

分 类 号 _____

学号 M201570380

学校代码 10487

密级 _____

华中科技大学

硕士学位论文

基于点线综合特征的单目 SLAM 方法研究

学位申请人： 兰莎娜

学 科 专 业： 机械电子工程

指 导 老 师： 熊良才 副教授

答 辩 日 期： 2018 年 5 月

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science**

**A Monocular Visual SLAM Method Research Based
on Point-Line Feature Fusion**

Candidate : Lan Shayun

Major : Mechatronics Engineering

Supervisor : Associate Professor Xiong Liangcai

Huazhong University of Science and Technology

Wuhan, Hubei 430074, P. R. China

May, 2018

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权用户许可证书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：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电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密☐，在_____年解密后适用本授权书。

本论文属于

不保密☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

摘要

传统的视觉 SLAM（同时定位与建图）大多是基于特征点来估计相机的位姿，同时建立环境地图。然而在低纹理场景下，该类方法往往难以提取到足够的特征点，而且在快速动态环境下，由于视角的大幅变化，点特征匹配误差较大，容易出现跟踪丢失的情况。为弥补视觉 SLAM 中的上述不足，本文提出了综合处理点线特征的单目视觉 PL-SLAM，并在基于点特征开源的 ORB-SLAM 基础上完成了理论相关的代码实现。

本文首先简单介绍了单目视觉 SLAM 的原理和方法，并在 ORB-SLAM 基础上扩展了 PL-SLAM 的整体框架体系。重点论述了线特征的提取和匹配，从线特征的表达方式入手，利用线特征的对极约束，创新性的解决了空间线的三维重建问题；建立线特征的误差模型，并基于李代数的扰动模型推导了误差模型中关于相机位姿、线特征的雅克比矩阵。

其次详细阐述了 PL-SLAM 中跟踪线程和局部建图线程的工程实现，线特征与点特征是两种不同的图像基元，在 SLAM 体系中的处理流程相似，具体实现方法完全不同，因此详细说明引入线特征后每个流程的实现，重点介绍了基于误差模型及其雅克比矩阵在后端优化中的工程实现。

最后，本文采用 TUM 中的 RGB-D 数据集从定位精度和计算时间对 PL-SLAM 进行了评估。实验结果表明，在相机快速运动和场景中存在动态障碍物时，综合点、线特征的单目视觉 PL-SLAM 系统具有较好的精度和鲁棒性。由于线特征提取相比点特征提取较为耗时，本文的 SLAM 方法处理图像的帧率下降，但能基本达到实时状态。

关键词：单目视觉 SLAM，线特征，图优化

Abstract

Traditional visual SLAM(Simultaneous Localization and Mapping) estimates the pose of camera mostly based on feature points and builds environmental map. However, this method is often difficult to extract enough feature points under the low texture scene. The error of the point feature matching is large and the tracking loss is easy to appear because of the large change of the view angle in the fast-dynamic environment. In order to make up the above shortcomings in visual SLAM, this paper proposes a monocular vision PL-SLAM which combines the feature of point and line. Completes the code implementation on the basis of the ORB-SLAM which is based on the point feature.

Firstly, this paper briefly introduces the principle and method of monocular vision SLAM, then extend the PL-SLAM on the basis of ORB-SLAM. And discuss the extraction and matching of line features in detail. Innovatively solve the three-dimensional reconstruction of lines using the line features to polar constraints and establish the error model of line features with the expression of line features. The Jacobian matrix of camera pose and line characteristics to the error model is derived based on the disturbance model of Lie algebra.

Secondly, describe the engineering implementation of Tracking thread and Local Mapping thread in detail. The line feature and point feature are two different image primitives. The processing flow in the SLAM system is similar but the specific implementation method is completely different. Therefore, the implementation of each process is described in detail after the line features are introduced. The emphasis is the engineering implementations of the back-end optimization on the error model and its Jacobian matrix.

Finally, the RGB-D dataset in TUM is used to evaluate the PL-SLAM from location accuracy and computation time. The experimental results show that the monocular vision PL-SLAM system with integrated point and line features has better accuracy and robustness when the dynamic obstacle exists in the fast motion and scene of the camera. Because line feature extraction is time-consuming compared with point feature extraction, the SLAM method in this paper deals with image frame rate degradation, but it can basically achieve real-time state.

Keywords: Monocular Visual SLAM, Line Segment Features, Graph Optimization

目 录

摘 要.....	I
Abstract	II
目 录	III
1 绪论	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.3 本文主要内容.....	5
2 单目 SLAM 原理和综合框架	7
2.1 前端视觉里程计.....	7
2.2 后端优化	14
2.3 闭环检测	18
2.4 综合框架	19
2.5 本章小结	20
3 视觉 SLAM 中的线特征	21
3.1 线特征的提取与匹配.....	21
3.2 线特征的表达方式.....	27
3.3 线特征的三维重建.....	28
3.4 线特征的误差模型.....	31

华中科技大学硕士学位论文

3.5 点线特征的综合误差模型	35
3.6 本章小结	37
4 点线综合特征的单目 SLAM	38
4.1 跟踪线程	38
4.2 局部建图线程.....	43
4.3 后端优化	44
4.4 本章小结	47
5 实验与分析.....	48
5.1 实验环境	48
5.2 基于标准数据集的实验分析	48
5.3 本章小结	56
6 总结与展望.....	57
6.1 全文总结	57
6.2 论文展望	57
致 谢	59
参考文献	60

1 绪论

1.1 研究背景与意义

随着科技的快速发展，移动机器人在工业生产领域和我们的日常生活中越来越常见，比如车间中的物流机器人，家用扫地机器人，具有辅助驾驶功能的汽车等。对于这些广义机器人，实现其功能需要其具备自主移动能力和环境适应能力。

自主移动机器人在环境中自主移动时，需要知道外界环境的信息以及机器人自身在环境中的位置和姿态。而在一个未知的环境里，环境地图信息事先并不知道，属于先验信息，则机器人就只能通过自身携带的传感器在未知环境下进行定位，同时根据传感器的信息来建立环境地图，这就是同时定位与建图（SLAM, Simultaneous Localization and Mapping）的过程^[1]。这意味着机器人在创建地图时，机器人必须在已经建立的地图中精确定位，从而根据新的传感器信息扩展地图。地图创建和定位的相互作用在 SLAM 中是至关重要的，如果定位出现问题，根据新的传感器信息建立的地图就不能和已经建立的地图保持全局一致；同时，如果不能正确建立环境地图，机器人采集的传感器信息不能和从地图估计的位姿信息对应，从而出现错误定位。

机器人采用的传感器通常有两大类^[2, 3]：一种用于感知载体本身的运动信息，常见的有加速度传感器、角速度传感器等，如很多机器人采用的惯性测量单元就包含了加表和陀螺仪；另外一种用于感知外界的环境信息，如激光雷达、摄像头等传感器。其中，激光雷达可用于直接测量传感器周围的距离信息，如百度无人车采用的 64 线激光雷达，米家扫地机器人采用的单线激光雷达，若使用这类传感器的 SLAM 多使用概率估计^[4]的方法，常用的有卡尔曼滤波，粒子滤波法等。另一类传感器主要是相机，由于视觉类传感器无法直接获取距离信息，因此计算量较距离类传感器大，使用滤波器算法的视觉 SLAM 往往会受制于场景大小^[5]。而最近几年，由于图优化中海森矩阵稀疏性的发现，视觉 SLAM 开始采用图优化的方法^[6]，并且能应用于大型场景。SLAM 问题的解决方法通常取决于感知外界信息的传感器种类和应用场景所施加的时间约束（如在线或是批量处理）。根据解决方法和传感器所获取的信息的不同，SLAM 所得到的地图在其维度（2D 或 3D）和环境地图表示（例如点云地图或占用栅格地图）上也有所不同。

虽然激光雷达这类传感器能直接获取距离信息，从而降低计算量，但是其价格昂贵，由于相机价格低廉，获取环境信息丰富，视觉 SLAM 已经越来越受到关注，但其面临的挑战也越来越多。例如，在低纹理场景下，点特征数量急剧减少，此时基于点特征的视觉 SLAM 不能稳定工作，但是在这类场景中，会出现很多线特征，尤其是在人造环境中。如何利用更多特征信息，使得机器人能够在低纹理场景中稳定工作，对于 SLAM 技术的发展和應用有重大意义。

1.2 国内外研究现状

视觉 SLAM 发展到现在，其基本框架已经成熟，整体可以分为前端和后端两个部分，整体框架如图 1-1 所示^[7,8]。前端用于收集传感器信息预估模型，后端用于对模型进行进一步的优化计算。

