征，W_1，W_2 和 W_3 分别为大小为 1×1，3×3 和 5×5 的卷积操作，BN 为归一化，[·] 为拼接操作。

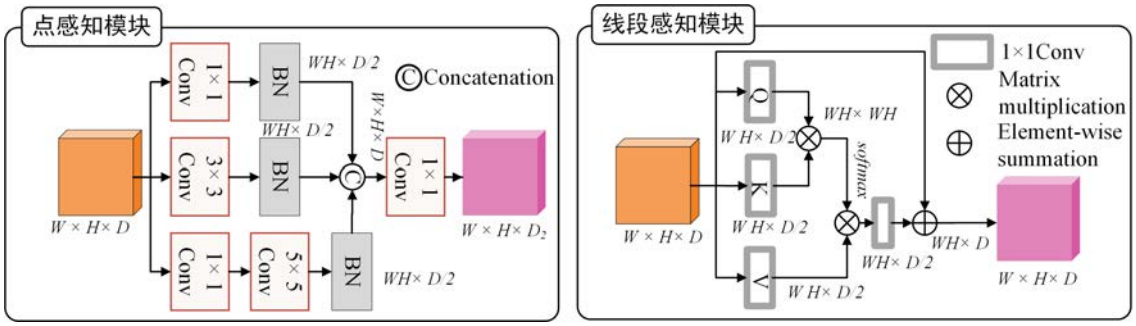


图 3.4　点线感知模块

Fig. 3.4 Point-Line aware module

3.2.3　线段检测模块

线段检测模块经过 ORB-FPN 模块后提取平面图像各层特征，然后将其输入线段提取模块生成具有两个对称端点的中点作为线段检测结果。线段中心点的提取[76]使用分类模型判断每个像素是否为线段中心点。由于线段形状窄而长，需要较大的接受域来对线段中心进行分类。因此，引入了一个混合卷积模块，通过叠加三个卷积层、一个 3×3 可变形卷积层和两个 3×3 空洞卷积层，在减少网络的参数的同时，增大网络的感受野。然后经过三个线段感知模块增强特征表示能力。最后通过一个反卷积层将输出映射的大小恢复到 512×512，输出图上线段中心点。线段提取分支的位移回归任务是预测端点相对于中点的角度和长度，由一个 3×3 可变形卷积层、两个 3×3 卷积层组成，其步幅均设为 1，通过输出映射按位置来索引相关的位移。最后使用 CAL[77]进行线段生成，线段的两个端点被定义如下：

$$\left(x_{l_s}, y_{l_s}\right)=\left(x_{l_c}, y_{l_c}\right)+\frac{\alpha}{2}(\cos\theta, \sin\theta) \tag{3.13}$$

$$\left(x_{l_e}, y_{l_e}\right)=\left(x_{l_c}, y_{l_c}\right)-\frac{\alpha}{2}(\cos\theta, \sin\theta) \tag{3.14}$$

其中，$\left(x_{l_c}, y_{l_c}\right)$为根节点坐标，$\alpha$为线段长度，$\theta$为旋转角度。

在线段检测中，这里提出了线感知模块（Line Aware Module）以有效地提取线形状特征。如图 3.4 所示，线段感知模块采用了改进的自注意力机制，以特征 $x^L \in \mathbb{R}^{W\times H\times D}$ 作为输入，通过融合自注意力特征和级联，最终输出特征 $y^L \in \mathbb{R}^{W\times H\times D}$，上述过程的定义如下：

$$y^L = x^L \oplus W_1\left(\alpha\left(W_q x^L \times W_k x^L\right)\times W_v x^L\right) \tag{3.15}$$

其中，x^L 是输入特征，W_1，W_q，W_k 和 W_v 为大小为1×1的卷积，⊕表示两个矩阵对应元素相加。

3.2.4 并联注意力模块

为了有效地融合关键点的描述子为线描述子增强特征，提出了并联注意力模块（Parallel Attention Module，PAB）。如图 3.5 所示，关键点描述子分支的输出特征包含线边缘信息，具有较强的关联性，为了提高线描述子的准确性，这里点描述子分支的输出特征通过应用一种轻量级注意机制对有用区域的特征分配更多的权重来增强对关键点特征的影响。点描述子分支的输出特征图表示为 $X_E \in \mathbb{R}^{C\times H\times W}$。$X_E$ 进行一维卷积得到空间注意图 $A_E \in \mathbb{R}^{C\times H\times W}$，边缘特征图 $X_E^S \in \mathbb{R}^{C\times H\times W}$ 计算为 $X_E^S = a(X_E \odot A_E) + X_E$。其中，$a$ 为可学习的参数，并初始化为 0。应用 CAEU 模块计算通道注意图，重新校准通道的权重。得到融合的特征图 $X_F^S \in \mathbb{R}^{C\times H\times W}$ 计算为 $X_F^S = X_E^S \otimes \delta\left(Conv1\times1\left(Conv1\times1\left(GAP\left(\mathbf{X}_E^S\right)\right)\right)\right)$。最后，将 X_T^S 和 X_F^S 级联在一起，得到 PAB 的最终输出 $X_F^{SC} \in \mathbb{R}^{2C\times H\times W}$。

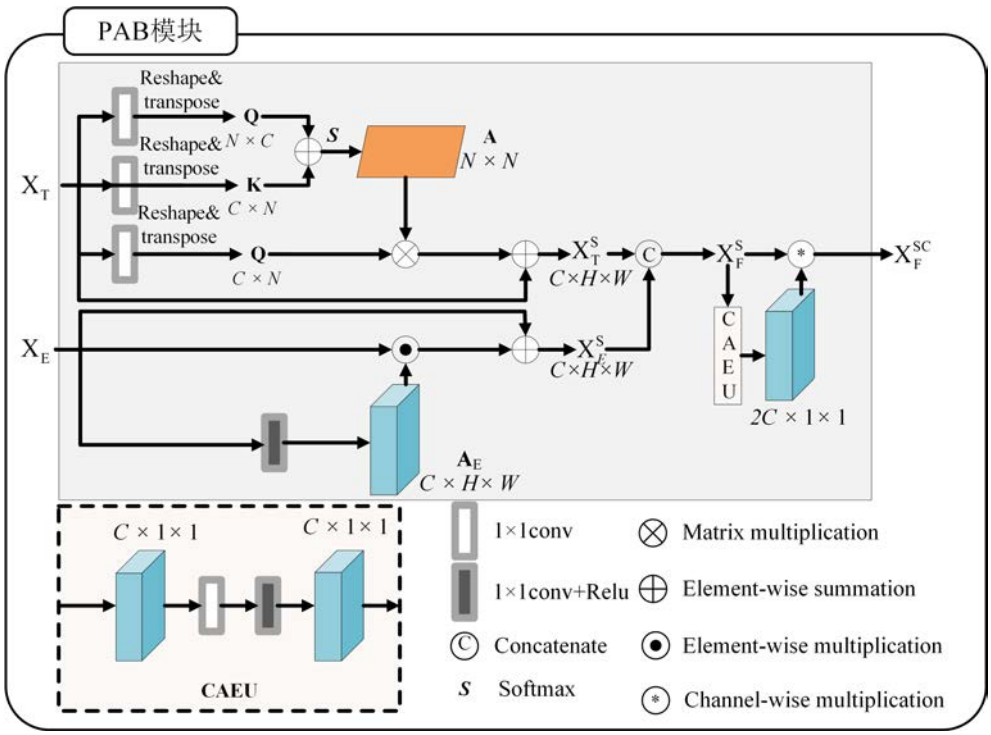


图 3.5 并联注意力模块

Fig. 3.5 Parallel attention block

3.2.5　网络输出蒸馏

为了减少由于引入注意力机制和点线感知模块而导致的计算量增加的问题，对点线检测模型 PL-Net 进行进一步模型压缩，同时，采用感知迁移方法将教师模型的信息转移到学生网络中，使用知识蒸馏（Knowledge Distillation，KD）[78,79]策略微调恢复模型精度。如图 3.2 所示，将关键点检测、线段中心点检测和线段回归三个不同任务模块通过知识蒸馏将教师网络与学生网络连接起来，用于指导学生特征提取网络的训练。在训练过程中，首先实现自适应加权多任务蒸馏，用 X_{tea}、Y_{tea} 和 Z_{tea} 分别表示教师模型的关键点、线段中心点和线段回归特征层输出，其学生模型对应特征层 X_{stu}、Y_{stu} 和 Z_{stu}，定义均方误差（Mean Squared Error，MSE）函数应用于多任务蒸馏，训练的蒸馏的损失函数为：

$$\begin{cases} L_p^S = \left\| X_{tea} - X_{stu} \right\|^2 \\ L_{root}^S = \left\| Y_{tea} - Y_{stu} \right\|^2 \\ L_{dis}^S = \left\| Z_{tea} - Z_{stu} \right\|^2 \end{cases} \tag{3.16}$$

其中，L_p^S、L_{root}^S 和 L_{dis}^S 分别表示关键点、线段中心点和线段回归任务的蒸馏损失。然后，采用的加权蒸馏损失定义为：

$$L_{MSE}^s = \sum_l \omega_l L_l^s \tag{3.17}$$

其中，L_{MSE}^s 多任务蒸馏损失，ω_l 表示验证损失的权重值。

3.2.6　层间知识蒸馏

除了蒸馏网络输出外，通过教师模型输出与学生模型输出之间进行层间知识蒸馏，以进一步增强“学生”网络。每个学生层的感知模块与相关的目标层感知模块进行关联，以进行知识迁移。多层知识蒸馏损失定义为：

$$L_{FMD} = \sum_{(s_l, t_l) \in C} Dist\left(Trans^t\left(F_{tl}^t\right), Trans^s\left(F_{sl}^s\right) \right) \tag{3.18}$$

进而得到整体损失为：

$$L_{total} = L_{MSE}^s + \beta L_{FMD} \tag{3.19}$$

其中，$Trans(\cdot)$ 表示通过注意力图将感知模块的特征映射转化为特定的手工表示，C 为对应的感知模块集合，F_{tl}^t 和 F_{sl}^s 分别为学生模型和教师模型的第 l 层的特征层，距离函数 $Dist(\cdot)$ 用于计算特征图之间的蒸馏损失，超参数 β 用于平衡两个损失项。

3.3 异构图注意力网络

异构图注意力网络（HAGNN）如图 3.6 所示，将待匹配的两张图像记为 I，I'，分别有 m，n 个特征，设 $d \in \mathbb{R}^D$ 为特征描述子，其中 D 为描述子维度。为了增强匹配的精度，利用注意力图神经网络对来整合其它上下文线索并且提高其特征表达能力。

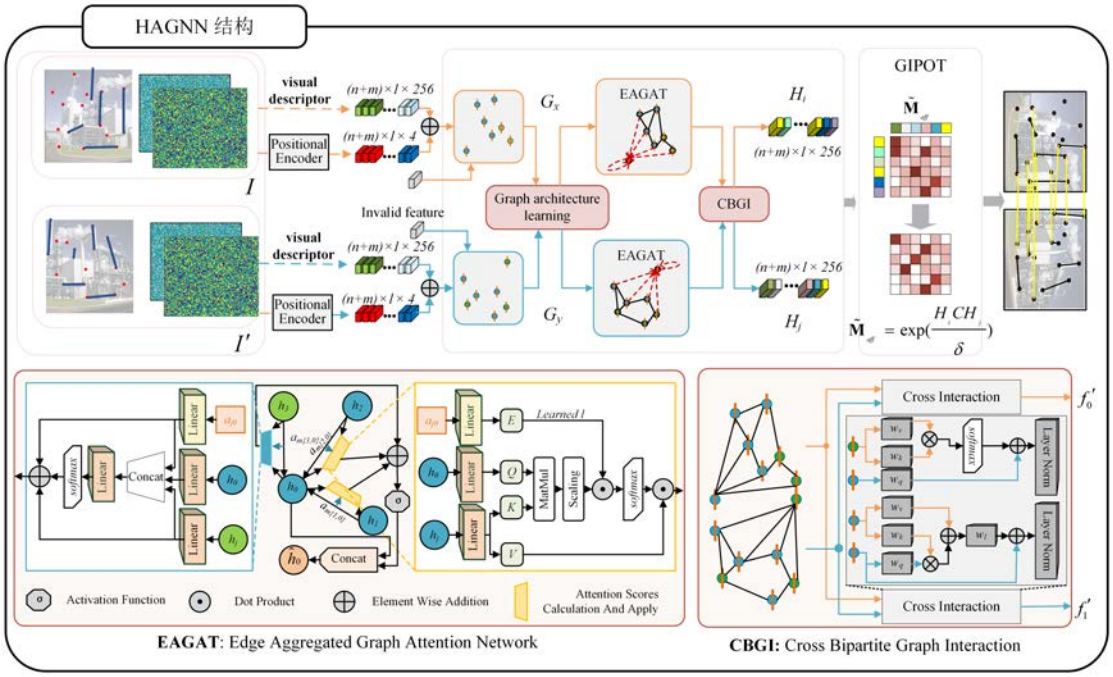


图 3.6 HAGNN 整体结构

Fig. 3.6 The overall structure of HAGNN

首先对于输入的两路特征分别使用位置编码器，通过将 $\widehat{F}_1$ 和 $\widehat{F}_2$ 中添加位置编码将关键点和线段坐标嵌入到一个高维向量中，即：

$$f_i = d_i + \mathbf{MLP}\left(P_i\right) \tag{3.20}$$

$$\mathbf{MLP}\left(P\right)_i = W_a \sigma\left(W_b P\right)_i \tag{3.21}$$

其中，P_i 为特征的位置，d_i 为描述子信息，$\sigma(\cdot)$ 为 Relu 激活函数。

3.3.1 边缘聚集图注意力模块

本章根据 GAT[80]提出了边缘聚集图注意力网络（Edge Aggregated Graph Attention Network，EAGAT），在聚合过程中利用边缘特征进行特征增强。为了充

分利用边缘特征的信息，将边缘特征与节点的特征采用两种聚合方式。对于相同性质的特征（点与点、线与线）采用自注意机制进行聚合。对于不同性质的特征（点与线）采用级联方式进行聚合。设图中顶点 v_i' 的特征为 f_i，则有：

$$\begin{aligned} f_i' = \sigma\Bigg(& \sum_{j\in\mathcal{N}_s} \mathrm{softmax}\left(\frac{W^a f_i W^\beta f_j}{\sqrt{d_k}} W^\gamma a_{ji}\right) W^\varepsilon f_j + \\ & \sum_{j\in\mathcal{N}_d} \mathrm{softmax}\left(\left[W^a f_i \,\|\, W^\beta f_j \,\|\, W^\gamma a_{ji}\right]\right) W^\varepsilon \Bigg) \Bigg\| f_i \end{aligned} \tag{3.22}$$

其中，W^a、W^β、W^γ和 $W^\varepsilon \in \mathbb{R}^{f_\eta' \times f_\eta}$ 表示权重参数，$\mathcal{N}_s$ 表示相同性质特征集合，$\mathcal{N}_d$ 表示不同性质特征集合，σ 为 Relu 激活函数。

3.3.2　交叉异构图迭代模块

针对图匹配中图间消息传递的亲和学习问题，提出点线异构图消息传递方法（Cross Heterogeneous Graph Interaction，CHGI）来交互关联增强节点特征。这里采用两种方式对边缘特征与节点的特征进行聚合。对于相同性质的节点（点与点、线与线）采用线性注意进行聚合，对于不同性质的节点（点与线）采用类似自注意机制的聚合方法，特征 $f_{si}' \in \mathbb{R}^{f_\eta'}$ 表示为：

$$\begin{aligned} f_{si}' = & \mathrm{LN}\left\{\sum_{j\in\mathcal{N}_{di}} \left(\mathrm{softmax}\left[\left(W_v f_j\right)\left(W_k f_j\right)^T\right] + W_q f_{si}\right)\right\} + \\ & \mathrm{LN}\left\{\sum_{j\in\mathcal{N}_{si}} \left(W_l\left(W_q f_{si}\right)\left(W_k f_j\right)^T + W_k f_j\right) + f_{si}\right\} \end{aligned} \tag{3.23}$$