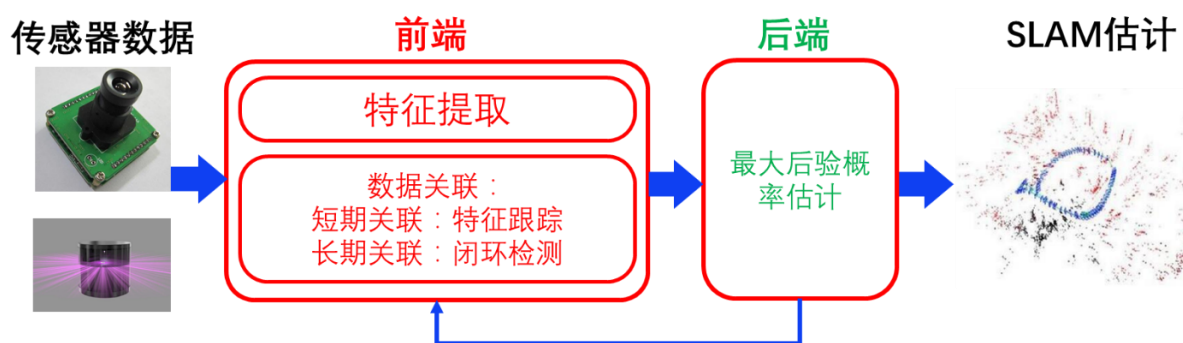


图 1-1 主流 SLAM 的整体框架

从前端分类角度而言，视觉 SLAM 主要分为特征法和直接法。基于特征法的视觉 SLAM 是检测图像的特征点，通过特征点匹配估算相机位姿和环境地图。基于直接法的 SLAM 是直接对图像的像素进行处理，通过最小化帧间的光度误差来计算相机位姿。

从后端角度分类，视觉 SLAM 主要分为滤波器法和基于关键帧的图优化方法。在视觉 SLAM 研究初期学者们主要利用滤波器来实现目标，核心是用状态向量表示相机位姿和路标，用概率密度函数描述状态向量，迭代计算状态方程和观测方程，更新状态向量的均值和方差。在三维重建领域，光束平差法（BA）是主流的算法，但人们普遍认为视觉 SLAM 中的路标点数量太多，对应的计算量太大，因此无法将 BA 应用到视觉 SLAM 中。近年来，研究者慢慢认识到了由于路标相对位姿数量较多，光束平差法具有很强的稀疏性，大大降低了求解的复杂度，进而使得视觉 SLAM 容易落地。

1.2.2 主流的视觉 SLAM

Davison 等人于 2003 年构建了第一个单目视觉 SLAM 系统^[9], 该方法的前端使用图像块特征, 后端采用扩展卡尔曼滤波(EKF)迭代计算环境地图和相机位姿, 之后 Davison 又对视觉 SLAM 的初始化、尺度漂移等问题进行了研究。然而扩展卡尔曼滤波器存在线性约束、误差累计、计算量不断增大等问题, 针对这一问题, Pupilli 等人将粒子滤波(Particle Filter, PF)应用到了 SLAM 的后端优化滤波器^[10], 粒子滤波器利用随机采样可建立不受限制的概率密度分布, 从而解决非线性问题, 但是该方法在大场景使用时所需的粒子数较多, 对计算机的运算性能要求较高。

基于关键帧的方法是视觉 SLAM 的一个里程碑, Georg Klein 和 David Murray 提出的 PTAM (Parallel Tracking and Mapping)^[11]是该类方法的一个代表, 该方法将跟踪任务和建图任务分别采用独立的线程进行, 这样可以使得计算耗时的优化过程不与 SLAM 的实时性相关联, 使得基于关键帧的优化方法在实时性和精度方面都有了很大的提高。PTAM 虽然在大场景应用和快速跟踪方面还存在不足, 但其将前端和后端在实现上进行分离, 独立为两个线程的框架, 为后续主流的视觉 SLAM 所采用。2015 年 Raul 沿用了 PATM 的框架, 并对各部分内容进行了改进, 首先采用 ORB 方法进行特征点的识别、匹配以及后续的闭环检测^[12], ORB 方法对特征点的识别鲁棒性好且时间短; 在单目初始化时, 根据重投影误差自适应选择单应性矩阵或者基础矩阵初始化相机的位姿和点的坐标; 增加了闭环检测线程, 提高了定位和建图的精度。

上述介绍的视觉 SLAM 均需要提取点特征, 并计算特征描述子, 由于特征点的描述子计算耗时, 而且在运动模糊时难以提取足够的特征点, 因此基于直接法的视觉 SLAM 诞生。直接法假设同一位置点的像素灰度值不变, 并且同一个图片的像素具有相同的运动, 然后最小化光度误差估计相机位姿。Engle 等人提出的 LSD-SLAM^[13]是直接法的典型方法, 该方法直接对图像进行配准, 同时采用滤波法估计半稠密地图, 通过基于关键帧结合相似变换的方法获取全局地图, 从而能够有效修正尺度漂移。此外, 该方法能够在 CPU 甚至智能手机上实时运行。Foester 等人提出的 SVO (半直接视觉里程计) 方法^[14]介于特征点法和直接法, 该方法是对图像中的特征点图像块进行直接匹配, 然后使用图优化估计相机位姿。

视觉 SLAM 得到大量学者的研究，各种新的方法也不断出现，但视觉 SLAM 的工程应用仍存在较多问题有待解决。上述基于点特征的视觉 SLAM，在场景光线、明暗等发生变化，或者场景没有足够的点特征的情况下，直接会影响相机的位姿估计。直接法不依赖点特征，但计算量往往较大，对计算实时性挑战较大。而在现实人造的场景中，存在大量其他特征，如线特征。与点特征相比，线特征是一种层次属性更高的特征，同时线特征构建的地图也会更加直观易读，因此如何构建包含线特征的视觉 SLAM，提高其鲁棒性也就成为了视觉 SLAM 的一个重要课题。

1.2.3 包含线特征的视觉 SLAM

在视觉 SLAM 中，点特征和线特征是互补的，如图 1-2 所示，对同一张图片提取特征时，点特征数量较少的地方存在很多线特征。虽然点特征的相关算法已经较为成熟，但是其容易受相机视角以及场景纹理的影响。而线特征对相机视角的变换和场景结构较为鲁棒，也具有较好的光照不变性，但是由于其检测不稳定，初始化和数据关联都比点特征麻烦，因此在视觉 SLAM 中的使用具有一定的挑战性。



图 1-2 点、线特征提取示意图

Paul Smith 等人于 2006 年将线特征集成在单目 EKF-SLAM 系统^[15]中，其中，线特征的检测采用假设-测试方法，将 Fast 角点和 Sobel 算子检测出来的边缘连接为线段，该方法能够快速检测线段，从而使 SLAM 系统达到实时性能。Eade 等人于 2009 年在基于粒子滤波算法的 SLAM 系统^[16]中添加边缘信息，使用线段中点和与边缘同向的单位向量表示边缘。该方法能够基本达到实时运行的状态，由于在初始化时设置了虚拟地标和相关参数，系统的鲁棒性有待提升。

2011 年, Guoxuan Zhang 等人提出了一种采用 2D 方法表述线特征的视觉 SLAM 方法^[17], 利用地面上的水平线和竖直线特征估计相机的运动和环境地图。其中竖直线采用其与地平面的 2D 交点表示, 使用反向深度进行参数化, 水平线采用其在地平面上的两个端点表示。最后, 将两种线特征整合到 EKF-SLAM 系统中。之后, Huizhong Zhou 等人于 2015 年提出了 StructSLAM^[18], 该方法在 EKF-SLAM 中综合使用了结构线特征和点特征。其中, 结构线包含了全局方向信息, 随着时间的变化, 其能够约束相机的方向, 从而消除累积定位误差, 并减小里程计的漂移。需要注意, 由于这两种方法高度依赖线特征, 因此只适用于非常结构化的场景。

以上所述的方法都是采用滤波器实现基于同时定位与建图, 之后, Guoxuan Zhang 等人于 2015 年又提出了基于线特征双目视觉 SLAM 系统^[19], 该系统采用线特征实现了相机位姿估计, 并用光束平差法实现了位姿优化, 文中采用了两种 3D 线特征的描述方式, 采用普吕克坐标对新观测到的线特征进行初始化和 3D 投影, 采用正交法进行后端图优化。同年, Yan Lu 采用 RGB-D 相机, 提出了综合点线特征的视觉里程计^[20], 该方法是将采用最小二乘法, 将图像上线段对应的三维点拟合成空间线, 从而构建环境地图。

1.3 本文主要内容

本文主要研究基于关键帧优化和点线特征的单目视觉 SLAM 系统, 在 ORB-SLAM 的基础上进行改进, 同时使用点特征和线特征优化相机位姿, 并构建包含点特征和线特征的环境地图。论文的组织架构如下:

第一章首先简单介绍本文的研究背景和意义, 对视觉 SLAM 的分类、框架、主流方法进行详细叙述, 详细介绍了目前基于点线特征的主流算法。

第二章针对基于特征点的单目视觉 SLAM, 详细分析了其各个模块, 通过对基于点特征的 SLAM 问题的分析, 说明了引入线特征需要解决的问题。

第三章研究了视觉 SLAM 中的线特征, 包括线特征的提取与匹配, 线特征的表达以及对线特征的误差模型。着重介绍了单目情况下线特征的三维重建过程, 推导出线特征在 BA 后端优化中的雅克比矩阵。

第四章结合前两章的内容, 对本文提出的 SLAM 系统的前端和后端进行具体设计,

华 中 科 技 大 学 硕 士 学 位 论 文

主要包括跟踪线程和局部建图线程。

第五章用标准数据集对本文提出的 SLAM 系统进行了测试，从定位精度和时间效率两个方面做了对比分析。

第六章总结全文工作，然后对研究内容的下一步工作进行了展望。

2 单目 SLAM 原理和综合框架

在视觉 SLAM 研究领域，单目 SLAM 因其传感器结构简单、价格低廉，获取图像信息丰富等特点，成为了目前该领域的研究热点。当前主流的视觉 SLAM 的框架主要分为前端视觉里程计、后端非线性优化、回环检测和建图几个部分。视觉里程计用于初步处理传感器数据，估算相机运动位姿，建立局部地图；后端接收前端估计的不同时刻的相机位姿以及回环检测的信息，对数据进行整体优化，从而得到全局一致的相机位姿和环境地图；回环检测判断机器人是否到达过先前的位置，并将回环信息传递给后端非线性优化。本章将对单目 SLAM 的原理进行详细介绍，然后在开源 ORB-SLAM 的基础上提出了基于点线综合特征的单目 SLAM 的框架，并指出引入线特征之后需要解决的相关问题。

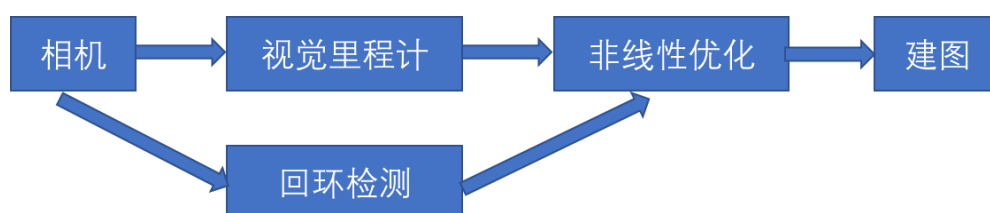


图 2-1 经典视觉 SLAM 原理框架

2.1 前端视觉里程计

单目视觉 SLAM 前端视觉里程计的作用是通过相邻图像的信息来估计每一时刻相机的位姿，并计算局部环境地图。首先对图像提取特征，再通过特征匹配找到相邻图像之间的对应关系，然后通过图像间的对极约束对帧间的位姿进行初步估计，最后建立局部地图。

2.1.1 特征提取

前端视觉里程计的主要问题是根据图像来估计相机运动。如果从图像的矩阵层面进行估计，计算量不仅大而且麻烦，因此可以在图像中选取代表性的点，也就是特征点。根据这些特征点特征来估计相机位姿，这些特征点在视觉 SLAM 中也称为路标。考虑到相机视角变化时，图像会发生旋转、亮度、光照等变化，在视觉里程计中，希望特征点在相机运动之后能够保持稳定。因此，一个好的特征点应该具有以下属性：

(1) 不变性：当图像的外部条件变化，如相机旋转平移、所在场景的光照环境变化时，仍然有特征点能被检测到；

(2) 独特性：不同的特征具有不同的表达，能够对不同的特征点进行检测和分类；

(3) 高效性：在一张图片中，像素的数量应该远大于特征点的数量；

(4) 重复性：相同场景对应的特征点应该能在不同的图像中检测到；

(5) 局部性：特征只和其所在的图像区域相关。

特征点包含关键点和描述子两部分，关键点是指特征点在图像中的位置，具有方向和大小等属性；描述子是一个向量，描述了关键点周围像素的信息。目前，常用的特征提取算法有 SIFT、SURF 和 ORB 等方法。

其中，SIFT 是最为经典的一种点特征^[21]，该算法使用高斯差分函数在尺度空间中快速寻找极值点，根据局部图像梯度确定关键点的方向，从而确定其位置、尺度和旋转不变性。SIFT 对外界环境的影响相当鲁棒，但其计算时间较其他特征点多，目前尚不能实时地在普通 CPU 上进行计算；SURF 是 SIFT 特征的加速版^[22]，其采用与 SIFT 相同的建立尺度空间的方法，在计算高斯图像的二阶偏导数时，使用二阶差分模板作为滤波器进行近似，从而缩短了特征点的提取时间，SIFT 和 SURF 的描述子均使用梯度统计直方图来描述。

ORB 特征^[23]不同于上两种，其对 FAST 角点^[24]进行改进，使其在能快速检测点特征的情况下，也能使特征点具有方向信息。随后再采用二进制描述子 BRIEF 对其进行描述，方便其后续进行特征匹配，但 ORB 特征和 SIFT、SURF 相比，精度和对环境变化的鲁棒性就相对弱化。ORB 能快速检测点特征的特征，使得以 ORB 为特征提取方法的 SLAM 可以在普通的 PC 上实时运行。

本文测试了 SIFT、SURF 和 ORB 提取特征和计算描述子的时间，如下表所示：

表2-1 不同特征点提取所耗时间

特征点类型	SIFT	SURF	ORB
时间，秒	0.616	0.184	0.027

2.1.2 特征匹配

视觉 SLAM 的前端用于解决数据关联问题，可以采用特征匹配来确定帧间特征点

的对应关系，即确定不同时刻 SLAM 的路标的对应关系。将图像之间或者图像与地图之间的描述子进行正确匹配，可以为后续的位姿估计、图优化等内容做准备工作，减小后续计算量。

特征匹配通常采用的方法有暴力匹配法（BFM）和快速近似最近邻法（FLANN）。暴力匹配法是计算第一张图像中每一个特征点与第二张图像中所有特征点的描述子之间的距离，然后选取最近的一个作为匹配点，该方法常用于帧间特征点匹配。当特征点数量较多时，暴力匹配法的计算量会很大，此时就可以采用快速近似最近邻法（FLANN），实现高效匹配。FLANN^[25]包含最合适最近邻搜索的算法集合，并且能够根据数据集自动选择最佳算法和最佳参数，所以在特征点数量非常大时，比如在环境地图中搜索对应某一帧的匹配时，快速最近邻算法就能保证实时性。

无论采用哪种特征匹配算法，均需要计算特征点所对应的描述子之间的距离，从而判断图像特征点之间的相近程度，常用的距离包括欧式距离、汉明距离和马氏距离等。

若两帧图像有对特征点，对应的两个描述子 L_1 和 L_2 ：

$$L_1 = (x_0, x_1, \dots, x_n) \quad (2-1)$$

$$L_2 = (x_0, x_1, \dots, x_n) \quad (2-2)$$

（1）欧式距离

欧式距离是计算两点的真实距离，是最为常用的距离度量方法之一，其计算方法为：

$$d(L_1, L_2) = \sqrt{(L_1 - L_2)(L_1 - L_2)^T} = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} \quad (2-3)$$

（2）汉明距离

汉明距离在主要用于评估二进制字符串类描述子，对于两个具有相同维度的描述子向量，其汉明距离就是对应位置元素不同的数量，其计算方法为：

$$d(L_1, L_2) = \sum_{i=0}^{255} (x_i \cap y_i) \quad (2-4)$$

（3）马氏距离

马氏距离常用于概率统计，表示数据的协方差距离，该距离能够有效计算两个未知样本集的相似度。与欧式距离相比，该方法考虑到数据各种属性间的联系，并且与尺度

无关，其计算方法为：

$$d(L_1, L_2) = \sqrt{(L_1 - L_2) \Sigma^{-1} (L_1 - L_2)^T} \quad (2-5)$$

其中， Σ 为协方差矩阵。

2.1.3 帧间位姿跟踪

帧间位姿跟踪是指当相机在场景中运动时，通过相机获取的图像信息计算相机在每一时刻的位姿，从而获取相机的运动轨迹，具体的方法是通过对极几何恢复帧间的相机运动。

2.1.3.1 对极几何

在经过图像的特征点检测与匹配之后，就能获得两张图像之间匹配好的特征点对，可以通过对极几何^[26]得到两张图像之间的位姿变化。如图 2-2 所示，现在要求取两帧图像之间的运动，设第一帧到第二帧的运动为 \mathbf{R} 、 \mathbf{t} ， O_1 与 O_2 是两帧的相机中心， p_1 与 p_2 是空间同一点 P 在两张图像上的对应点。 O_1 、 O_2 与 P 三点确定一个平面，称之为极平面。 O_1O_2 连线与像平面的交点 e_1 、 e_2 称为极点， O_1O_2 称为基线。极平面与两个像平面的交线 l_1 、 l_2 称为极线。从图中可知，射线 O_1p_1 是像平面上像素所对应的空间点 P 可能出现的位置，对应到像平面上，射线在像平面上的投影是极线，也就是空间点 P 在像平面上可能出现的投影位置。现在，通过特征点匹配，可以确定的像素位置，因此可以推断出 P 的空间位置，以及相机的运动。