其中，$v_{l/r} \in \mathbb{R}^{f_\eta'}$ 是两幅图的特征节点，$W_{(\cdot)}$ 为权重参数，LN 表示层归一化（Layer Normalization，LN）。

3.3.3　贪婪最邻近点迭代匹配模块

设图注意力网络的输出 H_i 为图 I 的特征集合，H_j 为图 I' 的特征集合，点线亲和矩阵 $G \in \mathbb{R}^{+N_1 \times N_2}$ 可以表示为：

$$G = f_{aff}\left(H_i, H_j'\right), \quad i \in v_1, j \in v_2 \tag{3.24}$$

其中，f_{aff} 为加权双线性函数，定义为：

$$f_{aff}\left(H_i, H_j'\right)=\exp\left(\frac{H_{\mathbf{i}}^T K H_{\mathbf{j}}^T}{\tau}\right) \tag{3.25}$$

其中，特征是一个 n 维向量，即 $\forall i \in v_1$， $j \in v_2$， $H_{\mathbf{i}}^T$ 和 $H_{\mathbf{j}}^T \in \mathbb{R}^{n\times n}$；$K \in \mathbb{R}^{n\times n}$ 为该亲和函数的权值矩阵，τ 是正则化参数。

在匹配过程中由于点线类型不一致，若直接融合可能会造成点线类型的误匹配。为此，引入单位分块对角矩阵赋予初始的耦合矩阵 $\boldsymbol{\Gamma}^{(1)}$，使点特征和线特征之间在迭代过程中失去匹配代价，从而提高匹配精度。初始耦合矩阵可表示为：

$$\boldsymbol{\Gamma}^{(1)} \leftarrow \begin{bmatrix} 11_p & 0 \\ 0 & 11_L \end{bmatrix} \tag{3.26}$$

其中，P 和 L 分别表示补全后点和线特征的数量。点线离散分布 Sinkhorn[82,83]距离定义为：

$$W_{\in}(u,v)=\min_{\boldsymbol{\Gamma}\in\sum(\boldsymbol{u},\boldsymbol{v})}\langle \boldsymbol{C},\boldsymbol{\Gamma}\rangle+\lambda h(\boldsymbol{\Gamma}) \tag{3.27}$$

其中，u，v 为概率向量，$W_{\in}(\boldsymbol{u},\boldsymbol{v})$ 是 u，v 之间的距离，矩阵 $\boldsymbol{C}=\left[c_{ij}\right]\in\mathbb{R}^{+n\times n}$ 为代价矩阵，c_{ij} 为 u_i 和 v_j 之间的距离，正则化项 $h(\boldsymbol{\Gamma})=\sum_{i,j}\boldsymbol{\Gamma}_{ij}\ln\boldsymbol{\Gamma}_{ij}$。

对于式 3.27，这里使用广义的最邻近点法[83]进行求解。首先，根据最邻近点迭代方法，定义 D_h 为 Bregman 散度：

$$D_h(x,y)=\sum_{i=1}^{n}x_i\ln\frac{x_i}{y_i}-\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}y_i \tag{3.28}$$

这里引入最邻近点迭代后，公式 3.27 可以改写为：

$$\boldsymbol{\Gamma}^{(t+1)}=\underset{\boldsymbol{\Gamma}\in\sum(u,v)}{\operatorname{argmin}}\langle \mathbf{C},\boldsymbol{\Gamma}\rangle+\beta^t D_h\left(\boldsymbol{\Gamma},\boldsymbol{\Gamma}^{(t)}\right) \tag{3.29}$$

然后，将 Bregman 散度 D_h 公式 3.28 代入公式 3.29 中可以得到：

$$\boldsymbol{\Gamma}^{(t+1)}=\underset{\boldsymbol{\Gamma}\in\sum(\boldsymbol{u},\boldsymbol{v})}{\operatorname{argmin}}\langle \boldsymbol{C}',\boldsymbol{\Gamma}\rangle+\beta^t h(\boldsymbol{\Gamma}) \tag{3.30}$$

其中，$\boldsymbol{C}'=\boldsymbol{C}-\beta^t\ln\boldsymbol{\Gamma}^{(t)}$，利用贪婪策略更新最佳行或列，定义更新方程为：

$$\rho(x,y)=y-x+\log\frac{x}{y} \tag{3.31}$$

最后，根据公式 3.29 与公式 3.30 对亲和矩阵更新来寻找最佳匹配关系，具体算法流程如表 3.1 所示。

表 3.1 算法流程
Table 3.1 Algorithm process

算法：贪婪最邻近点迭代匹配算法（GIPOT）

输入：初始的耦合矩阵 $\boldsymbol{\Gamma}^{(1)}\leftarrow\begin{bmatrix}11_p & 0\\ 0 & 11_L\end{bmatrix}$，代价矩阵 $\boldsymbol{C}\leftarrow f_{aff}\left(H_i,H_j'\right)$

输出： $\boldsymbol{\Gamma}^{(t+1)}$

1. For $t=1,2,3...$ do
2. $\boldsymbol{Q}\leftarrow\boldsymbol{C}\odot\boldsymbol{\Gamma}^{(t)}$
3. $\boldsymbol{I}\leftarrow\text{argmax}_i\rho\left(\boldsymbol{u_i},\boldsymbol{u_i}\left(\boldsymbol{Q}\right)\right)$
4. $\boldsymbol{J}\leftarrow\text{argmax}_j\rho\left(\boldsymbol{u_j},\boldsymbol{u_j}\left(\boldsymbol{Q}\right)\right)$
5. If $\rho\left(\boldsymbol{u}_I,\boldsymbol{u}_I\left(\boldsymbol{Q}\right)\right)>\rho\left(\boldsymbol{u}_J,\boldsymbol{u}_J\left(\boldsymbol{Q}\right)\right)$ then
6. $\quad\mathbf{a}_I\leftarrow\mathbf{a}_I+\log\frac{u_I}{u_I\left(\boldsymbol{Q}\right)}$ //贪婪地更新某一列
7. Else
8. $\quad\mathbf{a}_J\leftarrow\mathbf{a}_J+\log\frac{\boldsymbol{u}_J}{\boldsymbol{u}_J\left(\boldsymbol{Q}\right)}$ //贪婪地更新某一行
9. $\boldsymbol{\Gamma}^{(1)}\leftarrow diag\left(\exp\left(a\right)\right)\boldsymbol{Q}diag\left(\exp\left(b\right)\right)$ //*diag*()为构建对角矩阵
10. End for
11. Return $\boldsymbol{\Gamma}^{(t+1)}$

3.4　损失函数

为了实现点线匹配，采用点线提取损失、描述子提取损失和点线匹配损失作为模型训练的损失函数，用于衡量不同方面的模型性能。

3.4.1 点线提取损失

在点线检测器分支的训练阶段，输出包括根节点置信图、关键点图、位移图。这三个任务的损失被合并为公式 3.32，定义如下：

$$L_{PLE} = L_{root} + L_p + L_{dis} \tag{3.32}$$

线段根节点在训练中使用标注线段的中间点作为线段根节点的真值，L_{root} 使用二元交叉熵损失进行监督训练：

$$L_{root} = -\sum_i \widetilde{R}_i \log R_i + \left(1-\widetilde{R}_i\right)\log\left(1-R_i\right) \tag{3.33}$$

其中，$\widetilde{R}_i$ 和 R_i 分别代表线段根节点预测坐标值与标注真值。并且，关键点的真值由 ORB 进行预处理获取。L_p 定义如下：

$$L_p\left(X,Y\right) = \frac{1}{H_c w_c}\sum l_p\left(T^{ij},\widetilde{T^{ij}}\right) \tag{3.34}$$

$$l_p\left(T^{ij},\widetilde{T^{ij}}\right) = -\log\left(\frac{\exp\left(T_k^{ij}\right)}{\sum_1^{64}\exp\left(T_k^{ij}\right)}\right) \tag{3.35}$$

线段相对根节点的位移部分用于定位线段的长度和角度，分别使用 L1 损失和 L1 平滑损失，定义为：

$$L_{dis} = \sum_{i=1}^{m}\begin{cases}\left|\theta^i-\hat{\theta}^i\right| + 0.5\times\left(\rho^i-\hat{\rho}^i\right)^2, & if\ \left|\rho^i-\hat{\rho}^i\right|<1\\ \left|\theta^i-\hat{\theta}^i\right| + \left|\rho^i-\hat{\rho}^i\right| - 0.5, & otherwise\end{cases} \tag{3.36}$$

其中，$\left(\theta^i,\rho^i\right)$ 表示真实的线段长度和角度，$\left(\hat{\theta}^i,\hat{\rho}^i\right)$ 表示预测的线段长度和角度。

3.4.2 点线描述子损失

令二维图像为 I，通过仿射变换将其映射得到图像 I'，变换矩阵 θ 代表图像的变形和变换。由于变换矩阵已知，因而可以获得 I 与 I' 上关键点与线段的对应关系。由此可以定义损失函数为：

$$L_d\left(\theta,\left\{d_a^\theta,d_+^\theta,d_-^\theta\right\}\right) = \left[m + \left\|d_a^\theta - d_+^\theta\right\|^2 - \min_{d_-^\theta\in d_-^\theta}\left\|d_a^\theta - d_-^\theta\right\|^2\right]_+ \tag{3.37}$$

其中，参数 m 设置为 0.5，d_a^θ 为图像 I 上关键点线的描述符，d_+^θ 表示正样本图像 I' 上匹配的描述子，d_-^θ 表示负样本图像 I' 上的不匹配的描述子。

3.4.3　匹配损失

对于匹配网络使用交叉熵损失作为匹配预测损失，最后一个最优传输层的输出为 $M(i,j)$，该损失函数可以表示为：

$$L_m = \frac{1}{\left|M_{gt}\right|}\sum_{(i,j)\in M_{gt}}\frac{1}{\sigma^2(i)}\left\|M(i,j)-M_{gt}(i,j)\right\|_2 \tag{3.38}$$

其中，$\sigma^2(i)$ 为特征 i 的置信度方差，$M(i,j)$ 为特征 i 与特征 j 的匹配概率，M_{gt} 为单应变换得到的真实值矩阵。

3.4.4　归一化

总损失函数为上述各项损失函数之和：

$$L_{sum} = \lambda_1 L_{PLE} + \lambda_2 L_d + \lambda_3 L_m \tag{3.39}$$

其中，λ_1、λ_2 与 λ_3 表示各损失函数权重的值，分别为 0.25，0.25，0.5。

3.5　实验与分析

3.5.1　训练细节

在模型训练过程中，原始输入为大小 512×512 的 RGB 图像，输出提取点线的匹配关系。使用 Wireframe[84]和 YorkUrban[85]的训练集的真值来训练算法模型。训练过程中引用了数据增强技术，如随机高斯噪声、运动模糊和亮度变化，以提高网络对光线和视角变化的鲁棒性能力。针对端到端训练的点线匹配网络，我们使用了开源机器学习框架 Pytorch，并在 NVIDIA GeForce GTX 2080Ti GPU 上进行了实验，使用 Adam 优化器和随机初始化的权重来训练网络。以 1e−5 的初始学习率训练网络，并在每个 epoch 上将学习率递减 20%。在不同的数据集上进行了广泛的实验，以证明本章算法的有效性。本章算法使用了几个评价标准在 KITTI 数据集上与典型的 SLAM 方法进行比较。

3.5.2　KITTI 数据集评估

在 KITTI 数据集[86]测试所提出的算法。实验采用的对比性能指标为绝对轨迹误差（Absolute Trajectory Error，ATE）[87]以及依据平移和旋转的相对位姿误差（Relative Pose Error，RPE）[87]，如表 3.2 所示。可以看出，所提算法的性能优于 ORB−SLAM2，特别是在较为困难的序列（具有强照明、运动模糊和低纹理等区域）

中，如 06 和 09。通过多特征融合，在提高算法的准确性的同时，也能够充分利用不同特征之间的互补性和稳定性，从而提高位姿估计的鲁棒性和精度。

图 3.7 展示了 ORB−SLAM2 和本文方法在 KITTI 部分序列上的运行比较结果。可以看出，所提算法整体与 ORB−SLAM2 相当，但在如 00、02 和 09 序列中具有更为复杂的运动环境，相较于 ORB−SLAM2，本章所提算法具有更高的定位精度，这是因为该算法中包含线段特征有效降低相机位姿估计误差，提高系统的鲁棒性，而仅点特征方法的 ORB−SLAM2 特征跟踪容易丢失。

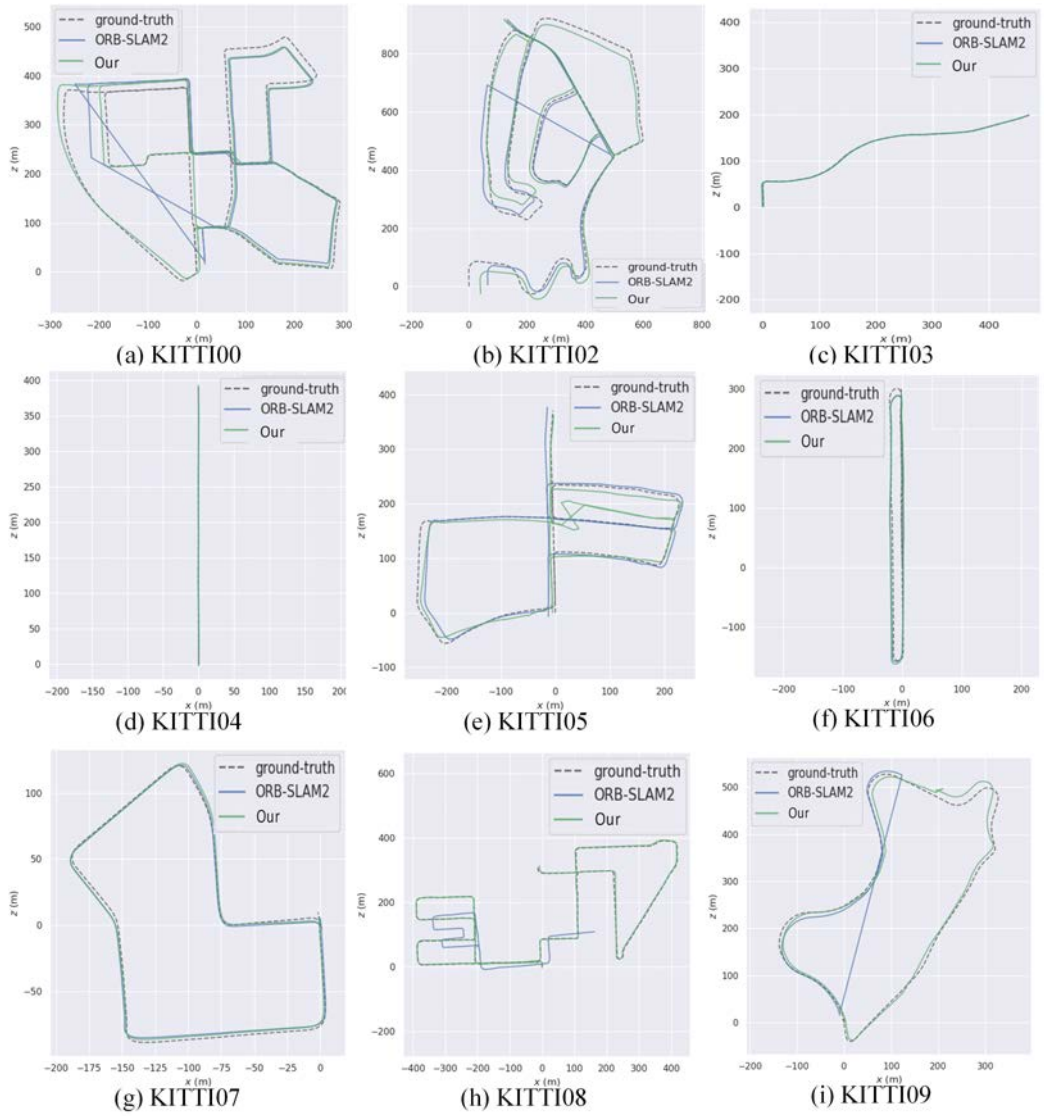


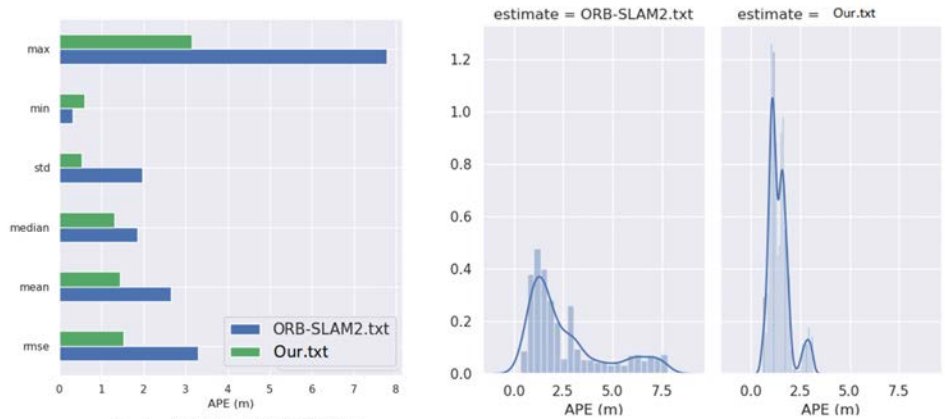
图 3.7 KITTI 数据集上的实验结果

Fig. 3.7 Experimental result on KITTI dataset

表 3.2 不同 SLAM 算法绝对轨迹误差（ATE）与相对位姿误差（RPE）对比

Table 3.2 Comparison between ATE and RPE of different SLAM algorithms

	ORB−SLAM2			本文方法		
序列	ATE（m）	RPE_{trans}（%）	RPE_{rot}（deg/m）	ATE（m）	RPE_{trans}（%）	RPE_{rot}（deg/m）
00	1.266	52.5	0.363	**1.233**	**2.9**	**0.122**
01	4.296	**3.4**	**0.420**	**2.616**	4.8	0.044
02	12.790	4.3	0.107	**12.721**	**3.6**	**0.077**
03	0.403	**0.8**	0.072	**0.385**	2.0	**0.055**
04	0.466	2.2	0.055	**0.192**	**2.1**	**0.040**
05	**0.348**	2.3	0.144	0.402	**1.7**	**0.056**
06	1.184	3.9	0.089	**0.572**	**1.8**	**0.042**
07	0.439	**1.3**	0.076	**0.436**	1.6	**0.046**
08	3.122	12.1	0.076	**2.874**	**3.9**	**0.054**
09	3.319	15.0	0.104	**1.537**	**2.2**	**0.054**
10	**0.927**	2.6	0.090	0.989	**2.1**	**0.060**



（a）误差量化指标图　　（b）绝对位姿误差柱状分布图

图 3.8 本文算法与 ORB-SLAM2 在 KITTI 09 序列上的误差统计特性比较

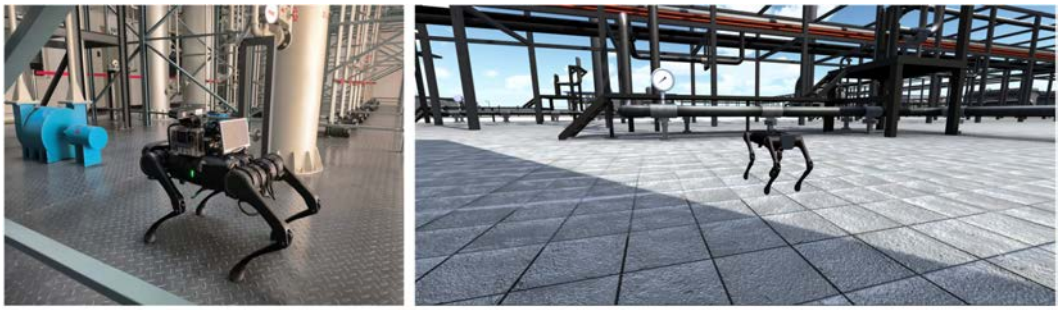
Fig. 3.8 Comparison of error statistical property between Our system and ORB-SLAM2 on KITTI 09 sequence

如图 3.8 显示从 ORB−SLAM2 和本章 SLAM 系统在困难场景下的 KITTI 09 序列中选择的 APE 指标对比图。如图 3.8（a）所示，本章 SLAM 系统与 ORB-SLAM2

相比较，在均方根误差（范围小于 2m）、误差平方和的值及误差极值有较大幅度下降。图 3.8（b）给出的绝对位姿误差柱状统计比较可以看出，本章所提算法 APE 分布范围都优于 ORB-SLAM2。

3.5.3 实际数据评估

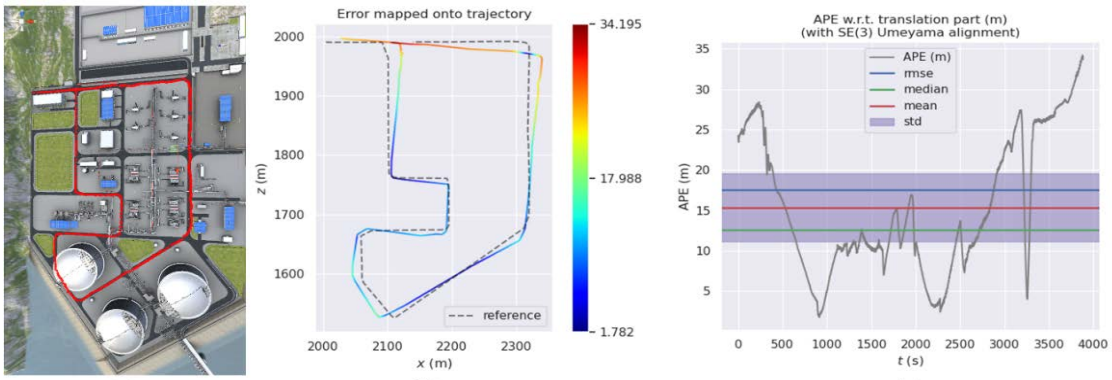
为了验证算法在复杂场景的性能，分别在实际油气场站工业现场和虚拟仿真平台上验证算法可行性。四足机器人及场站环境如图 3.9 所示。采样平台采用 Unity3D 以某油气场站为原型进行开发，具有高逼真的工厂相关设备和环境模拟能力，并通过设置不同的天气情况来进一步验证算法对于环境的鲁棒性。



（a）场站四足机器人　　（b）虚拟场站机器人

图 3.9 实验平台

Fig. 3.9 Experimental platform



(a)　　(b)　　(c)

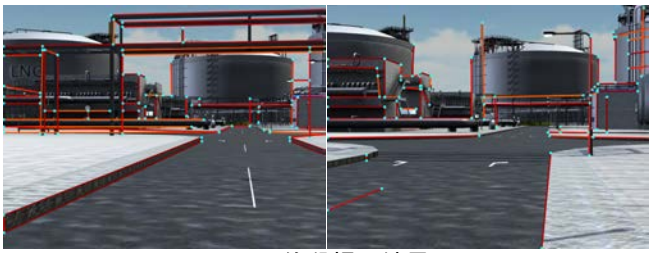
图 3.10 仿真平台轨迹跟踪结果

Fig. 3.10 Simulation platform trajectory tracking results

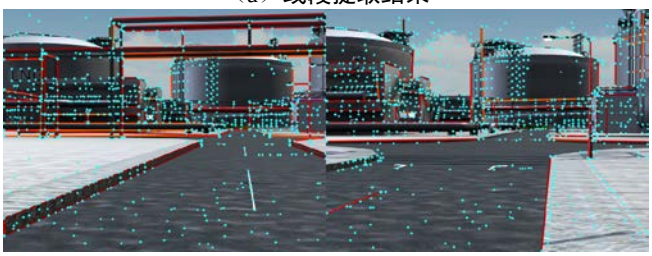
四足机器人按照规划模拟路线以慢走的速度进行巡检，将由视觉传感器获取的图像序列作为算法的输入数据，进而在预测出机器人的位姿的同时进行场站的

地图构建。图 3.10 所示，（a）为设定好的机器人巡检路线，由轨迹误差热力图（b）观察到，当载体突然转向或是快速运动时，SLAM 估算的位姿误差平均误差在 20m 以内，在平缓运动时误差可以缩小到 5m，算法更贴近于真实轨迹。（c）中 ATE 曲线也稳定在较小的范围内，表明了所提算法具有较好的准确性和稳定性。

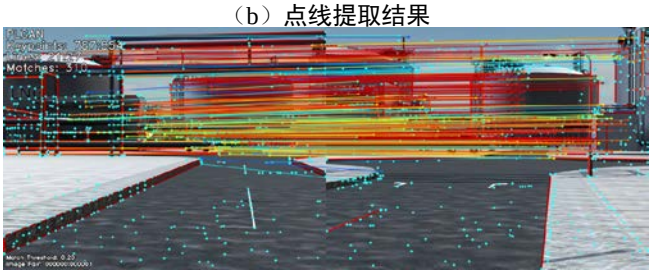
为了展现 SLAM 系统在虚拟场站中对点、线特征的提取和匹配效果，随机截取了相机获取的两帧相邻图像进行提取和匹配。如图 3.11 所示，（c）中颜色代表匹配的置信度，可以明显观察到通过融合点和线特征的匹配，SLAM 系统能够获取更为丰富、更具有效性的特征数据，从而提高其定位和建图的精度和鲁棒性。



（a）线段提取结果



（b）点线提取结果



（c）点线匹配结果

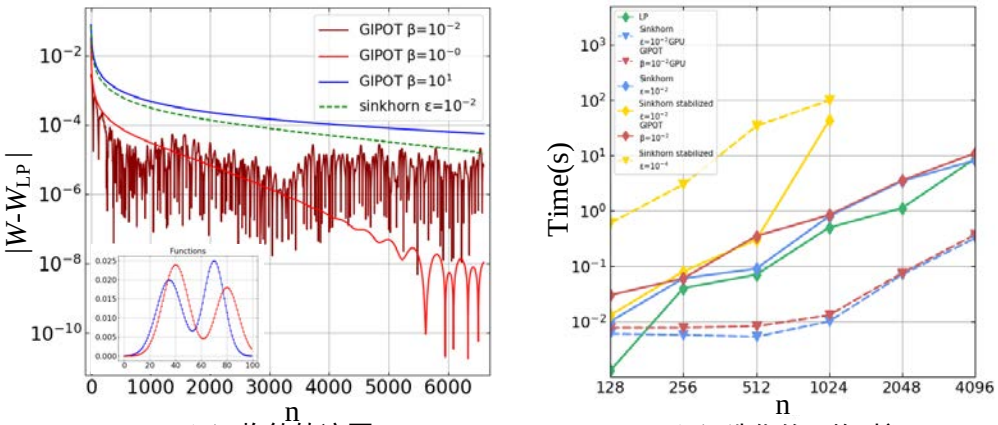
图 3.11 点线特征跟踪的效果

Fig. 3.11 The effect of point-line feature tracking

3.5.4　GIPOT 对比实验

为了说明 GIPOT 的收敛特性，度量两个一维高斯分布的 Wasserstein 距离作为评价指标。如图 3.12 所示，其中蓝色方程为 $0.5N\left(70,8\right)+0.5\left(35,10\right)$；红色方程为

$0.4N(80,9)+0.6N(40,10)$，其中 $N(\mu,\sigma^2)$ 为服从一维高斯分布的概率密度函数，μ 和 σ^2 表示均值和方差。图 3.12（a）显示了不同 β 条件下 GIPOT 的收敛性，所提算法比 Sinkhorn 方法在迭代收敛效率上有明显提升。图 3.12（b）显示了不同算法平均计算时间，结果表明 GIPOT 可以收敛到精确的 Wasserstein 距离，具有良好的性能，其时间复杂度与 Sinkhorn 相当。



（a）收敛轨迹图　　（b）迭代的平均时间

图 3.12 Wasserstein 距离差异图

Fig. 3.12 The difference graph of the Wasserstein distance

3.5.5 消融实验

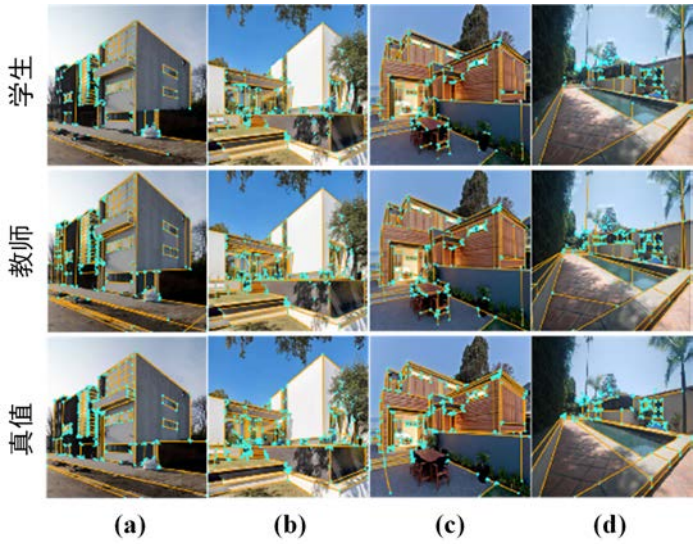


图 3.13 在 Wireframe 数据集上知识蒸馏的定性评价

Fig. 3.13 Quantitative assessment of knowledge distillation method on Wireframe dataset

为了更加充分地验证学生网络对于教师网络的学习效果，分别展示教师网络和学生网络的输出结果。如图 3.13 所示，与真值相比，两种模型都能够高精度地识别关键点和线段，虽然学生网络的结果存在一些小的线段缺失和连接错误，但环境中的线段结构表达基本准确。定量比较如表 3.3 所示，学生网络性能虽然略低于教师网络，但运算速度上提高了 73%左右。

表 3.3 在 Wireframe 数据集和 YorkUrban 数据集上 PL-Net 知识蒸馏方法的定量评价

Table 3.3 Quantitative evaluation of PL-Net point-line detection knowledge distillation method on Wireframe dataset and YorkUrban dataset

方法	Wireframe dataset			YorkUrban dataset			FPS
	F^H	*sAP*	*LAP*	F^H	*sAP*	*LAP*	
Student	77.5	**58.9**	59.8	64.6	25.9	32	**12.5**
Teacher	**80.6**	57.6	**61.3**	**67.2**	**27.6**	**34.3**	7.2