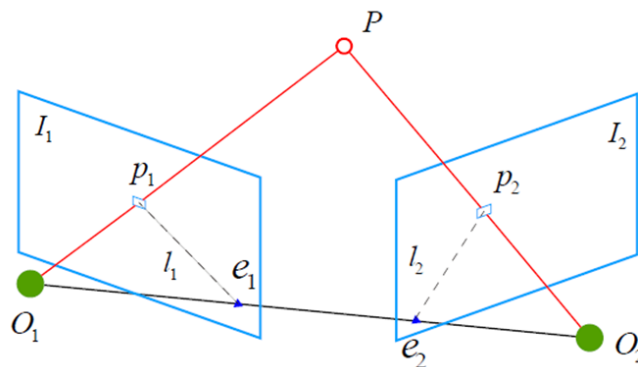


图 2-2 对极几何约束

2.1.3.2 约束矩阵

(1) 基础矩阵和本质矩阵

在第一帧坐标系下，设 \mathbf{P} 的空间位置为：

$$\mathbf{P} = (X, Y, Z)^T \quad (2-6)$$

根据相机的针孔模型^[27]，可得两个像素点 p_1 、 p_2 的像素坐标满足以下关系：

$$s_1 \mathbf{p}_1 = \mathbf{K} \mathbf{P}, s_2 \mathbf{p}_2 = \mathbf{K}(\mathbf{R} \mathbf{P} + \mathbf{t}) \quad (2-7)$$

其中， \mathbf{K} 是相机内参矩阵， \mathbf{R} 、 \mathbf{t} 为第二帧相机的运动。如果使用齐次坐标，也可以把上式写成在乘以非零常数下成立的等式：

$$\mathbf{p}_1 = \mathbf{K} \mathbf{P}, \mathbf{p}_2 = \mathbf{K}(\mathbf{R} \mathbf{P} + \mathbf{t}) \quad (2-8)$$

用 x_1 和 x_2 分别表示 p_1 、 p_2 在归一化平面上的坐标，可得：

$$\mathbf{p}_1 = \mathbf{K} \mathbf{x}_1, \mathbf{p}_2 = \mathbf{K} \mathbf{x}_2 \quad (2-9)$$

带入 2-8 式，可得：

$$\mathbf{x}_2 = \mathbf{R} \mathbf{x}_1 + \mathbf{t} \quad (2-10)$$

两边同乘以 \mathbf{t}^\wedge 的反对称矩阵 \mathbf{t}^\wedge ，得：

$$\mathbf{t}^\wedge \mathbf{x}_2 = \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 \quad (2-11)$$

两边同乘以 \mathbf{x}_2^T ，得：

$$\mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{x}_2 = \mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 \quad (2-12)$$

上式左侧式中， $\mathbf{t}^\wedge \mathbf{x}_2$ 与 \mathbf{x}_2 垂直，内积相乘为零，因此可得：

$$\mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 = 0 \quad (2-13)$$

将 \mathbf{p}_1 和 \mathbf{p}_2 带入上式，有：

$$\mathbf{p}_2^T \underbrace{\mathbf{K}^{-T} \mathbf{t}^\wedge \mathbf{R} \mathbf{K}^{-1}}_{\mathbf{F}} \mathbf{p}_1 = 0 \quad (2-14)$$

式 2-13、2-14 都为对极约束，其几何意义是 $O_1 P_1 P_2$ 三点共面，对极约束中同时包含了平移和旋转，中间部分可以记为两个矩阵——基础矩阵 \mathbf{F} 和本质矩阵 \mathbf{E} ：

$$\mathbf{E} = \mathbf{t}^\wedge \mathbf{R}, \mathbf{F} = \mathbf{K}^{-T} \mathbf{t}^\wedge \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \quad (2-15)$$

(2) 单应矩阵

基本矩阵和本质矩阵描述了最一般的特征点几何关系，当场景中的特征点都位于同一平面时，可采用单应矩阵 \mathbf{H} 来估计相机运动。假设两个图像上有一对匹配好的特征点 p_1 和 p_2 ，这对特征点在三维空间中位于平面 \mathbf{P} 上，该平面的法向量为 \mathbf{n}^T ，则有如下关系式：

$$\mathbf{n}^T \mathbf{P} + d = 0 \Rightarrow -\frac{\mathbf{n}^T \mathbf{P}}{d} = 1 \quad (2-16)$$

由式 2-8 可得：

$$\begin{aligned} \mathbf{p}_2 &= \mathbf{K}(\mathbf{R}\mathbf{P} + \mathbf{t}) \\ &= \mathbf{K} \left(\mathbf{R}\mathbf{P} + \mathbf{t} \cdot \left(-\frac{\mathbf{n}^T \mathbf{P}}{d} \right) \right) \\ &= \mathbf{K} \left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d} \right) \mathbf{P} \\ &= \underbrace{\mathbf{K} \left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d} \right) \mathbf{K}^{-1}}_{\mathbf{H}} \mathbf{p}_1 \end{aligned} \quad (2-17)$$

可得到一个描述图像坐标 p_1 和 p_2 之间的变换，为：

$$\mathbf{p}_2 = \mathbf{H}\mathbf{p}_1 \quad (2-18)$$

其中， $\mathbf{H} = \mathbf{K} \left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d} \right) \mathbf{K}^{-1}$

对极约束给出了两个匹配点的空间位置关系，因此，相机位姿估计问题可以分为以下两步：

- (1) 根据配对点的像素位置计算出 \mathbf{E} 或者 \mathbf{F} 或者 \mathbf{H} 矩阵；
- (2) 若步骤 1 求出的是 \mathbf{E} 、 \mathbf{F} 矩阵，可通过八点法^[26]求出 \mathbf{R} ， \mathbf{t} ；若步骤 (1) 求出的是 \mathbf{H} 矩阵，可采用直接线性变换法求出 \mathbf{R} ， \mathbf{t} 。

需要说明，在求解相机运动时，如果特征点共面或者相机发生纯旋转运动，基础矩阵或本质矩阵的自由度会下降，求解出的相机位姿受噪声影响的程度变大，出现退化现象，此时就可以使用单应矩阵来估计相机的运动。在 ORB-SLAM2^[12]中，作者就同时估

计基础矩阵 F 和单应矩阵 H ，选择重投影误差较小的矩阵作为最终的运动估计矩阵，从而达到自适应求解最优相机运动的目的。

由上所述，相机的运动可分解为旋转运动和平移运动，其中平移运动可由一个向量表示，旋转运动可以通过旋转矩阵、旋转向量、欧拉角或者单位四元数等表示^[28]。旋转矩阵必须满足行列式为 1 且为正交矩阵的条件；欧拉角会出现万向锁问题，不适于插值和迭代；而单位四元数需要满足模为 1 的条件，使用这几种方式表达旋转运动会带来额外约束，增大优化的复杂度。因此，在后续表示相机的位姿时，采用李代数表达方式，也为后端优化做准备。

2.1.4 空间点重建

得到两帧图像相机的位姿后，就可以通过三角测量对图像中匹配特征点对的空间三维点坐标进行重建，从而建立局部地图。三角测量是通过两处观察同一个点的夹角，从而确定该点的距离，其原理如图 2-3 所示：

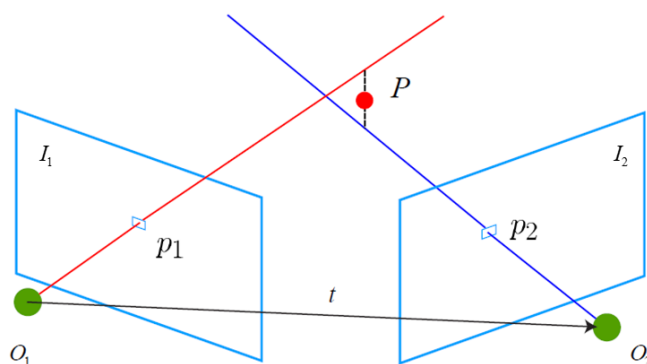


图 2-3 空间点重建示意图

考虑图像 I_1 和 I_2 ，以左图 I_1 为参考帧， I_2 相对于 I_1 的相机运动为 R, t ，相机光心为 O_1 和 O_2 。在 I_1 中有特征点 p_1 ，其在 I_2 中对应的特征点为 p_2 。理论上直线 O_1p_1 和 O_2p_2 在场景中会相交于一点 P ，该点即为两个特征点所对应的地图点在三维场景中的位置。然而由于测量数据有噪声，这两条直线通常无法相交，可以通过最小二乘法求解。

设第一帧的相机位姿为 $\mathbf{T}_1 = \begin{pmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{pmatrix}$ ，第二帧的相机位姿为 $\mathbf{T}_2 = \begin{pmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{pmatrix}$ ，据前所

述, $\mathbf{T}_2 = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \cdot \mathbf{T}_1$ 。以第一帧为例, 根据相机的投影模型, 可得:

$$s\mathbf{p}_1 = s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}_1\mathbf{P}_w + \mathbf{t}) \stackrel{\text{齐次转换}}{=} \mathbf{KT}_1\mathbf{P}_w = \mathbf{M}_1\mathbf{P}_w \quad (2-19)$$