表 3.4 绝对轨迹误差（ATE）消融实验结果（单位：m）

Table 3.4 Results of ablation experiment in term of the RMSE of ATE（Unit: m）

序列	P-SLAM	L-SLAM	PL-SLAM	ORB-SLAM2
00	1.203	6.233	**1.233**	1.266
01	3.934	12.367	**2.616**	4.296
02	**7.689**	—	12.721	12.790
03	0.393	5.457	**0.385**	0.403
04	0.347	13.824	**0.192**	0.466
05	0.863	—	0.402	**0.348**
06	0.884	—	**0.572**	1.184
07	**0.255**	—	0.436	0.439
08	3.122	—	**2.874**	3.122
09	2.625	4.783	**1.537**	3.319
10	**0.447**	5.824	0.989	0.927

通过对不同特征组合下的 ATE 指标进行均方误差（Root Mean Square Error，RMSE）计算，全面验证了多种特征融合对 SLAM 系统的作用，并对其性能表现进

行了充分评估。如表 3.4 所示，与 ORB-SLAM2 和特征类型消融进行对比实验表明，采用点线特征组合方法在大多数序列中显著提升了位姿估计的准确性。

为了验证本算法的鲁棒性，对图匹配网络 HAGNN 进行了消融研究。如表 3.5 所示，“无 EAGAT”用 CHGI 层取代了所有的 EAGAT 层，匹配准确率下降了 9.7%。“无 CHGI”用 EAGAT 层取代了所有的 CHGI 层，匹配准确率下降了 22.6%，“无 HAGNN”用单一的线性投影取代了图神经网络，匹配准确率下降了 26.1%。由此说明所有的模块都是有效的。

表 3.5 HAGNN 消融实验结果
Table 3.5 Ablation of HAGNN

	Known		UnKnown	
	匹配精度（%）	匹配分数（%）	匹配精度（%）	匹配分数（%）
无 EAGAT	79.6	29.5	55.3	15.6
无 CHGI	66.7	25.3	48.2	18.5
无 HAGNN	63.2	19.4	51.2	10.3
Full	**89.3**	**34.2**	**78.3**	**23.8**

表 3.6 位姿估计实验结果
Table 3.6 Experimental results of pose estimation

Feature	Matcher	Pose estimation AUC（%）			P（%）	MS（%）
		@5°	@10°	@20°		
SIFT	NN	7.89	10.22	35.3	43.4	1.7
SIFT	SuperGlue	23.68	36.44	49.44	74.1	7.2
SuperPoint	NN	9.80	18.99	30.88	22.5	4.9
SuperPoint	SuperGlue	34.18	44.32	64.16	84.9	11.1
LSD+LBD	NN	5.43	7.83	28.54	32.5	1.3
SOLD2	NN	18.34	13.22	23.51	63.6	6.2
SuperPoint+SOLD2	Ours	**35.86**	**44.73**	**64.43**	**85.3**	**12.3**
Ours	Ours	**36.67**	**44.26**	**64.73**	**86.6**	**12.7**

将 HAGNN 与最近邻匹配（The Nearest Neighbor，NN）[46]和 SuperGlue[20]特征匹配方法进行比较。如表 3.6 所示，与传统的手工特征和学习的特征匹配相比，本文的特征提取和匹配方法的组合具有显著更高的姿态估计精度，表明其具有更高的特征表达能力。

3.6　本章小结

本章提出一种面向点线感知的异构图注意力视觉 SLAM 算法，通过点线的几何关系同步特征提取网络 PL-Net 实现了两类特征的并行关联提取，并通过感知迁移知识蒸馏策略对网络模型进行优化，提高了特征提取的准确率和实时性。为了提升图像点线匹配精度，提出了一种点线异构图注意力网络，利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间的特征聚合进而提高几何特征的表达能力。最后，将点线匹配问题转换为一个最优传输问题，构建贪婪策略的最邻近点迭代方法对该优化问题进行求解。通过在 KITTI 数据集及多种生产场景进行实验，结果验证了所提算法的有效性。

第 4 章　面向动态环境的实时语义 SLAM

当前大多数视觉 SLAM 的方法都依赖于静态环境假设，这导致在动态环境中的性能表现很不稳定。近年来一些研究人员提出在 SLAM 系统中引入依据深度学习的语义信息来消除动态对象的影响，但这些方法的计算量较高，无法处理未知对象。针对上述问题，本章提出了一个针对动态环境的实时语义 SLAM 系统。首先为了降低计算量，所提算法只对关键帧进行语义分割，以去除动态对象，同时保证静态环境下进行相机跟踪。其次，为了检测未知的动态目标，本章提出了一个轻量级的分割模块，该模块能够获得动态目标的像素级分割结果。最后，通过运动自估计模块计算相机姿态，并采用两种不同的策略来剔除动态特征点。对该方法在公共数据集和现场环境下进行了评价，成功实现了在低功耗嵌入式平台上实时运行的语义 SLAM 系统，同时在动态环境中提供了高定位精度。

4.1　问题与分析

尽管现有的视觉 SLAM 系统取得了显著进展，但这些方法中的大多数严重依赖于静态环境假设，这极大地限制了在现实环境下的部署。因为移动的人、动物、车辆等动态物体对姿态估计和地图重建有负面影响。尽管像 RANSAC 这样的鲁棒估计技术可以用来过滤掉一些异常值，这种改进仍然是有限的，因为它们只能处理弱动态场景，并且当移动的物体覆盖大部分相机视图时仍有可能会失败。动态环境下的图优化模型如图 4.1 所示，运动模型可以表示为连接相机位姿的边，测量模型表示为连接相机位姿和观测地标的边。测量模型可表述为：

$$z_{k,j} = h\left(\xi_k, p_j\right) + v_{k,j} \tag{4.1}$$

其中，ξ_k 表示相机在 k 时刻的姿态的李代数表示，p_j 表示第 j 个路标点的坐标。$z_{k,j} \triangleq \left[u_s, v_s\right]$ 是路标点 v_s 在观测位置 u_s 下的图像中的二维像素坐标，对应于相机在 k 时刻观察到的地标 j。$h(\cdot)$ 为非线性模型，$v_{k,j} \sim \mathcal{N}\left(0, Q_{k,j}\right)$ 是假设均值为零的高斯噪声，$Q_{k,j}$ 表示测量值的协方差。对于此观察结果，误差项可以定义为：

$$e_{k,j} = z_{k,j} - h\left(\xi_k, p_j\right) \tag{4.2}$$

定义代价函数如下：

$$J(x)=\frac{1}{2}\sum_{k}^{m}\sum_{j}^{n}e_{k,j}\left(x\right)^{T}Q_{k,j}^{-1}e_{k,j}\left(x\right) \tag{4.3}$$

其中，x 表示所有相机的位姿和地标位置。这是一个典型的非线性最小二乘问题，如果场景中存在动态对象，它们将会破坏测量模型。在公式 4.1 中，对于动态对象 $v_{k,j}$ 为非零值，但非零值的异常噪声会给优化过程带来干扰信息。如图 4.1（b）所示，动态地标破坏了相机位姿的约束（红色实线边）以及相机姿态和地标之间的约束（红色虚线边）。

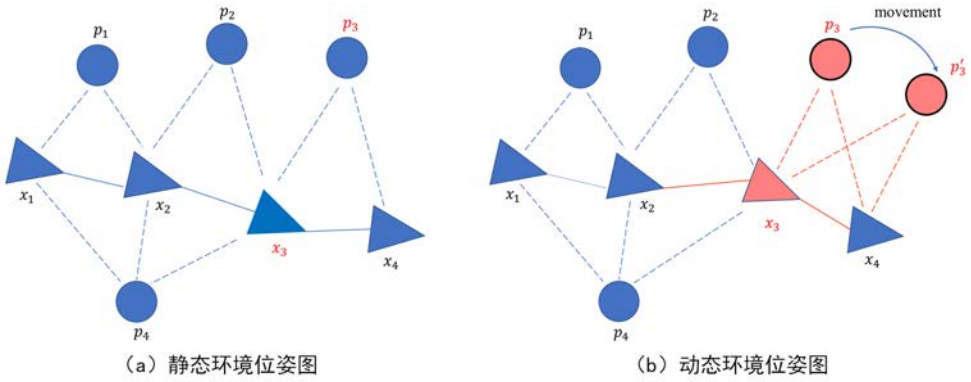


图 4.1 图优化模型图

Fig. 4.1 The graph optimization model

设 $d_{k,j}$ 表示地标 j 在 k 时刻的二维图像上的位置变化，将公式 4.2 改写为：

$$e'_{k,j}=z_{k,j}+\beta_{k,j}d_{k,j}-h\left(\xi_k,p_j\right) \tag{4.4}$$

其中，若地标 j 在 k 时刻为动态的，$\beta_{k,j}$ 设为 0，否则设为 1：

$$\beta_{k,j}=\begin{cases}1, & \text{dynamic}\\ 0, & \text{otherwise}\end{cases} \tag{4.5}$$

为了消除动态地标对位姿估计的负面影响，代价函数方程公式 4.3 也将进行相应的微调来准确地估计相机的位姿和地标位置。区分动态物体的方法将在第 4.2 节中进行详细描述。

为了利用语义信息识别动态对象，现有方法对每个输入的图像帧进行语义分割。这将导致相机跟踪线程耗时增加，因为跟踪过程必须等到分割完成。本章只从关键帧中提取语义信息，以去除潜在的动态对象，并在建图的过程中只进行对静态特征点的相机跟踪。由于关键帧和地图更新过程在独立的线程中，并且对速度的要求不高，因此使用语义分割的总跟踪时间可以显著减少。此外，为了处理未知的运动对象，本章提出了一个不需要关于运动对象的先验信息的高效几何模块。利用 K-Means 算法将深度图像分割成几个区域，并根据深度图像的平均重投影误差来识别动态区域。与在密集优化框架中使用几何聚类来检测动态对象[88]不同，所提算法直接从稀疏特征的重投影误差，进而加快处理速度，使其对动态内容具有更强的鲁棒性。

4.2 算法实现

本节描述了所提出的动态视觉 SLAM 方法，由四个模块组成：语义分割、运动自估计、动态点检测和特征 SLAM 框架。在后续的章节中，将详细介绍上述内容。

4.2.1 方法概述

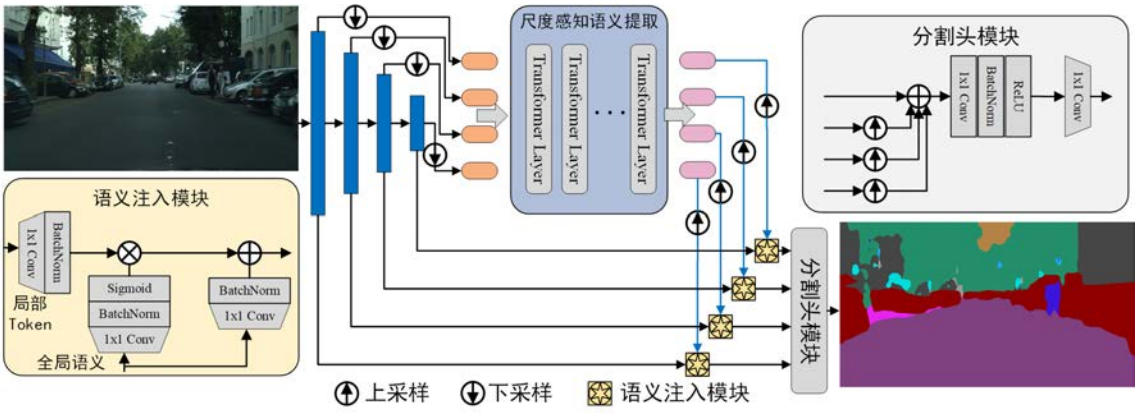


图 4.2 TopFormer 结构图

Fig. 4.2 The architecture of the token pyramid transformer

图 4.3 展示了单目和 RGB-D/双目情况下的算法框架，输入帧首先由轻量级网络 TopFormer 进行分割，网络结构图如图 4.2 所示，网络由几个部分组成：token 金字塔模块、语义提取、语义注入模块和分割头模块。其中，token 金字塔模块将

一个图像作为输入，并生成各像素级的 token 向量来获得丰富的语义和较大的接受域。Tranformer[89]被用作语义提取器，它将 token 金字塔生成的各层特征作为输入，并生成多尺度感知语义。进而语义被注入到相应尺度的 token 中，以增强语义注入模块的表达能力。最后，分割头模块使用增强的 token 金字塔特征来执行分割任务。

网络输出语义分割结果后，去除在动态分割区域内的特征点以进行初始的运动自估计。这里采用了文献[27]中提出的低成本跟踪算法来计算初始位姿，由于姿态估计中涉及的特征点较少，且缺乏局部束调整，因此计算出的初始姿态并不准确。为此，提出了两种不同的策略来检测运动自估计后的动态特征点。在单目情况下（虚线）使用外极约束来提取动态点，异常值是通过匹配的点到其对应的外极线的距离来区分。在 RGB−D/双目情况下（实线）特征点具有来自深度图像或三角测量的深度信息，将前一帧的特征点重新投影到当前帧中，并使用重投影偏移向量[*u*,*v*]来描述静态区域的偏移分布，然后将不遵循先验分布的特征点视为动态点。最后，利用动态特征点来确定动态掩码（Mask）区域，并剔除区域内的所有特征点，得到的静态点被输入到视觉 SLAM 算法中进行运动跟踪和定位。

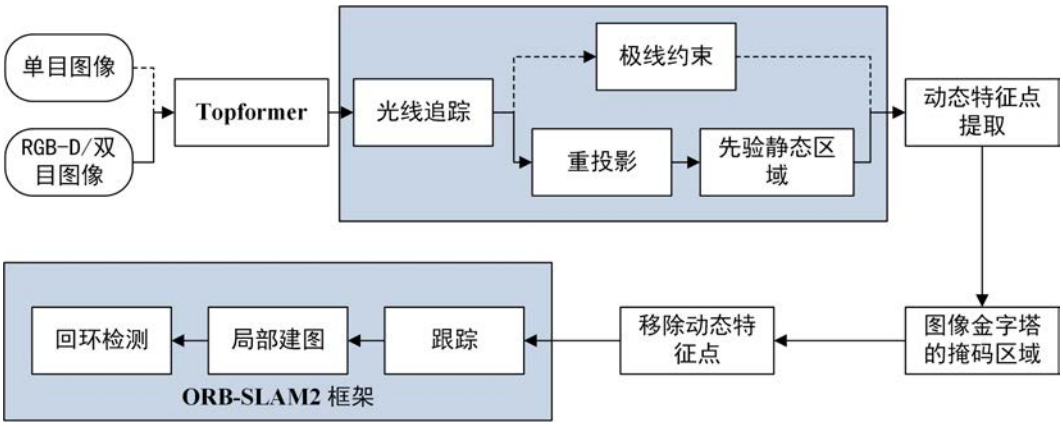


图 4.3 系统框架图

Fig. 4.3 System framework diagram

4.2.2　潜在移动目标分割

TopFormer 是一种高效的轻量级实例分割网络，在 COCO 数据集[90]上训练出 80 个类进行实例分割。在 COCO 上使用预先训练过的模型，从这 80 个类别中选择 20 个常见的动态物体，如人、自行车、汽车、摩托车、鸟等。假设在实际实验场景中，这 20 个类别代表了所有潜在动态对象。TopFormer 被设计为一个相对独

立的模块进行分割以获得潜在的动态区域掩码，在这些潜在移动物体区域的特征点被认为是不可靠的，利用初步得到的相机位姿和运动一致性约束对特征点的运动轨迹进行分析，从而确定哪些特征点是静态的，哪些是动态的。如图 4.4 所示，图（a）为由 TopFormer 生成 RGB 图像的初始语义分割结果，图（b）为使用上述 20 个类提取的潜在移动对象。

（a）像素级语义分割　　（b）潜在移动对象区域

图 4.4 语义分割示意图

Fig. 4.4 Diagram of semantic segmentation

4.2.3 运动自估计

在对图像中潜在的动态对象进行分割后，初步假设分割后的区域是动态的，并丢弃这些区域中存在的所有特征点和接近区域边缘的特征点。本章依然采用了在文献[27]中提出的轻量级跟踪算法，与 ORB−SLAM2 中的方法不同，本章不采用局部 BA 和新的关键帧决策，而是直接将特征点与前一帧匹配，并将关联的映射点投影到当前帧。最后，通过最小化公式 4.2 中的重投影误差来估计相机的姿态，该模块可以降低计算初始位姿的成本。

4.2.4 动态点检测

在运动自估计模块中，剔除所有落在潜在动态区域中的点，并计算了初始位姿（有误差），然后应用位姿提取动态点来确定掩码区域特征是否为动态。如上所述，提出了两种不同的方法来检测动态特征点。首先，在 RGB−D/双目视觉情况下，通过运动自估计，得到了相机的初始姿态（透视变换矩阵）。这里使用 Lucas−Kanade 算法[91]计算光流，以获得两帧之间匹配的特征点。

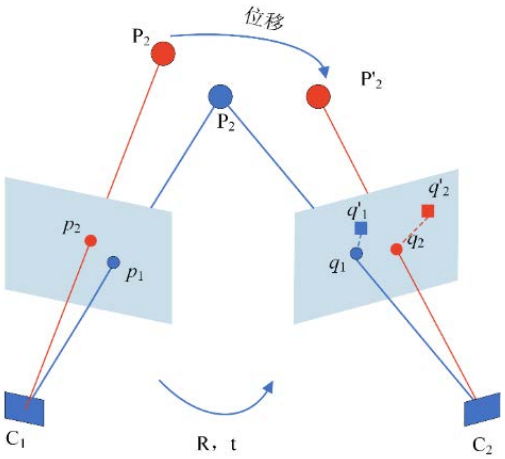


图 4.5 极线约束示意图

Fig. 4.5 Diagram of epipolar constraint

然后，将前一帧的特征点重新投影到当前帧中，如图 4.5 所示。$\{p_1, p_2\}$表示时刻$t-1$时前一帧中的特征点，$\{q_1, q_2\}$表示t时刻当前帧中的特征点，$\{\hat{q}_1, \hat{q}_2\}$表示当前帧中与特征点$\{p_1, p_2\}$相匹配的重投影点。q_1'和q_2'通常不重合，因为估计的变换矩阵不准确。因此，每个点都会有一个偏移向量，如$q_1 - q_1'$，当地标P_2在t时刻移动到P_2'位置时，$q_1 - q_1'$和$q_2 - q_2'$之间的距离和角度的差异就会急剧增加。图 4.6（a-c）显示了三种情况下的静态区域和移动的人之间的差异，可以看出，即使由运动自估计模块计算出的初始相机姿态并不完全准确，静态点和动态点分布的差异仍可以帮助区分这两种点。

为了得到一个用于区分动态点的自适应阈值，提出了一种加权平均方法来描述静态区域（除图像中潜在的动态掩码外的所有其它区域）中的偏移向量。静态区域中的偏移向量集表示为$V_{state} = \{q_i, i = 1, 2, 3, \ldots, n\}$。首先，计算该集合中每个向量的角度$\theta_i$和模值$\epsilon_i$。然后，指标$T_i$和$\phi_s$的定义如下：

$$T_i = \frac{\epsilon_i}{\sum_{k=1}^{n} \epsilon_k} + \frac{|\theta_i|}{\sum_{k=1}^{n} |\theta_k|}, i = 1, 2, 3, \ldots, n \tag{4.6}$$

T_i的平均值记为ϕ_s：

$$\phi_s = \frac{\sum_{i=1}^{n} T_i}{n}, i = 1,2,3,\ldots,n \tag{4.7}$$

其中，$V_{\text{other}} = \{q_j, j = 1,2,3,\ldots,m\}$ 表示潜在移动区域的偏移向量集，$P_{\text{other}} = \{p_j, j = 1,2,3,\ldots,m\}$ 表示对应的特征点集。对于 V_{other} 中的每个偏移向量，通过计算 T_j 来表示，并与 ϕ_s 进行比较。以下不等式用于判断特征点是否为动态：

$$\begin{cases} T_j > \phi_s, & \text{if } p_j \text{ is dynamic} \\ T_j < \phi_s, & \text{if } p_j \text{ is static} \end{cases} \tag{4.8}$$

动态点提取的结果如图 4.6（d-f）所示，移动的人身上的点被划分为动态点（红点）。由于运动的不明显，坐着的人身上的大部分点被划分为静态点（绿色点）。

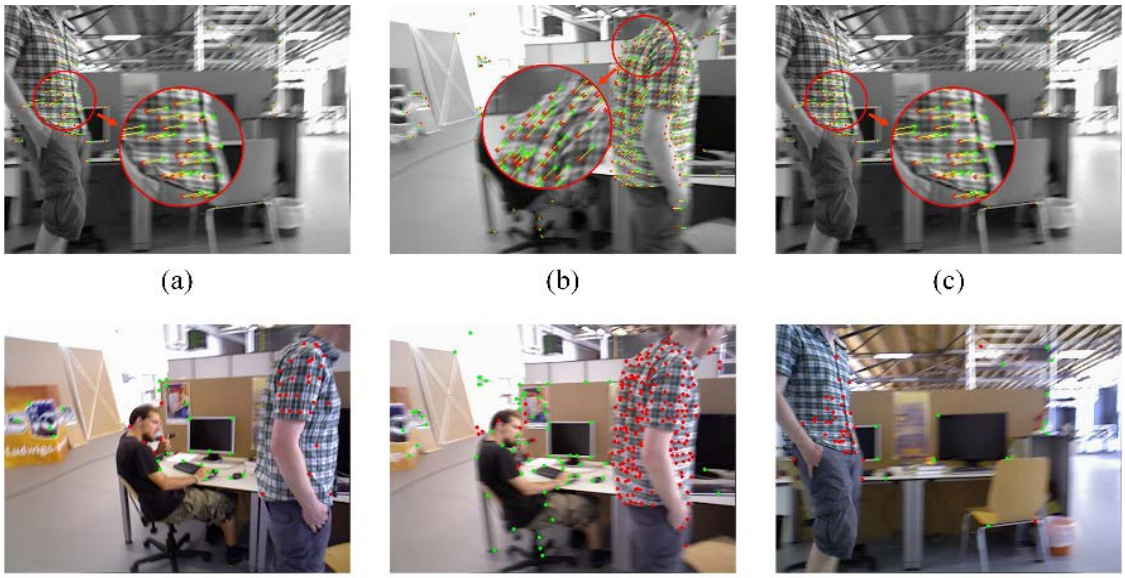


图 4.6 特征点运动状态分类结果

Fig. 4.6 Classification result of motion state of keypoints

在单目情况下，使用外极约束来提取动态点。外极约束可以表示为：

$$q_i^T F p_i = 0, i = 1,2,3,\ldots,m \tag{4.9}$$

其中，F 为表示基本矩阵，q_i 和 p_i 表示当前帧和前一帧中匹配的特征点。对于当前帧的每一个点，计算它到相应的外极线的距离 $Fp_i=[x,y,z]^T$。距离定义如下：

$$D=\frac{\left|q_i^T F p_i\right|}{\sqrt{\| x\|^2+\| y\|^2}}, i=1,2,3,\ldots,m \tag{4.10}$$

其中，(x,y) 表示的当前帧特征点的像素坐标，D 表示距离。如果该距离超过了预设的阈值，则将此点视为一个动态点。

最后，根据移动区域内动态点数量来判断该对象的运动状态，进而剔除从图像金字塔中提取的区域内所有特征点，整体算法流程如表 4.1 所示。

表 4.1 算法流程
Table 4.1 Algorithm process

算法：动态点检测算法

输入：提取的关键点集合 P^T、特征偏移向量集 V^T、分割对象数目 T

输出：静态点集合 P_{static}

1. 初始化运动距离阈值 ϕ_s，特征点数量阈值 $\boldsymbol{K}$
2. For $t=1,2,3...T$ do
3. count $\leftarrow 0$ //初始化计数
4. For each P_i^t in P^t do
5. If $V_i^t>\phi_s$ then
6. count $\leftarrow$ count+1
7. End for
8. End if
9. If count<$\boldsymbol{K}$ then
10. 将 P_i^t 从 P^T 中移除
11. End if
12. End for
13. Return $P_{static}\leftarrow P^T$

4.3 实验与分析

在本节中，为了验证算法的性能，使用公共数据集 TUM 和 KITTI 上进行对比实验。将其与 DS-SLAM[25]、DynaSLAM[27]和 ORB-SLAM2[8]进行比较来演示在动态场景中的提升效果。所有的实验都是在一台带有 Intel i7-10700 CPU、16 GB 内存和 GeForce GTX 2080 Ti GPU 的电脑上进行的。

4.3.1 TUM 数据集

TUM RGB-D 数据集[92]包含彩色和深度图像以及精确的真值轨迹。本章将室内环境下的 39 个序列划分为动态和静态两种类型，按照环境中动态区域的运动幅度又可以将动态场景分为高、低两种类型。在低动态场景中，两个人在一张桌子前聊天和做手势，运动幅度较小。在高动态场景中，两个人在桌子前来回走动，运动幅度较大。按照相机运动方式数据集分为四种类型，halfsphere 表示相机运动轨迹在直径 1 米的球面上，xyz 表示相机沿着 x-y-z 进行直线平移运动，rpy 相机的姿态发生变化，可以是俯仰、横滚、偏航等运动方式，static 表示相机几乎静止。

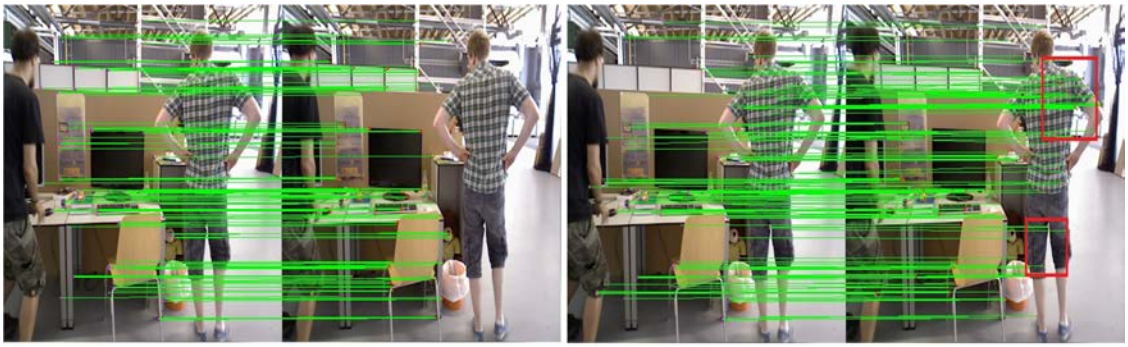


图 4.7 帧间特征匹配结果

Fig. 4.7 Feature matching results between frames

本章利用度量绝对轨迹误差（ATE）[87]进行定量评价，图 4.7 显示了高动态场景中 fr3/w/xyz 序列的特征提取和匹配结果。实验结果表明，本章方法可以消除在红色区域的行人的影响。同时，利用 ORB-SLAM2、本章方法和 DynaSLAM 系统进行轨迹精度进行比较分析，给出了 ATE 的均方根误差（RMSE）和标准差（Standard Deviation，STD）的值。提升值定义如下：

$$\zeta = \left(1 - \frac{\gamma}{\mu}\right) \times 100\% \tag{4.11}$$

其中，ζ 为提升值，γ 为该算法的 RMSE 值，μ 为 ORB−SLAM2 的 RMSE 值。由此，提升值 ζ 代表算法与 ORB−SLAM2 系统相比的提升。

从表 4.2 中可以看出，高动态场景的 RMSE 提升值的平均值为 96.27%，消除了动态对象在姿态估计中的影响。与 ORB−SLAM2 相比，本章算法的性能有一个数量级的提升，然而，对于低动态场景，平均 RMSE 提高率只有 20.19%，这是由于场景中物体的运动振幅较小并且 ORB−SALM2 本身具有一定的动态特征剔除能力，故其与 ORB−SALM2 相比的提升较少。例如，在 fr3/s/rpy 序列中，拍摄场景为坐在椅子上的两个人进行简单手臂动作，由于运动幅度较小并且运动区域较小，这些动作并不会显著影响相机的姿态估计。

表 4.2 RGB-D 实验的绝对轨迹误差（ATE）[m]的定量比较结果

Table 4.2 Quantitative comparison results of the absolute trajectory error (ATE) [m] in meters for the stereo experiments

序列	ORB-SLAM2		DS-SLAM	DynaSLAM	本文方法		提升幅度	
	RMSE/m	STD/m	RMSE/m	RMSE/m	RMSE/m	STD/m	RMSE/%	STD/%
fr3/w/xyz	0.7137	0.3584	0.0241	0.0158	0.0263	0.0072	97.93	97.99
fr3/w/rpy	0.8357	0.4169	0.3741	0.0402	0.0328	0.0194	96.08	95.35
fr3/w/static	0.3665	0.1448	0.0081	0.0080	0.0079	0.0040	97.84	97.24
fr3/w/half	0.4068	0.1698	0.0282	0.0276	0.0274	0.0137	93.26	91.93
fr3/s/static	0.0092	0.0039	0.0061	0.0064	0.0063	0.0032	31.52	17.95
fr3/s/rpy	0.0245	0.0172	0.0187	0.0302	0.023	0.0134	6.12	22.09
fr3/s/half	0.0231	0.0112	0.0148	0.0191	0.0178	0.0103	22.94	30.41

图 4.8 和图 4.9 显示了 ORB−SLAM2 和本章算法在 6 个序列的运动真实轨迹。图 4.8 显示了为高动态场景选择的 ATE 图，由于通过消除动态物体的负面作用，从而减少相机位姿估计误差，相比 ORB-SLAM2 有更精确的位姿估计。图 4.9 显示了针对低动态场景的选择的 ATE 图，原始的 ORB−SLAM2 算法也可以为这些场景提供良好的性能，与高动态场景相比，本章方法在低动态场景下的改善并不明显。