设 $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ 为投影矩阵 \mathbf{M}_1 的行向量, $(u_1 \ v_1 \ 1)^T$ 为第一帧上二维图像特征点的齐次坐标, \mathbf{P}_w 为特征点 \mathbf{P} 在世界坐标系下的三维坐标, s 是常数, 因此有:

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix} \mathbf{P}_w \Rightarrow \begin{matrix} \mathbf{m}_3\mathbf{P}_w u_1 = \mathbf{m}_1\mathbf{P}_w \\ \mathbf{m}_3\mathbf{P}_w v_1 = \mathbf{m}_2\mathbf{P}_w \end{matrix} \Rightarrow \begin{pmatrix} \mathbf{m}_3 u_1 - \mathbf{m}_1 \\ \mathbf{m}_3 v_1 - \mathbf{m}_2 \end{pmatrix} \mathbf{P}_w = 0 \quad (2-20)$$

同理, 设第二帧的投影矩阵为 \mathbf{M}_2 , 结合两帧的矩阵方程, 对世界坐标下的所有三维点集 $\{\mathbf{P}_{wi}, i=1,2,\dots,n\}$, 有:

$$\begin{pmatrix} \mathbf{m}_{13}u_{1i} - \mathbf{m}_{11} \\ \mathbf{m}_{13}v_{1i} - \mathbf{m}_{12} \\ \mathbf{m}_{23}u_{2i} - \mathbf{m}_{21} \\ \mathbf{m}_{23}v_{2i} - \mathbf{m}_{22} \end{pmatrix} \mathbf{P}_{wi} = \mathbf{A}_i \mathbf{P}_{wi} = 0 \quad (2-21)$$

其中, \mathbf{m}_{ij} 是第 i 个投影矩阵的第 j 行。通过最小二乘法对矩阵 \mathbf{A}_i 进行 SVD 分解, 即可求得第 i 个点的三维坐标。

2.2 后端优化

2.2.1 图优化理论

前端的视觉里程计只能计算出相邻帧间的相机运动, 随着误差的累积, 相机的运动位姿会偏离真值越来越远, 因此需要对相机位姿进行修正。后端优化是一个最大后验估计问题^[7], 已知包含有噪声的相机的观测数据, 估计 SLAM 系统中相机的位姿以及环境地图中路标的坐标。

假设在 SLAM 中待估计的变量为 χ , 该变量集合包含机器人的轨迹 (即离散的位姿) 和环境的路标; 观测值为 $Z = \{z_k : k = 1, \dots, m\}$, 可表示为状态变量 χ 的函数:

$z_k = h_k(\chi_k) + \varepsilon_k$ ，其中 $\chi_k \subseteq \chi$ 是状态变量的子集， $h_k(\bullet)$ 是一个已知的函数，即为相机观测模型， ε_k 是一个随机测量噪声。

在最大后验估计问题中，需要通过计算变量 χ^* 使后验概率 $p(\chi|Z)$ 达到最大，通过贝叶斯定理，可得：

$$\chi^* \doteq \arg \max_{\chi} p(\chi|Z) = \arg \max_{\chi} p(Z|\chi)p(\chi) \quad (2-22)$$

其中， $p(Z|\chi)$ 是测量值 Z 的似然函数， $p(\chi)$ 是先验概率。先验概率包含所有关于 χ 的先验知识，如果不能获得先验知识， $p(\chi)$ 便是一个常数，可以从优化中剔除。在这种情况下，最大后验概率等同最大似然估计。注意，和卡尔曼滤波不同，最大后验概率估计不需要明确区分运动模型和观测模型，两种模型可以都视为因子，包含在估计过程中。此外，需要注意的是，卡尔曼滤波和最大后验概率估计在线性高斯情况下返回相同的估计值，但通常情况下，SLAM 的运动方程和观测方程是非线性函数，尤其是视觉 SLAM 中，相机模型需要使用相机内参模型和李代数表示的位姿，更不可能是一个线性系统。

假设测量值 Z 是独立的，也即噪声是不相关的，式 2-22 可以因式分解为：

$$\begin{aligned} \chi^* &= \arg \max_{\chi} p(\chi) \prod_{k=1}^m p(z_k | \chi) \\ &= \arg \max_{\chi} p(\chi) \prod_{k=1}^m p(z_k | \chi_k) \end{aligned} \quad (2-23)$$

其中， z_k 只依赖于状态变量 χ_k 。

式 2-23 可以用一个因子图表示^[29]，如图 2-4 所示，其中蓝色圆圈代表机器人位姿 (x_1, x_2, \dots) ，绿色圆圈代表路标位置 (l_1, l_2, \dots) ，棕色圆圈代表相机内参， p 代表先验知识。变量对应于因子图中的节点， $p(z_k | \chi_k)$ 和 $p(\chi)$ 称为因子，它们通过节点来约束概率值。因子图是一个图模型，将 k 个因子（也就是测量值 z_k ）和对应的变量 χ_k 联系在一起。因子图的第一个好处就是可以对问题进行可视化，从图中可以看出，问题的变量就是机器人的位姿、路标位置和相机参数，以及这些变量中施加约束的因素。因子图的第二个优点是通用性，可以对复杂的变量和因子进行建模，而且可以在任意节点之间建立联系。此外，因子图中因子之间的关联性会决定 SLAM 问题的稀疏性。

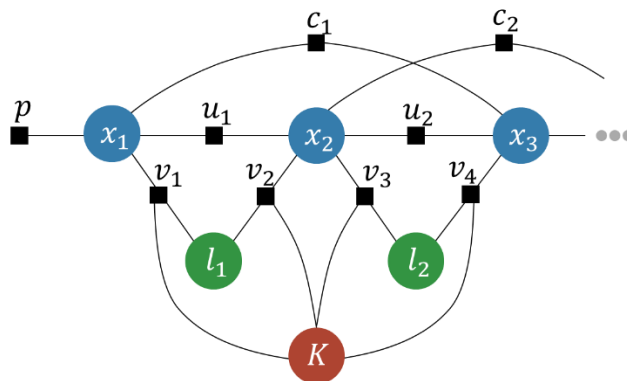


图 2-4 视觉 SLAM 的因子图模型

为了将式 2-23 表达的更为具体，假设对于观测方程 $z_k = h_k(\chi_k) + \varepsilon_k$ ，噪声项 $\varepsilon_k \sim N(0, \Omega_k)$ 。式 2-23 的最大似然函数可以表达为：

$$p(z_k | \chi_k) \propto \exp\left(-\frac{1}{2} \|h_k(\chi_k) - z_k\|_{\Omega_k}^2\right) \quad (2-24)$$

其中，使用表达式 $\|e\|_{\Omega}^2 = e^T \Omega e$ 。类似的，假设先验部分可以写为：

$$p(\chi) \propto \exp\left(-\frac{1}{2} \|h_0(\chi) - z_0\|_{\Omega_0}^2\right)$$

求最大后验概率相当于求上式的最小负对数，因此，最大后验概率可以表达为：

$$\begin{aligned} \chi^* &= \arg \min_{\chi} -\log \left(p(\chi) \prod_{k=0}^m p(z_k | \chi_k) \right) \\ &= \arg \min_{\chi} \sum_{k=0}^m \|h_k(\chi_k) - z_k\|_{\Omega_k}^2 \end{aligned} \quad (2-25)$$

该式是一个非线性最小二乘问题， $h_k(\bullet)$ 是非线性函数。注意，2-25 式是假设噪声服从正态分布的。噪声分布的其他假设会有不同的代价函数，例如，如果噪声服从拉普拉斯分布，式 2-25 中的平方二范数就会变成一范数。为了提高对异常值的适应能力，也可以用平稳的代价函数（例如 Huber 或者 Tukey）来代替平方二范数^[30]。

式 2-25 的求解一般通过泰勒展开连续线性化进行求解，例如 Gauss-Newton 或者 Levenberg-Marquardt 方法。连续线性化方法通过迭代方式，从一个给定的初始值 $\hat{\chi}$ 开始，用二次代价函数在 $\hat{\chi}$ 处做近似，可以通过求解一组线性方程以闭环形式做优化。记式 2-

25 中的误差函数为: $f(\chi_k) = h_k(\chi_k) - z_k$, 则具体的连续化方法过程为:

- (1) 给定一个初始值 $\hat{\chi}$;
- (2) 对于第 k 次迭代, 寻找一个增量 $\Delta\chi_k$, 使得 $\|f(\chi_k + \Delta\chi_k)\|^2$ 达到极小值;
- (3) 若 $\Delta\chi_k$ 足够小, 则停止迭代, 否则另 $\chi_{k+1} = \chi_k + \Delta\chi_k$, 继续迭代。

其中 $\Delta\chi_k$ 可以通过 Gauss-Newton 或者 Levenberg-Marquardt 方法进行求解。

经典 SLAM 求解过程中的关键点在于增量方程的系数矩阵的稀疏性, 而该系数矩阵的结构由因子图的结构决定。此外, 也可以设计增量求解器, 更新 χ 的估计值作为新的观测值。当前 SLAM 的库 (GTSAM^[31], g2o^[32], Ceres^[33], iSAM^[34], SLAM++^[35]) 都能在数秒以内求解成千上万个变量。事实证明, 基于非线性优化的最大后验概率估计方法在解决 SLAM 问题上, 要优于传统滤波器方法^[6]。