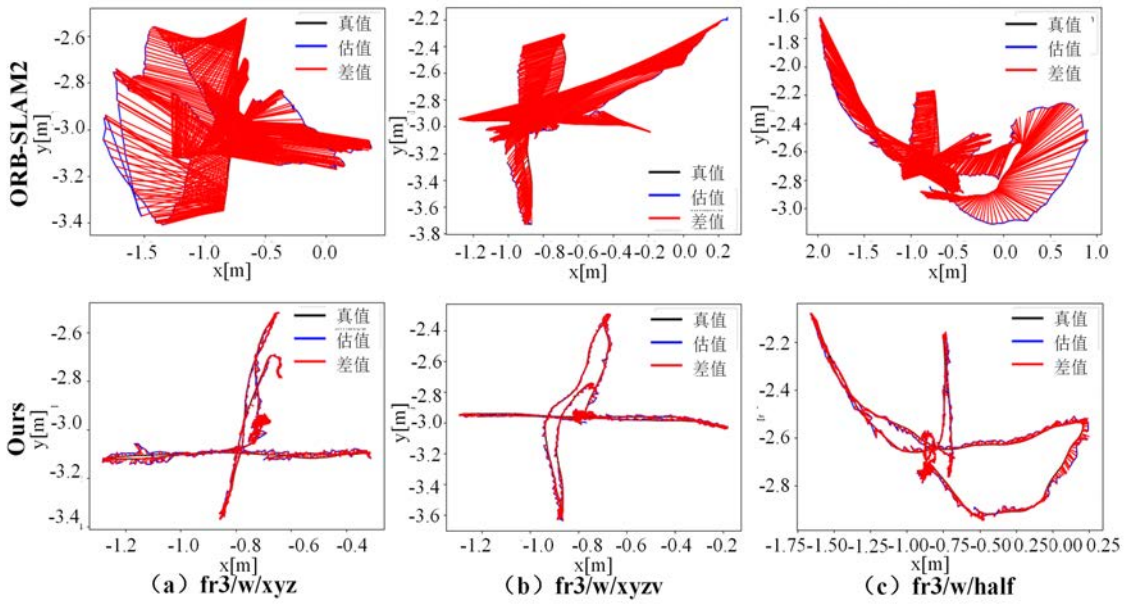


图 4.8 高动态场景绝对轨迹误差对比图

Fig. 4.8 ATE comparison for high-dynamic scenes

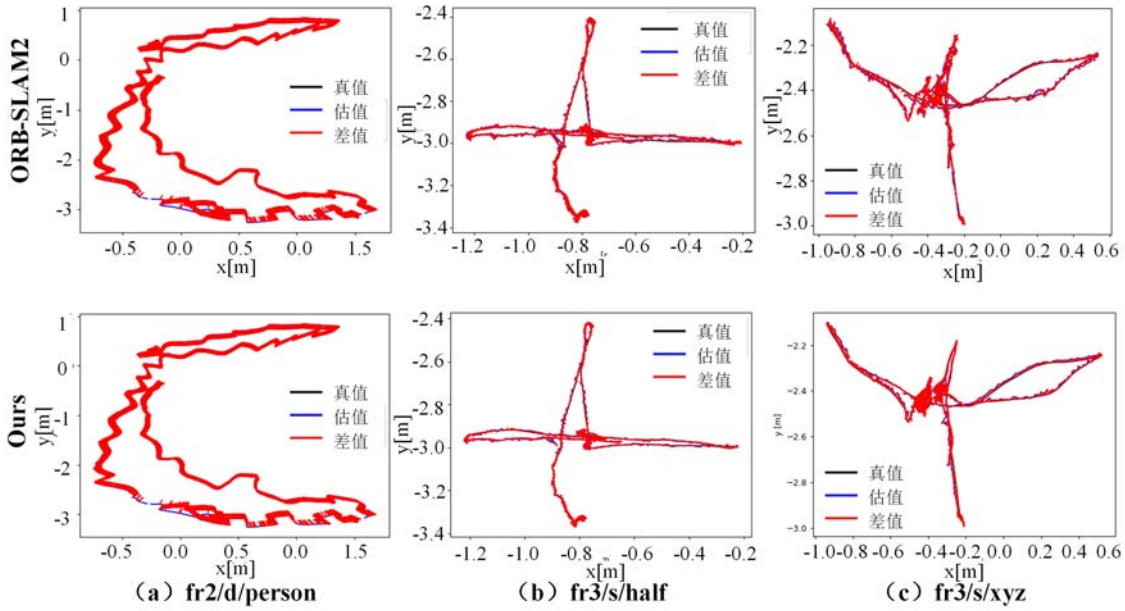


图 4.9 低动态场景绝对轨迹误差对比图

Fig. 4.9 ATE comparison for low-dynamic scenes

对 TUM 数据集中的单目图像进行了实验，实验结果如表 4.3 所示，与表 4.2 相比，ORB−SLAM2[8]可以在比 RGB−D 的情况下获得更准确的结果，原因是单目

初始化算法比 RGB-D 情况有更严格的限制，ORB-SLAM2 只有在满足匹配和视差条件后才能成功初始化。因此，在动态场景中初始化可能需要很长时间，然而，由于消除了动态对象的影响，本章方法的初始化比 ORB-SLAM2 的初始化更快。表 4.3 中跟踪轨迹的百分比是成功跟踪的帧与图像总数的比值，该值表明本章方法生成轨迹比 ORB-SLAM2 更完整。

表 4.3 单目 ATE [m]的 RMSE 和跟踪轨迹的百分比[%]的定量比较结果

Table 4.3 Quantitative comparison results of the RMSE of the ATE [m] and percentage of tracked trajectory [%] for monocular cameras

序列	ATE RMSE（m）			跟踪轨迹（%）		
	ORB-SLAM2	DynaSLAM	本文方法	ORB-SLAM2	DynaSLAM	本文方法
fr3/w/xyz	0.014	0.018	0.020	85.61	87.37	87.97
fr3/w/rpy	0.017	0.021	0.024	90.12	97.84	97.26
fr3/w/static	0.066	0.052	0.050	85.82	85.11	87.01
fr3/w/half	0.005	0.004	0.004	89.30	90.01	91.33
fr3/s/rpy	0.008	0.013	0.014	95.47	95.51	95.64
fr3/s/xyz	0.042	0.021	0.021	80.36	54.39	62.68

在某些序列中，ORB-SLAM2 的准确性略高于本章所提算法，其原因是 ORB-SLAM2 估计轨迹距离太短导致累积误差较小。与 ORB-SLAM2 相比，本章方法初始化创建了一个没有动态对象的地图，并且具有更多的轨迹信息，这更有助于后续的信息重用（如密集地图的重建）。

4.3.2　KITTI 数据集

KITTI 数据集[86]是一个被广泛应用于评估自动驾驶场景中计算机视觉算法性能的数据集。它涵盖了来自不同环境（如城市、农村和高速公路）的双目图像数据。为了更直观地展示动态点的去除效果，在 KITTI 数据集上进行了姿态跟踪实验。图 4.10 显示了和 ORB-SLAM2 系统在姿态估计过程中跟踪的特征点，本章方法在相机姿态估计过程中剔除移动的汽车或自行车上的特征点，因此可以消除动态物体的影响。

然后，对 11 个序列进行了轨迹精度分析，表 4.4 显示了与 ORB−SLAM2 和 DynaSLAM 的实验结果的比较，使用绝对轨迹误差（ATE）的 RMSE 作为评价指标。由表可知，在包含一定数量的运动物体的序列中，如汽车和自行车，本章方法可以通过消除道路上动态物体的影响来实现更高的跟踪精度。然而，在少量序列中，大部分分割的车辆都停在道路旁边，导致本章方法的定位精度略低于 ORB−SLAM2。原因是由 TopFormer 产生的潜在的动态区域（例如，停放的汽车）是静态的，在去除这些区域中的所有静态点后，用远处点计算出的初始运动自估计存在误差。在动态点提取模块中，一些静态点被误认为是动态的，不会用于跟踪和建图，导致跟踪精度略有下降。

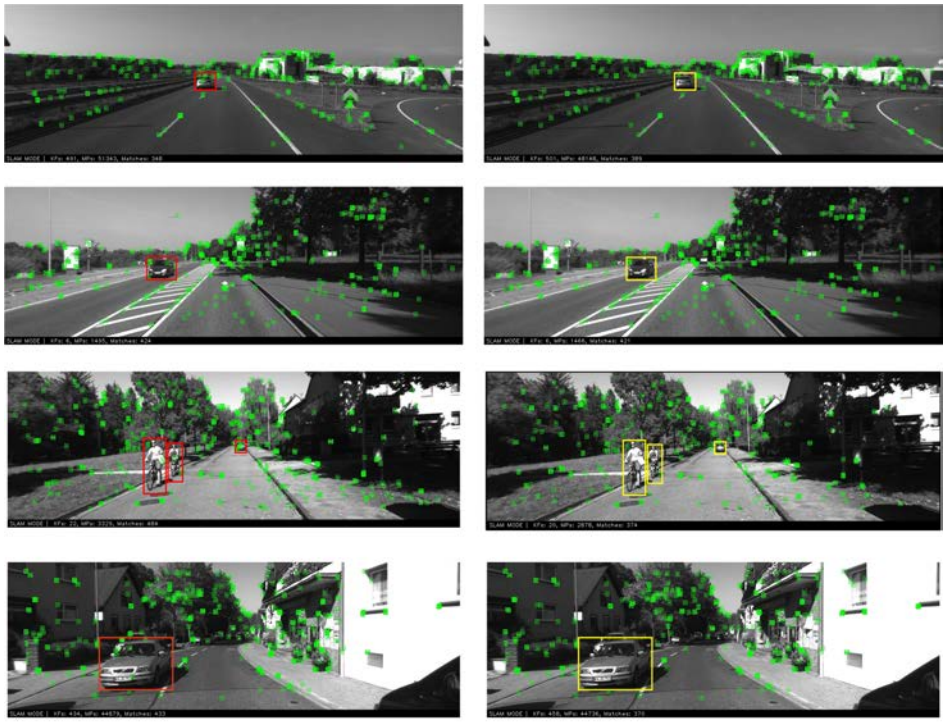


图 4.10 KITTI 跟踪实验对比

Fig. 4.10 Comparison of tracking experiment on KITTI

图 4.11 和图 4.12 给出所提算法（图中用 Ours 表示）与 ORB−SLAM2 在四个序列的绝对姿态误差（APE）和运动轨迹对比图。由于 DynaSLAM 的绝对轨迹误差与 ORB−SLAM2 相似因此不再绘制。图 4.11 显示了本章方法与真实轨迹误差比较。可以看出，误差较大的区域基本分布在轨迹的拐角处。从图 4.12 可以看出，

本章方法的姿态误差明显低于 ORB-SLAM2，特别是在序列 9 中。总体而言，本章算法的运动轨迹与真值一致，具有较高的精度。

表 4.4 双目 ATE 的 RMSE 的定量比较结果 （单位：m）

Table 4.4 Quantitative comparison results of the RMSE in ATE for stereo cameras (Unit: m)

序列	ORB-SLAM2	DynaSLAM	本文方法
KITTI 00	**1.3**	1.4	1.4
KITTI 01	11.4	9.4	**9.1**
KITTI 02	6.2	6.7	**4.6**
KITTI 03	0.6	0.6	**0.6**
KITTI 04	0.2	0.2	**0.2**
KITTI 05	0.8	0.8	**0.7**
KITTI 06	0.8	0.8	0.8
KITTI 07	**0.5**	0.5	0.6
KITTI 08	3.8	3.5	**3.3**
KITTI 09	3.4	**1.6**	1.7
KITTI 10	1	1.2	**1.1**

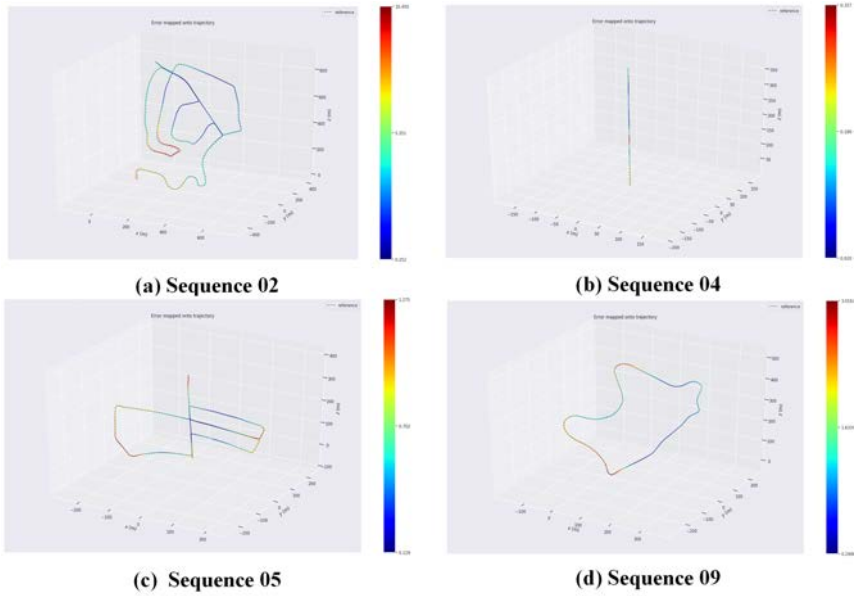


图 4.11 3 维轨迹误差热力图

Fig. 4.11 3D trajectory error heatmap

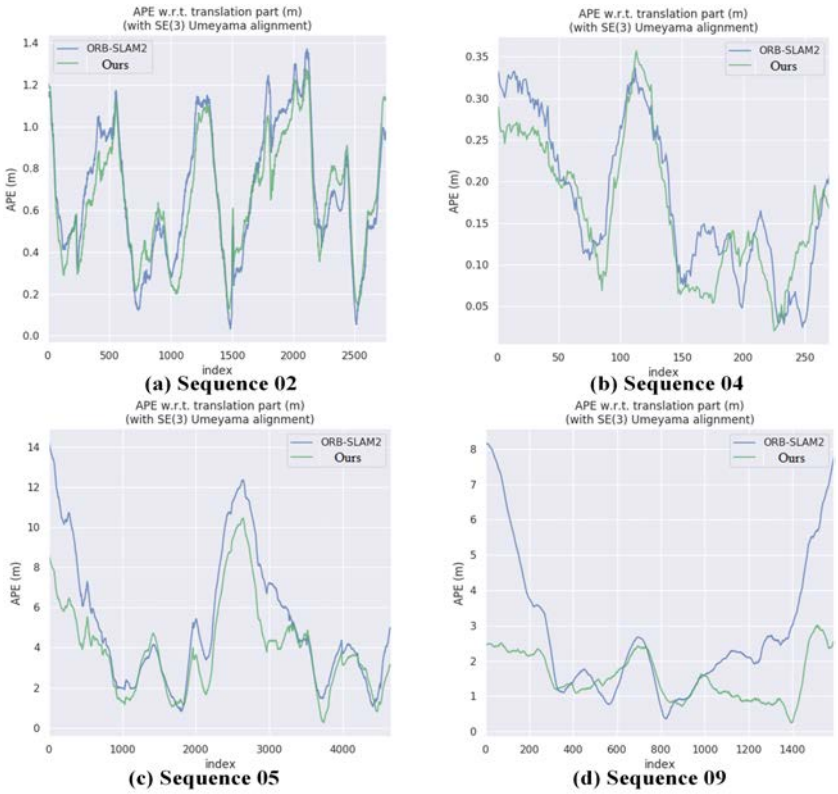


图 4.12 KITTI 跟踪实验对比图

Fig. 4.12 Comparison of tracking experiment on KITTI dataset

4.3.3 实时性评估

为了保证实验的完整性，表 4.5 展示了各模块在 TUM 和 KITTI 数据集上耗时分析。所有的实验都是在一台带有 Intel i7 CPU、16 GB RAM 和 GeForce GTX 2080 Ti GPU 的电脑上运行的，操作系统为 Ubuntu 18.04。

表 4.5 不同组件平均计算时间（单位：ms）

Table 4.5 Average computation time for different components (Unit: ms)

数据集	语义分割	运动自估计	动态点检测
TUM	201.02	3.16	40.64
KITTI	210.11	7.03	94.59

本节计算了在不同方法的 TUM 和 KITTI 数据集上的平均时间。由于添加了 TopFormer，语义分割模块是最耗时的。未来有两种解决方案来解决这个问题，一是用更轻量级的语义分割网络替代 TopFormer，适合小设备，但精度可能较低。二