2.2.2 点特征的误差模型

对于图像上任意一个特征点 p_i 及其对应的空间点 P_i , 设相机的位姿为 \mathbf{R}, \mathbf{t} , 对应的李代数表示为 ξ , 特征点 p_i 的像素坐标为 $\mathbf{u}_i = (u_i \ v_i \ 1)^T$, 空间点 P_i 的坐标为 $\mathbf{P}_i = (X_i \ Y_i \ Z_i)^T$ 。根据相机的投影模型, 即为:

$$s_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \Rightarrow s_i \mathbf{u}_i = \mathbf{K} \exp(\xi^\wedge) \mathbf{P}_i \quad (2-26)$$

该式即为上一节所提的观测模型, 由于相机位姿未知, 并且存在观测噪声, 该式并不会准确成立, 存在误差:

$$\mathbf{e}_i = \mathbf{u}_i - \frac{1}{s_i} \mathbf{K} \exp(\xi^\wedge) \mathbf{P}_i \quad (2-27)$$

可以根据式 2-25, 对上述误差构建一个最小二乘问题:

$$\xi^* = \arg \min_{\xi} \sum_{i=1}^n \|\mathbf{e}_i\|^2 = \arg \min_{\xi} \sum_{i=1}^n \left\| \mathbf{u}_i - \frac{1}{s_i} \mathbf{K} \exp(\xi^\wedge) \mathbf{P}_i \right\|^2 \quad (2-28)$$

该误差是将特征的像素坐标与 3D 点按照当前估计的位姿进行投影得到的位置相比

较得到的误差，称之为重投影误差。几何意义如图 2-5 所示， p_1 与 p_2 是匹配的特征点对，并与空间点 P 对应，由于相机位姿不确定，在初始值中， P 在右图像上的投影 \hat{p}_2 与实际的观测值 p_2 存在一定的距离。最小二乘问题就是通过调整相机的位姿，使得该距离变小。

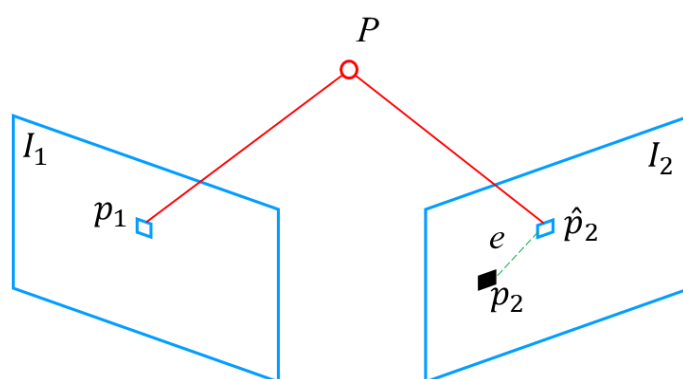


图 2-5 点特征的重投影误差示意图

2.3 闭环检测

如前文所述，前端对图像进行特征点的提取与匹配，提供相机轨迹、环境地图的初值，后端用于对所有数据进行优化。由于前端视觉里程计只考虑相邻帧的关联，因此前一时刻的计算误差会累积到下一个时刻，整个 SLAM 就会出现累积误差，虽然后端优化能够估计最大后验误差，但是只有相邻关键帧数据时，并不能完全消除累积误差。此时，闭环检测能够检测出相机经过同一个地方，可以为后端优化提供更多约束，使之得到一个全局一致的地图。

闭环检测本质上也是前端内容，是前端中的长期数据关联问题。在前端视觉里程计中，通常采用暴力匹配法来检测帧间的数据关联，当做闭环检测时，需要将当前帧与环境地图中的信息做匹配，暴力匹配法就显得不切实际，且基于匹配的方法对环境变化更为敏感。对于视觉 SLAM，目前最为常用的是词袋模型（Bag-of-words，简称 BoW）^[36]，该方法将特征空间进行量化，从而允许采用更有效的搜索算法来避免大量的计算。首先采用聚类算法（如 K-means）将从大量图像中提取到的特征点进行分类；然后采用层次聚类，将分类好的特征点构建成树结构^[37]，以便在大规模数据集中进行快速查找。词袋

模型在单回路闭合检测问题表现很可靠^[38]，但是当外界场景的光照条件发生剧烈变化时，词袋模型适用性变差，此时，需要采用其他方法进行检测，如通过图像序列匹配^[39]，或者使用空间和外观信息进行检测^[40]等。

通过闭环检测和后端位姿图优化，SLAM 系统已经能够获得优化后机器人（即相机）的准确位姿和特征点的三维空间坐标，即完成了地图的构建。

2.4 综合框架

本文提出的基于点线综合特征的单目视觉 SLAM 方法，是在单目 ORB-SLAM 的基础上进行扩展的，采用类似的主体框架结构，系统框架图如图 2-6 所示。系统整体分为 3 个并行的线程：跟踪线程、局部建图线程和闭环检测线程。

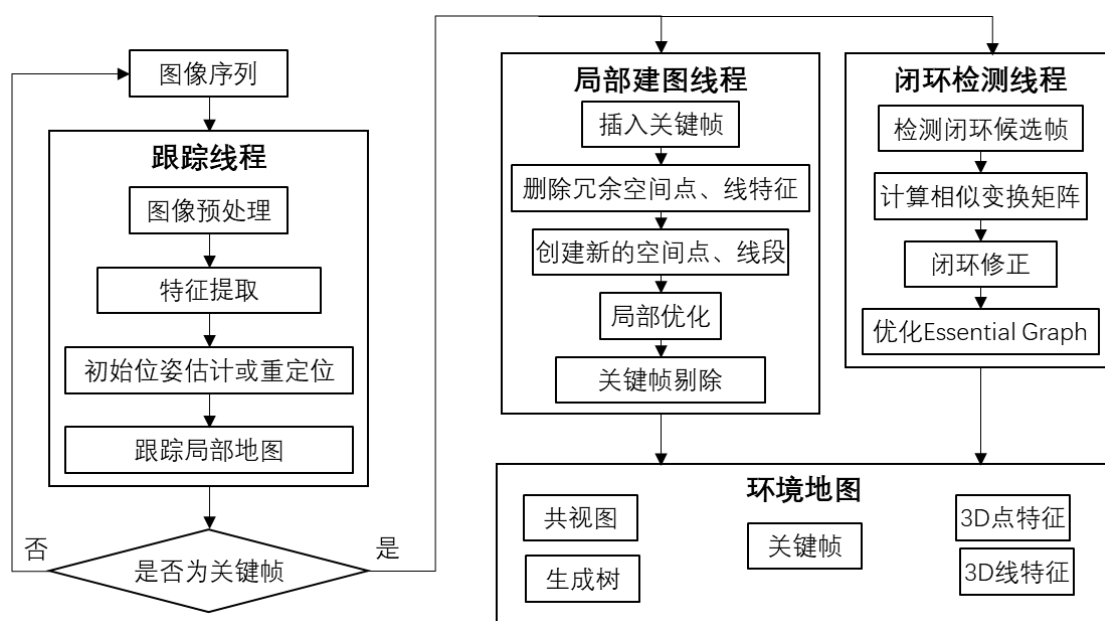


图 2-6 单目视觉 SLAM 框架图

(1) 跟踪线程：输入为单目相机的图像序列，每一个时刻的图像称为帧。该线程的作用是根据每一帧对相机进行定位，判断是否插入关键帧。首先对图像进行预处理，纠正图像畸变；然后分别提取点特征和线特征；根据特征之间的匹配关系，从上一帧估计相机的初始位姿，如果跟踪失败，就通过全局重定位对相机进行初始位姿估计；一旦相机有了初始位姿，就可以对点特征和线特征进行重投影找到局部地图的匹配点，根据两种特征的匹配关系对相机位姿进行优化；最后判断是否插入一个关键帧。

(2) 局部建图线程：该线程的作用是处理新的关键帧，优化局部地图中的点、线特征的三维坐标以及相机位姿。在关键帧中，对于还未匹配的特征点、特征线段，就在关键帧的共视图中搜索新的匹配来三维重建新的特征点和特征线。经过一段时间的重建，就会剔除错误的空间点和空间线，同时删除局部地图中含有冗余信息的关键帧。

(3) 闭环线程：当跟踪线程确定一个关键帧后，闭环检测会对该关键帧检测，确定是否存在闭环。如果检测到闭环，就计算一个相似变换来纠正闭环的累积漂移误差。然后，闭环两侧的关键帧进行配准，融合重复的特征点和特征线段。最后，基于位姿图优化，使环境地图全局一致。

由于在全部的环境地图中对线特征匹配的计算量太大，所以在闭环检测和计算相似变换时没有使用线特征。根据整体框架流程以及对前端视觉里程计、后端位姿优化以及闭环检测的分析，为了完成跟踪线程和局部建图线程中的工作，需要解决关于线特征的以下问题：