是在高配置服务器上部署语义分割模块，将结果传输到本地设备，可提高精度，但需要考虑传输成本和延迟。具体方案应综合考虑计算资源、精度、延迟和数据传输成本等因素。

4.4　本章小结

本章提出了一种结合语义分割的视觉 SLAM 方法，该方法可以在 RGB-D、双目和单目相机的高动态环境下提供良好的性能。该方法可分为语义分割、运动自估计、动态点检测和依据特征的 SLAM 框架四个模块。本章为 ORB-SLAM2 添加了一个前端模块，以过滤出与动态对象相关联的数据。在 RGB-D/双目的情况下，利用特征点的重投影信息来构建一个自适应阈值，用于区分动态点。在单目的情况下，使用外极约束来提取动态点。在 TUM 数据集中，当面对高动态场景时，本章的方法在位姿估计方面相较于 ORB-SLAM2 表现出了数量级的提升。这意味着本章的方法在处理高速移动场景中的 SLAM 问题方面具有更好的性能，并且可以更好地捕捉快速移动的目标。在单目场景下与 ORB-SLAM2 项目比，定位精度平均提升了 12%，并且本章方法获得了更多的轨迹和地图信息。在 KITTI 数据集上，本章方法对大多数序列都实现了更高的跟踪精度，除了那些包含许多静态的潜在移动的对象（例如，停放的汽车）。为了克服这个问题，在后续研究中，考虑利用一个更准确的数学模型来描述分布重投影偏移向量在一个先验静态区域，而不是一个自适应阈值。此外，本章算法使用在强静态场景中跟踪的特征来估计相机的姿态，如果所有的场景都是动态的，那么由于缺乏静态特征，目前的方法无法获得准确的结果。在后续工作中，考虑利用动态对象的信息来协助位姿估计和实时重建。

第 5 章　油气场站仿真平台的设计与实现

为了验证本文提出的视觉 SLAM 系统性能，进行了仿真实验和分析。首先通过 Unity3D 虚拟引擎搭建了点特征稀少的油气场站仿真环境，并在该环境中自制相关数据集进而分析了本文所提算法的实验效果。该仿真平台涵盖了油气领域的理论知识和典型事故模拟，并将灾害机理和效果相结合。利用 Unity3D 自带的粒子系统模块进行逼真的事故模拟，并通过可视化的方式模拟了事故的场景。为了解和掌握油气事故的发生机理和影响提供了有力工具，并为事故预防和应急响应提供了支持。除此以外，系统以中石化北京某油气场站为原型进行建模开发，并在无人值守场站的实际工业应用中得到了验证，为实际工作中进行事故预防和相关数据采集提供了有价值的参考方案。

5.1　需求分析

数字化和工业化的深度融合使得油气场站的无人值守成为必要趋势。由于场站内部设施种类和数量众多，例如场站仪表等相关设备需要进行定期记录并检查。然而，这种工作量大且人工记录存在一定误差。此外，现有场站监控设备存在视觉盲区，设备状态信息利用率低。当前融合深度学习技术的场站设备检测应用较为少见，且大部分设备状态信息以文本或表格形式存储，无法便捷获取实时信息。通过实际场站搭建虚拟仿真平台，可以直观了解场站布局和组织结构，预测事故发生，并帮助管理人员提高对应急事件的反应能力，从而大幅提高场站日常管理效率。

根据本章内容，该系统主要包括场景漫游、自动巡检、仪表识别、事故模拟四个主要功能模块，功能需求的具体描述如下：

（1）场景漫游：方便用户进行场景中的漫游来熟悉对油气场站的结构布局。

（2）天气设置：模拟不同的天气情况，以验证各种算法在油气场站的适用性和准确率。

（3）事故模拟与仪表设置：利用 Unity3D 强大的粒子光效系统来模拟火灾，泄漏等事故，并可对仪表参数进行设置。

（4）自动巡检：管理者可对巡检目标进行设定，自动生成最优巡检路径，巡检机器人可通过携带的图像传感器对场站中阀门、仪表、控制台等各种设备定位识别和数据处理，以保证场站设备的正常运行。

（5）智能识别：对采集仪表图像进行预处理，提取相关仪器仪表读数数据并进行存储与分析，通过后台 AI 智能识别模块进行安全隐患的自动识别与报警，进而完成环境中安全事故的识别。

5.2　系统开发环境

本系统的运行环境配置情况如表 5.1 所示。

表 5.1 运行环境配置

Table 5.1 Operating environment configuration

序号	名称	配置信息
1	CPU 型号	Intel（R）Core（TM）i7-10700 CPU @ 2.9GHz
2	CPU 内存	32GB
3	CPU 核心数	8
4	GPU 型号	Nvidia RTX 2080S Super
5	3D Studio Max	2017
6	Unity3D	2018.4.35f1

5.3　总体架构设计

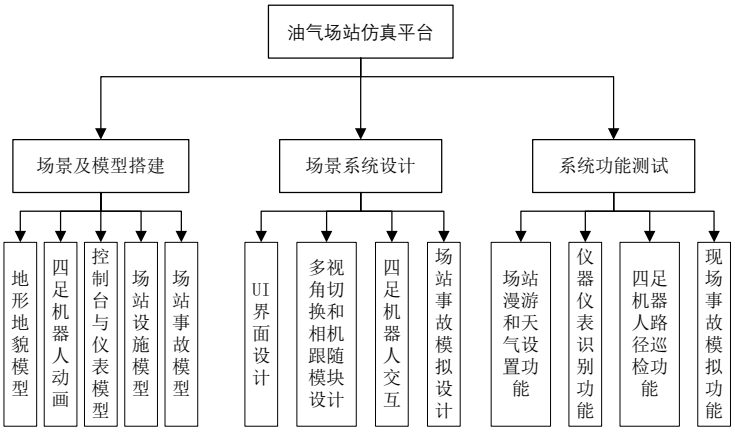


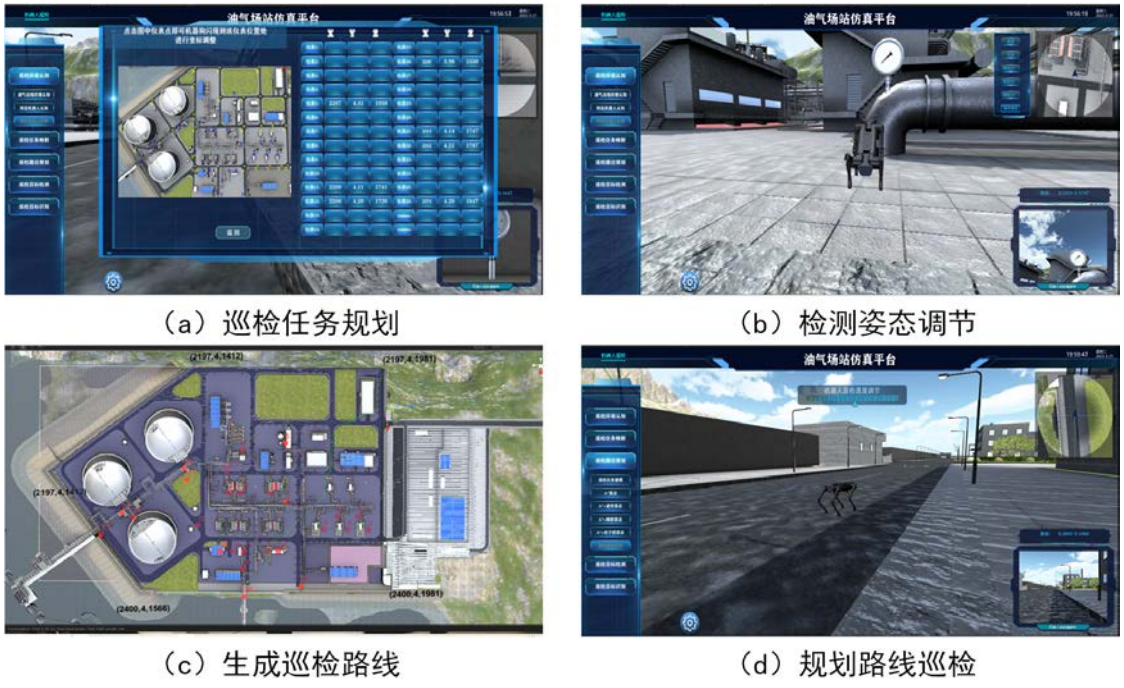
图 5.1 油气场站仿真平台总体架构图

Fig. 5.1 Overall visual interface Overall architecture diagram of simulation platform of oil and gas station

分析油气场站仿真平台的需求和功能，设计包括场景和模型的构建、视景系统的设计以及实验案例测试。为了使用户能够通过浏览器直接访问该系统平台，该虚拟仿真系统采用了 WebGL 技术，并将其部署到本地局域网服务器上。考虑到系统的扩展性和易维护性，采用模块化编程思想将系统划分为多个部分。该系统架构如图 5.1 所示。

5.4 系统概要设计

该系统采用了流式的开发模型，具体的运行流程为：首先，通过 SolidWorks 和三维重建技术完成场站的实体建模，并在 Unity3D 场景中对具体的设备进行三维建模。接下来，搭建后端相关的 AI 智能识别模块，实现火灾报警、泄漏识别和仪表识别等功能。最后，利用主流的网络编程通信技术来完成虚拟油气场站仿真平台的搭建。



（a）巡检任务规划　（b）检测姿态调节

（c）生成巡检路线　（d）规划路线巡检

图 5.2 巡检流程图

Fig. 5.2 Inspection flow chart

系统登录进入后，便可进入功能主页面。在开发过程中，使用 UGUI 设计了系统的 UI 界面，系统左侧目录采用三级菜单的结构设计，其中点击“巡检任务映射”可选择目标巡检点，并设置相关坐标和姿态。一旦完成选择，通过 A 星算法生成

最优路径，开始巡检并可自行调节四足机器人行走速度，并将巡检数据存储在数据库中保存。巡检过程如图 5.2 所示。

系统操作界面如图 5.3 所示，本工作系统主要分为四个部分，分别是：巡检环境认知、巡检路径规划、巡检目标检测和天气设置。首先，用户可以通过巡检环境感知进行场站进行漫游，并设置需要进行巡检的目标点。该场景中以四足机器人的第三人称视角进行漫游，并在右下角显示其第一人称视角，四足机器人可通过键盘和鼠标进行姿态控制和视角的转换。其次可以设定目标巡检点并利用本系统的各种路径规划算法来进行巡检，最后为了验证视觉 SLAM 算法在室外不同天气和时间的适用性，开发了天气设置模块，可以模拟各种天气情况。

图 5.3 整体可视化界面

Fig. 5.3 Overall visual interface

5.5　软件实现设计

在系统的巡检过程中，首先需要通过场景漫游设定巡检目标位置，由于油气场站范围广、生产设备多、生产工艺复杂，工作人员可能无法对整个油气场站实时监控，因此，油气场站仿真平台通过三维可视化技术使工作人员实时观察油气场站环境和设备。同时为了满足工作人员多视角观察油气场站运行情况，平台实现了对油

气场站运行过程的无死角观察，通过缩放、平移和旋转场景，用户可以实现对所显示信息的更有效的解读。使工作人员能够更好更有效地接收系统展示的信息。并且可以根据天气设置模块进行天气的设置，如图 5.4 所示。在每次巡检行驶流程中，平台可设定四足巡检机器人在规划路线下完成自动巡查漫游任务，还具有拖动速度控制条来调节四足机器人的行驶速度，控制键盘来观测油气场站中设备坐标信息，切换查看环境视角或暂停查看设备的功能。

图 5.4 天气场景图

Fig. 5.4 Weather scene map

图 5.5 巡检识别效果图

Fig. 5.5 Inspection identification effect diagram

在设定完成巡检目标后，进入路径规划模块，提供蚁群算法、A 星算法和迪杰斯特拉算法等多种路径规划算法进行选择，并生成最优路径。识别效果如图 5.5 所示，虚拟场站的四足机器人按照规定巡检路线进行巡检，并在巡检过程中进行目标检测。将识别出的仪表读数导入后台数据库中记录，并且搜集巡检过程中的视频图像，作为所提出的视觉 SLAM 系统测试的数据集。利用 Unity3D 的粒子系统和第三方的插件，场站中还设置了相关事故发生模拟功能，如火灾，石油泄漏，道路维修等，模拟效果如图 5.6 所示。

（a）石油泄漏　（b）火灾模拟　（c）地面塌陷

图 5.6 事故模拟图

Fig. 5.6 Accident simulation diagram

5.6　本章小结

本章详细描述了油气场站仿真平台的设计与实现。首先，简要阐述了油气场站仿真平台的业务需求及功能模块。其次，对系统的整体架构进行了描述，并详细介绍了油气场站仿真平台所需的各个功能模块。最后，依据上述设计进行系统开发，并展示了油气场站仿真平台的界面。

第 6 章　总结与展望

6.1　总结

全文以复杂环境下的视觉 SLAM 技术为研究对象，针对该环境下存在轨迹误差大、鲁棒性差等问题，提出对点线两种特征和动态环境下 SLAM 技术的原理和关键技术进行讨论和研究。论文的主要研究工作如下：

（1）首先介绍了本研究的相关研究背景及重要研究意义，并阐述了国内外视觉 SLAM 与深度学习技术的研究现状。然后详细描述了视觉 SLAM 的相关理论和每个模块的原理。最后对融合深度学习的动态视觉 SLAM 系统进行了研究和分析。

（2）围绕目前使用经典视觉 SLAM 系统对于工厂环境下存在轨迹误差大、鲁棒性差等问题，本文提出一种基于点线感知的异构图注意力视觉 SLAM 算法，通过点线的几何关系同步特征提取网络 PL-Net 实现了两类特征的并行关联提取，并通过感知迁移知识蒸馏策略对网络模型进行优化，提高了特征提取的准确率和实时性。为了提升图像点线匹配精度，提出了一种点线异构图注意力网络，利用边缘聚集图注意模块和交叉异构图迭代模块分别实现图内和图间的特征聚合进而提高几何特征的表达能力。最后，将点线匹配问题转换为一个最优化问题，构建基于贪婪策略的最邻近点迭代方法对该优化问题进行求解。通过实验数据证明了该算法可以在相关领域得到广泛应用，并取得良好的效果。

（3）针对动态环境下由于动态目标干扰而导致估计的位姿误差急剧增大进而失效的问题，提出了一种面向动态环境中的实时语义 SLAM 系统。为了降低计算量，该算法只对关键帧进行语义分割，以去除已知的动态对象，并保持一个静态建图环境，以进行鲁棒的相机跟踪。在此基础上，首先利用深度图像来获取场景中的物体信息，然后将深度图像聚类到几个区域，将相似的物体信息合并，从而识别出场景中的动态区域。进一步利用重投影误差来检测出物体的运动情况，从而实现对未知运动物体的识别。在公共数据集和真实世界条件下对该方法进行了评价。实现了在低功耗嵌入式平台上实时运行的语义 SLAM 系统，并在动态环境中提供高定位精度。

（4）为了验证所提出的视觉 SLAM 系统性能，将通过模拟不同场景下的视觉数据来测试系统的鲁棒性和准确性，以便更深入地了解系统在不同环境下的表现，以 Unity3D 为开发平台，实现了一个油气场站仿真平台系统。结合现实场景中存在

的业务需求，设计了该系统的实现流程。最后对该系统的主要功能进行了测试，展示了该系统的测试结果。

6.2　展望

论文在视觉 SLAM 等方面进行了分析与探索，并提出了视觉 SLAM 与深度学习融合的相关研究框架，相较于 ORB-SLAM2 算法，研究所提出的位姿估计方法有较大的提升，但研究内容仍然存在一些问题需要解决。未来研究工作可以从以下几个方面进行深入探索：

（1）在潜在移动目标分割模块，论文选取 Topformer 算法，该算法的实时性和精度都能满足 SLAM 系统的要求，但是算法实时性上还有进一步的提升空间，在尺度语义感知提取方面，该算法使用以多头自注意力机制为基础的 Transformer 模块，由于自注意力计算时间复杂度高，因此对硬件要求较高，未来可以考虑优化 Transformer 的结构，采用新型的注意力结构替换自注意力层，以减少计算量。

（2）在特征匹配阶段，所提出的 GIPOT 算法引入了贪婪和最邻近结合的思想，这种方法可以有效的提高传输的质量和结果的精确度，然而，使用熵正则化求解最优传输时，也面临着许多困难。除此以外，至今没有一种更为有效的方式来实现特征向量之间的传输，因此，这也成为未来研究的一个热点方向。

（3）在面向动态环境的语义 SLAM 系统中，研究通过分割结果剔除了动态特征点对于定位和建图的影响，但如果场景都是动态的，那么由于缺乏静态特征，目前的方法无法获得准确的结果。在后续工作中，考虑利用动态对象的信息来协助位姿估计和实时重建。

参 考 文 献

[1] KAZEROUNI I A, FITZGERALD L, DOOLY G, et al. A Survey of State-of-the-Art on Visual SLAM[J]. Expert Systems with Applications, 2022, 205: 117734-117751.

[2] SU T, ZHU H, ZHAO P, et al. A Robust LiDAR-Based SLAM for Autonomous Vehicles Aided by GPS/INS Integrated Navigation System[C]//6th International Conference on Automation, Control and Robotics Engineering. Piscataway: IEEE, 2021: 351-358.

[3] 马鑫, 梁新武, 蔡纪源. 基于点线特征的快速视觉 SLAM 方法[J]. 浙江大学学报(工学版), 2021, 55(02): 402-409.

[4] ARUN A, AYYALASOMAYAJULA R, HUNTER W, et al. P2slam: Bearing Based Wifi SLAM for Indoor Robots[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 3326-3333.

[5] SINGANDHUPE A, LA H M. A Review of SLAM Techniques and Security in Autonomous Driving[C]//Third IEEE International Conference on Robotic Computing. Piscataway: IEEE, 2019: 602-607.

[6] YE X, JI X, SUN B, et al. DRM-SLAM: Towards Dense Reconstruction of Monocular SLAM with Scene Depth Fusion[J]. Neurocomputing, 2020, 396: 76-91.

[7] GUPTA A, FERNANDO X. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges[J]. Drones, 2022, 6(4): 85-120.

[8] 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.

[9] NEWCOMBE R A, LOVEGROVE S J, DAVISON A J. DTAM: Dense Tracking and Mapping in Real-Time[C]//International Conference on Computer Vision. Piscataway: IEEE, 2011: 2320-2327.

[10] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: Large-Scale Direct Monocular SLAM[C]//13th European Conference on Computer Vision. Heidelberg: Springer International Publishing, 2014: 834-849.

[11] FORSTER C, PIZZOLI M, SCARAMUZZA D. SVO: Fast Semi-Direct Monocular Visual Odometry[C]//IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2014: 15-22.

[12] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-Time Single Camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 1052-1067.

[13] KLEIN G, MURRAY D. Parallel Tracking and Mapping for Small AR Workspaces[C]//6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Piscataway: IEEE, 2007: 225-234.

[14] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. Orb-slam3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[15] ROSTEN E, DRUMMOND T. Machine Learning for High-speed Corner Detection[C]//9th European Conference on Computer Vision, Heidelberg: Springer, 2006: 430-443.

[16] ROSTEN E, PORTER R, DRUMMOND T. Faster and Better: A Machine Learning Approach to Corner Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 32(1): 105-119.

[17] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: An Efficient Alternative to SIFT or SURF[C]//International Conference on Computer Vision. Piscataway: IEEE, 2011: 2564-2571.

[18] TATENO K, TOMBARI F, LAINA I, et al. Cnn-SLAM: Real-time Dense Monocular SLAM with Learned Depth Prediction[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017: 6243-6252.

[19] DETONE D, MALISIEWICZ T, RABINOVICH A. Superpoint: Self-Supervised Interest Point Detection and Description[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. Piscataway: IEEE, 2018: 224-236.

[20] SARLIN P E, DETONE D, MALISIEWICZ T, et al. Superglue: Learning Feature Matching with Graph Neural Networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 4938-4947.

[21] TANG J, ERICSON L, FOLKESSON J, et al. GCNv2: Efficient Correspondence

Prediction for Real-Time SLAM[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3505-3512.

[22] TANG J, FOLKESSON J, JENSFELT P. Geometric Correspondence Network for Camera Motion Estimation[J]. IEEE Robotics and Automation Letters, 2018, 3(2): 1010-1017.

[23] BOWMAN S L, ATANASOV N, DANIILIDIS K, et al. Probabilistic Data Association for Semantic SLAM[C]//IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2017: 1722-1729.

[24] KANEKO M, IWAMI K, OGAWA T, et al. Mask-SLAM: Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. Piscataway: IEEE, 2018: 258-266.

[25]YU C, LIU Z, LIU X J, et al. DS-SLAM: A Semantic Visual SLAM Towards Dynamic Environments[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2018: 1168-1174.

[26] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[27] 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.

[28] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-Cnn[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 2961-2969.

[29] BAHRAINI M S, RAD A B, Bozorg M. SLAM in Dynamic Environments: A Deep Learning Approach for Moving Object Tracking Using ML-RANSAC Algorithm[J]. Sensors, 2019, 19(17): 3699-3719.

[30] GIRSHICK R. Fast R-Cnn[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway: IEEE, 2015: 1440-1448.

[31] MACARIO BARROS A, MICHEL M, MOLINE Y, et al. A Comprehensive Survey of Visual SLAM Algorithms[J]. Robotics, 2022, 11(1): 24-51.

[32] SUN Y, LIU M, MENG M Q H. Improving RGB-D SLAM in Dynamic Environments:

A Motion Removal Approach[J]. Robotics and Autonomous Systems, 2017, 89: 110-122.

[33] WANG Y, HUANG S. Towards Dense Moving Object Segmentation Based Robust Dense RGB-D SLAM in Dynamic Scenarios[C]//13th International Conference on Control Automation Robotics & Vision. Piscataway: IEEE, 2014: 1841-1846.

[34] DU Z J, HUANG S S, MU T J, et al. Accurate Dynamic SLAM Using CRF-Based Long-Term Consistency[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 28(4): 1745-1757.

[35] LI A, WANG J, XU M, et al. DP-SLAM: A Visual SLAM with Moving Probability Towards Dynamic Environments[J]. Information Sciences, 2021, 556: 128-142.

[36] FERRERA M, MORAS J, TROUVÉ-PELOUX P, et al. Real-Time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments[J]. Sensors, 2019, 19(3): 687-706.

[37] LIU G, ZENG W, FENG B, et al. DMS-SLAM: A General Visual SLAM System for Dynamic Scenes with Multiple Sensors[J]. Sensors, 2019, 19(17): 3714-3734.

[38] BIAN J W, LIN W Y, MATSUSHITA Y, et al. Gms: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017: 4181-4190.

[39] VIDAL A R, REBECQ H, HORSTSCHAEFER T, et al. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios[J]. IEEE Robotics and Automation Letters, 2018, 3(2): 994-1001.

[40] HASHIM H A, ELTOUKHY A E E. Landmark and Imu Data Fusion: Systematic Convergence Geometric Nonlinear Observer for SLAM and Velocity bias[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(4): 3292-3301.

[41] SONG S, LIM H, LEE A J, et al. DynaVINS: A Visual-Inertial SLAM for Dynamic Environments[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 11523-11530.

[42] AGUDO A. Unsupervised 3D Reconstruction and Grouping of Rigid and Non-Rigid Categories[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(1): 519-532.

[43] AGUDO A, MORENO-NOGUER F, CALVO B, et al. Real-Time 3D Reconstruction of Non-Rigid Shapes with A Single Moving Camera[J]. Computer Vision and Image

Understanding, 2016, 153: 37-54.

[44] CRUZ-MOTA J, BOGDANOVA I, PAQUIER B, et al. Scale Invariant Feature Transform on the Sphere: Theory and Applications[J]. International Journal of Computer Vision, 2012, 98: 217-241.

[45] VISWANATHAN D G. Features from Accelerated Segment Test (Fast)[C]//Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services. Piscataway: IEEE, 2009: 6-8.

[46] MUJA M, LOWE D G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration[J]. International Conference on Computer Vision Theory and Applications, 2009, 2: 331-340.

[47] LOWE D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[48] FISCHLER M A, BOLLES R C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.

[49] RAGURAM R, FRAHM J M, POLLEFEYS M. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus[C]//European Conference on Computer Vision. Heidelberg: Springer, 2008: 500-513.

[50] YE X, MA J, XIONG H. Local Affine Preservation with Motion Consistency for Feature Matching of Remote Sensing Images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2021, 60: 1-12.

[51] BARROSO-LAGUNA A, MIKOLAJCZYK K. Key. Net: Keypoint Detection by Handcrafted and Learned Cnn Filters Revisited[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1): 698-711.

[52] YI K M, TRULLS E, ONO Y, et al. Learning to Find Good Correspondences[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 2666-2674.

[53] RANFTL R, KOLTUN V. Deep Fundamental Matrix Estimation[C]//Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 284-299.

[54] BRACHMANN E, ROTHER C. Neural-Guided RANSAC: Learning where to Sample Model Hypotheses[C]//Proceedings of the IEEE/CVF International Conference on

Computer Vision. Piscataway: IEEE, 2019: 4322-4331.

[55] ZHANG J, SUN D, LUO Z, et al. Learning Two-View Correspondences and Geometry Using Order-Aware Network[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 5845-5854.

[56] SUN J, SHEN Z, WANG Y, et al. LoFTR: Detector-Free Local Feature Matching with Transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 8922-8931.

[57] LI S, XU C, XIE M. A Robust O (n) Solution to The Perspective-N-Point Problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1444-1450.

[58] CHETVERIKOV D, SVIRKO D, STEPANOV D, et al. The Trimmed Iterative Closest Point Algorithm[C]//Object Recognition Supported by User Interaction for Service Robots. Piscataway: IEEE, 2002, 3: 545-548.

[59] Sivic J, Zisserman A. Video Google: A Text Retrieval Approach to Object Matching in Videos[C]//IEEE International Conference on Computer Vision. Piscataway: IEEE Computer Society, 2003, 3: 1470-1470.

[60] NISTER D, STEWENIUS H. Scalable Recognition with A Vocabulary Tree[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2006, 2: 2161-2168.

[61] 高兴波，史旭华，葛群峰，陈奎烨. 面向动态物体场景的视觉SLAM综述[J]. 机器人，2021，43(06)：733-750.

[62] 于雅楠，卫红，陈静. 基于局部熵的 SLAM 视觉里程计优化算法[J]. 自动化学报，2021，47(6)：1460-1466.

[63] 范涵奇，吴锦河. 基于拉普拉斯分布的双目视觉里程计[J]. 自动化学报，2022，48(3)：865-876.

[64] LEE T, KIM C, CHO D D. A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots[J]. IEEE Transactions on Industrial Electronics, 2018, 66(1): 318-328.

[65] QIN T, LI P, SHEN S. Vins-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.

[66] ZHANG L, KOCH R. An Efficient and Robust Line Segment Matching Approach

Based on LBD Descriptor and Pairwise Geometric Consistency[J]. Journal of Visual Communication and Image Representation, 2013, 24(7): 794-805.

[67] CHO H, KIM E K, KIM S. Indoor SLAM Application Using Geometric and ICP Matching Methods Based on Line Features[J]. Robotics and Autonomous Systems, 2018, 100: 206-224.

[68] 夏琳琳, 崔家硕, 宋梓维, 等. 线特征优化配置在室内 RGB-D SLAM 系统中的应用[J]. 中国惯性技术学报, 2022, 30(06): 760-767+776.

[69] ZHOU D, DAI Y, LI H. Ground-Plane-Based Absolute Scale Estimation for Monocular Visual Odometry[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(2): 791-802.

[70] YANG S, SCHERER S. Monocular Object and Plane SLAM in Structured Environments[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3145-3152.

[71] CHO H G, YEON S, CHOI H, et al. Detection and Compensation of Degeneracy Cases for Imu-Kinect Integrated Continuous SLAM with Plane Features[J]. Sensors, 2018, 18(4): 935.

[72] GOMEZ-OJEDA R, MORENO F A, ZUNIGA-NOËL D, et al. PL-SLAM: A Stereo SLAM System through the Combination of Points and Line Segments[J]. IEEE Transactions on Robotics, 2019, 35(3): 734-746.

[73] GRANT W S, VOORHIES R C, ITTI L. Efficient Velodyne SLAM with Point and Plane Features[J]. Autonomous Robots, 2019, 43(5): 1207-1224.

[74] SUN Q, YUAN J, ZHANG X, et al. Plane-Edge-SLAM: Seamless Fusion of Planes and Edges for SLAM in Indoor Environments[J]. IEEE Transactions on Automation Science and Engineering, 2020, 18(4): 2061-2075.

[75] LI H, YANG Y, CHEN D, et al. Optimization Algorithm Inspired Deep Neural Network Structure[C]//Asian Conference on Machine Learning. New York: PMLR, 2018: 614-629.

[76] DUAN K, BAI S, XIE L, et al. Centernet: Keypoint Triplets for Object Detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 6569-6578.

[77] ZHANG H, LUO Y, QIN F, et al. ELSD: Efficient Line Segment Detector and Descriptor[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2021: 2969-2978.

[78] HINTON G, VINYALS O, DEAN J. Distilling the Knowledge in A Neural Network[J]. arXiv preprint arXiv: 1503. 02531, 2015.

[79] ZHANG K, ZHANG C, LI S, et al. Student Network Learning via Evolutionary Knowledge Distillation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(4): 2251-2263.

[80] ZAGORUYKO S, KOMODAKIS N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer[J]. arXiv preprint arXiv: 1612. 03928, 2016.

[81] VALVERDE F R, HURTADO J V, VALADA A. There Is More than Meets the Eye: Self-Supervised Multi-Object Detection and Tracking with Sound by Distilling Multimodal Knowledge[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 11612-11621.

[82] CUTURI M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport[J]. Advances in Neural Information Processing Systems, 2013, 26: 1-9.

[83] XIE Y, WANG X, WANG R, et al. A Fast Proximal Point Method for Computing Exact Wasserstein Distance[C]//Uncertainty in Artificial Intelligence. New York: PMLR, 2020: 433-453.

[84] HUANG K, WANG Y, ZHOU Z, et al. Learning to Parse Wireframes in Images of Man-Made Environments[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 626-635.

[85] DENIS P, ELDER J H, ESTRADA F J. Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery[C]//10th European Conference on Computer Vision. Heidelberg: Springer, 2008: 197-210.

[86] 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.

[87] SCHUBERT D, GOLL T, DEMMEL N, et al. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2018: 1680-1687.

[88] WHELAN T, SALAS-MORENO R F, GLOCKER B, et al. ElasticFusion: Real-Time Dense SLAM and Light Source Estimation[J]. The International Journal of Robotics Research, 2016, 35(14): 1697-1716.

[89] ZHANG W, HUANG Z, LUO G, et al. TopFormer: Token Pyramid Transformer for Mobile Semantic Segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2022: 12083-12093.

[90] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft Coco: Common Objects In Context[C]//13th European Conference on Computer Vision. Heidelberg: Springer, 2014: 740-755.

[91] BAKER S, MATTHEWS I. Lucas-Kanade 20 Years on: A Unifying Framework[J]. International Journal of Computer Vision, 2004, 56: 221-255.

[92] STURM J, ENGELHARD N, ENDRES F, et al. A Benchmark for the Evaluation of RGB-D SLAM Systems[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2012: 573-580.