(1) 线特征的提取与匹配，快速从图像中提取丰富、均匀的线特征，确保线特征匹配的准确度；

(2) 线段特征参数表达，空间中的点特征有三个自由度，直线特征有四个自由度，表达方式的选择对后端优化至关重要；

(3) 线特征的三角化。空间点是空间三维物体的基元，线特征是更高层次的基元。理论上，可以先对每个点特征进行重建，再用曲线拟合出线特征，但这样计算时间长，需要对线特征进行整体重建；

(4) 线特征的后端优化。空间直线作为空间三维物体的另一种基元，需要定义其观测和误差模型，使其能在后端优化中和点特征同时进行优化。

2.5 本章小结

本章介绍了主流视觉 SLAM 的基本框架，以点特征为例阐述前端视觉里程计、后端优化、闭环检测的工作原理。最后，针对本文的目标，在 ORB-SLAM 的基础之上，设计了基于点线综合特征的 SLAM 的框架，结合对框架中不同线程的分析，引出了本文中对于线特征需要解决的问题。

3 视觉 SLAM 中的线特征

特征法视觉 SLAM 中，点特征和线特征是图像中两种不同的基元，点特征提取的是图像中角点，线特征提取的是图像中边缘。在一般情况下，点特征更为丰富，而线特征在后续建图时可视性更好，因此综合利用点、线特征，对于视觉 SLAM 既可以提高精度，又可以提高建图结果的直观性。对于点特征的提取、匹配、重建、误差模型的理论已较为成熟，由于线特征的复杂性，其在视觉 SLAM 的应用还有待研究。

3.1 线特征的提取与匹配

3.1.1 线特征的提取

本文的单目视觉 SLAM 中增加了线特征，则需要在前端识别图像中的线特征。线特征的识别有 Canny 算法、LSD 算法等，本文采用 LSD 算法。

Canny 算法^[41]是边缘检测算法的一种，它可以检测图像中的轮廓边缘特征，然后基于霍夫变换提取直线，再采用分割的方式计算直线的端点。Canny 算法实时性差，对不同场景的通用性差，为了识别到合理的线段特征经常需要修改参数，且在线段特征较多的局域会产生错误识别，所以本文线特征的识别没有选用 Canny 算法，而选择了 LSD 算法。

LSD (Line Segment Detector) 算法通过计算像素点间的梯度，将梯度相邻的像素点区域相连扩展成一个矩形，进而得到线段特征^[42]。LSD 作为一种亚像素级精度的局部线特征检测算法，能在任何灰度图上提取线段特征，同时能够控制算法的误检次数，平均而言，对每张图像可能会出现一次错误检测。

LSD 算法的具体流程如下：

(1) 预处理。对检测图像先进行降采样，再进行高斯滤波，防止图像出现模糊或者混叠，降低噪声的影响，避免阶梯效应造成的不利；

(2) 梯度计算。计算处理后图像的梯度的大小和方向，根据计算结果形成整个图像的梯度场；

(3) 构建线段支持域。采用区域增长算法，连接合并梯度场中梯度大小和方向接近的像素点，构建一个区域，也就是线段的支持域；

(4) 直线段检验。上述的线段支持域中含有一组像素点，这组像素点可形成可能的直线段，这需要计算检验，将线段支持域缩减到最小的外接矩形，计算缩减后矩形的主方向，若矩形的主方向与区域里某个像素点的梯度方向的角度差在给定的阈值内，则称此点为支持点。计算支持点的数量与该矩形区域内像素数量的比值，利用反向方法和 Helmholtz 准则判断该线段支持域是否是一条直线段。

LSD 算法的优点是其各个计算步骤的算法复杂度都与图像的像素数量成正比，计算快、计算时间是线性时间，且不需要额外调整参数。LSD 算法提取的线段具有方向性，这一特性在后续的线段匹配中有助于剔除错误配对，提高匹配速度。

下面针对同一个图像，分别采用 Canny+Hough 算法和 LSD 算法分别提取线段，提取结果见图 3-1 和图 3-2。由图 3-1 中可以看出，Canny+Hough 变换检测出来的直线分布不均匀，在场景边缘密集处检测的十分密集，且出现了大量误检情况；而由 LSD 算法基本能检测出场景中边缘存在的直线段，检测出的线段量丰富。



图 3-1 Canny+Hough 检测出的直线



图 3-2 LSD 检测出的直线

在使用 LSD 检测线段时，由于初始提取出来的线特征数量较多，在后续三维重建线特征时，计算量会很大。因此，为了减少计算量，并提高建图的可视性，需要对最初检测的线特征进行初始筛选，本文采用的初步筛选方法是根据线特征的长度与图像尺寸的占比进行筛选，图 3-3 是图 3-2 中的线特征经过筛选后的线特征。



图 3-3 LSD 筛选后的线特征

3.1.2 线特征的描述

在视觉 SLAM 中，线特征检测完毕后，需要对其进行帧间匹配。相对于点特征的匹配，线段特征由于存在断续、几何约束弱，在缺乏纹理的场景中辨识度不足等问题，其匹配难度更大。

为了解决这一问题，Zhiheng Wang 等提出将线段分割成多个子区域，然后计算各个区域内的梯度均值和标准差，从而建立 MSLD (mean-standard deviation line descriptor) 描述子^[43]，MSLD 描述子和 SIFT 描述子相似，具有旋转不变性和光照不变性的特点，且和线段的长度无关。Lilian Zhang 等对 MSLD 进一步优化，引入了全局和局部的高斯权重系数，称之为 LBD 描述子^[44]，与 MSLD 相比，LBD 的匹配效果更好，计算时间更短。

LSD 描述子先建立一个矩形的线段支持域 LSR，同时定义两个方向， d_L 和 d_\perp 。字母 m 代表矩形支持域的条带数目，字母 w 代表条带的像素宽度。为降低距离相对较远像素梯度的影响，采用全局高斯函数 f_g 作用于支持域内的各列；为降低条带之间的边界效应，采用局部高斯函数 f_l 作用于相邻的条带。

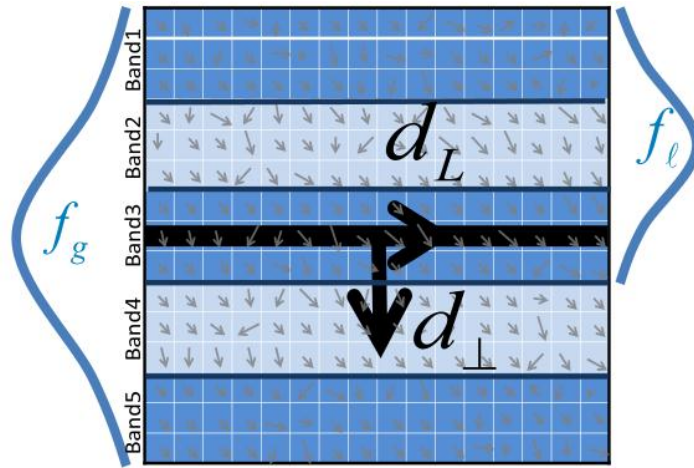


图 3-4 LSD 的矩形支持域

计算相邻条带间的梯度，则可以得到条带 B_j 相对应的特征向量 BD_j ，合并条带的特征向量得到 LBD 描述子：

$$LBD = (BD_1^T, BD_2^T, \dots, BD_m^T)^T \quad (3-1)$$

将 B_j 与相邻两个条带 B_{j-1} ， B_{j+1} 的局部梯度 $g' = (g'_{d_\perp}, g'_{d_L})$ 进行分别求和。其中，对于第 k 行，有：

$$\begin{aligned} v1_j^k &= \lambda \sum_{g'_{d_{\perp}} > 0} g'_{d_{\perp}} & v2_j^k &= \lambda \sum_{g'_{d_{\perp}} < 0} g'_{d_{\perp}} \\ v3_j^k &= \lambda \sum_{g'_{d_L} > 0} g'_{d_L} & v4_j^k &= \lambda \sum_{g'_{d_L} < 0} g'_{d_L} \end{aligned} \quad (3-2)$$

上式中 $\lambda = f_g(k)f_l(k)$ 为高斯系数，合并每行进行累加，构成特征向量 BD_j 对应的特征描述矩阵 BDM_j ：

$$\begin{aligned} BDM_j &= \begin{pmatrix} v1_j^1 & v1_j^2 & \dots & v1_j^n \\ v2_j^1 & v2_j^2 & \dots & v2_j^n \\ v3_j^1 & v3_j^2 & \dots & v3_j^n \\ v4_j^1 & v4_j^2 & \dots & v4_j^n \end{pmatrix} \in \mathbb{R}^{4 \times n} \\ n &= \begin{cases} 2w & j=1, m \\ 3w & \text{其它} \end{cases} \end{aligned} \quad (3-3)$$

根据特征描述矩阵 BDM_j 计算其均值方差 M_j^T 与均值向量 S_j^T ，合并均值方差和均值向量得到特征向量 $BD_j = (M_j^T, S_j^T)^T \in \mathbb{R}^8$ ，适当调整顺序得到 LBD 描述子：

$$LBD = (M_1^T, S_1^T, M_2^T, S_2^T, \dots, M_m^T, S_m^T) \in \mathbb{R}^{8m} \quad (3-4)$$

根据 Lilian Zhang 的研究结论，在 $m=9, w=7$ 时 LBD 描述效果最佳，特征向量此时是一个 72 维浮点型的向量。线特征匹配时需要计算描述子特征向量的距离，显然这样计算量太大，不符合视觉 SLAM 的实时性要求，借鉴 BRIEF 描述子对点特征的处理方式，将线特征的特征向量转化成二进制描述，形成二进制描述子，减小计算时间。LBD 的二进制描述子仅有 0 和 1 组成，采用 0 和 1 编码描述原 72 维浮点型的特征向量元素之间的关系。LBD 有了二进制的描述子后，采用汉明距离计算特征向量之间的距离，计算速度更快，计算效率更高。

3.1.3 线特征的匹配

根据前文所述，采用 LSD 算法提取线特征，并采用 LBD 描述子表征线特征，完成后需要对线特征路标进行关联，也就是线特征匹配。对于帧间线特征的匹配，本文采用暴力匹配中的 k-近邻算法，而由于 LBD 是二进制描述子，采用汉明距离计算描述子间的距离。

设图像 I_t 中提取到特征线段集 $\{x_l^m, m=1, 2, \dots, N\}$ ，在图像 I_{t+1} 中提取到的特征线段集

$\{x_{t+1}^m, m=1,2,...,N\}$ 。在对图像 I_t 和 I_{t+1} 之间进行线特征匹配时，计算 I_t 中每一个特征线段 x_t^m 与 I_{t+1} 中所有的 x_{t+1}^n 的描述子之间的汉明距离。考虑到 LSD 提取的是图像中的局部特征，会存在误匹配，因此选取与 x_t^m 距离最近的两个线特征，即最近距离对应的特征线段是 x_{t+1}^i ，次近距离对应的特征线段是 x_{t+1}^j ，计算线特征 x_t^m 与最近 x_{t+1}^i 的描述子距离和 x_t^m 与次近 x_{t+1}^j 的描述子距离之差，记为 Δx_t^m 。计算所有描述子距离差值的中位数绝对偏差 Δx_{mad} ，如果描述子最近距离与次近距离之差 Δx_t^m 大于中位数绝对偏差 Δx_{mad} ，则认为 I_{t+1} 中的线特征 x_{t+1}^j 是 I_t 中特征线段 x_t^m 的最优匹配。

其中，对于仅一个变量的数据集，可采用中位数绝对偏差 MAD 来度量样本的差异性，相对于标准差而言 MAD 的稳健性更好，避免异常偏差数据项对于数据集的影响。对于单个变量的数据集 $X = \{X_1, X_2, ..., X_n\}$ ，MAD 的计算如下：

$$MAD(X) = median(|X_i - median(X)|) \quad (3-5)$$

图 3-5、3-6 显示了对线特征的匹配结果，由图 3-5 中可以看出，未筛选的线特征匹配中存在大量的误匹配，经过上述方法的筛选过后（如图 3-6），匹配的线特征数量虽然降低，但提高了匹配的准确率。

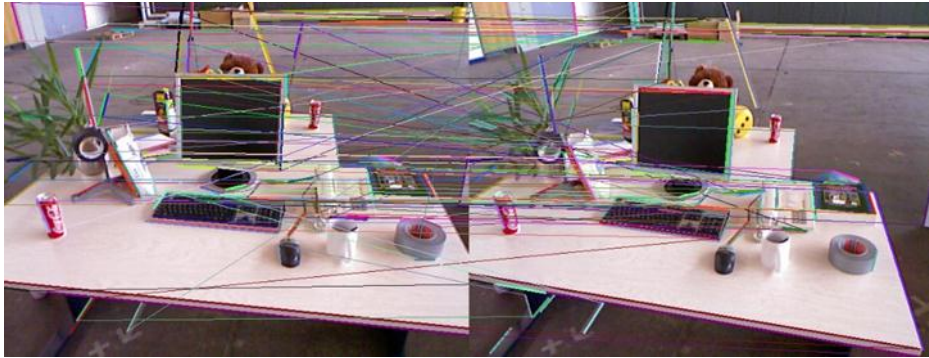


图 3-5 未筛选的匹配

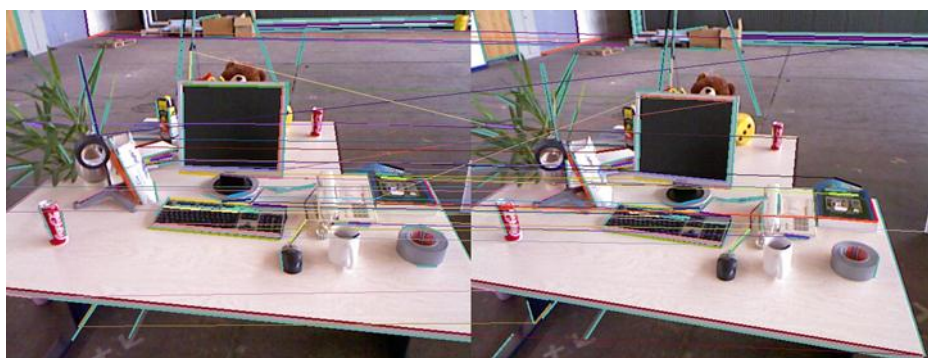


图 3-6 筛选后的匹配

3.2 线特征的表达方式

线特征的表达主要根据是否保留线特征的两个端点进行分类^[45]。若考虑直线特征，普吕克（Plücker）坐标是最常用的空间直线表达方法，它由两个三维向量 (\mathbf{u}, \mathbf{d}) 组成，向量 \mathbf{u} 是直线的方向或指向向量，如图 3-7 所示。向量 \mathbf{d} 由下式定义：

$$\mathbf{d} = \mathbf{u} \times \mathbf{m} \quad (3-6)$$

其中，直线上任何一点用 \mathbf{m} 表示。对于直线上不同的点 \mathbf{m} ，都对应同一个 \mathbf{d} ，因此，式 3-6 和空间直线是一一对应的；另外，如果对 \mathbf{u} 乘以一个不为零的标量 λ ，那么 \mathbf{d} 也相应的乘以 λ ，因此普吕克坐标是一组齐次坐标。此外，如果 \mathbf{u} 是一个单位向量，那么向量 \mathbf{d} 的模 $\|\mathbf{d}\|$ ，也就是原点到直线的距离。需要注意，普吕克坐标的六个参数是相互关联的，即 $\mathbf{u} \cdot \mathbf{d} = 0$ 。

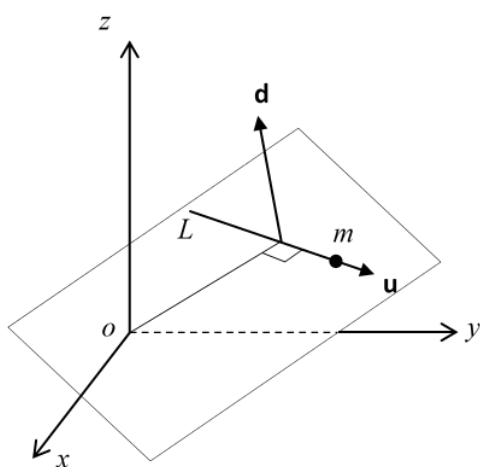


图 3-7 空间直线与普吕克坐标

普吕克坐标的优点是：用一个六维齐次坐标表示了三维射影空间中的任意一条直线，已经证明在计算机图形学中十分有用，且可以扩展用于机器人控制中的运动学理论。

而这种直接忽略掉直线段的端点，考虑无限长直线的处理方法的缺点：首先，直线段的信息量要比直线多，只考虑直线意味着我们忽视了所观测到的直线段在空间中的位置；其次，直线的不确定性并不能精确反映所观测到的直线段的不确定性。

考虑到直线段两个端点的定位作用，以及为了在后端优化中能够同时优化点特征和线段特征，本文采用端点表示法，即三维直线段由其两 endpoints 来表示。而直线段的端点的不稳定受三方面的影响：

(1) 算法的影响：LSD 在提取线段特征时，采用了降采样和高斯滤波来减少图像噪声和锯齿的影响，但是在尺寸缩小的图中检测出来的线段会损失一定的精度；

(2) 物理的影响：当图像场景中有物体被遮挡，则在不同的角度下的图像，线段的长度会发生变化；

(3) 光学的影响：图像中的直线在不同的光照条件下，物体表面成像也会随视角和强度的变化而变化，进而影响到场景中线段的成像。

本文为了进一步降低上述因素的影响，对线段匹配做了进一步的筛选，提高线段匹配的正确率，根据线段端点的重投影到线段在图像平面上所在直线的距离做筛选，设定的阈值为重投影点到直线的距离小于直线段长度的 5%。

3.3 线特征的三维重建

在对帧间的线特征做匹配后，需要根据匹配的线特征对计算线特征的三维空间坐标，即线特征的三维重建。本文采用端点法表示特征线段，但由于线特征的端点不稳定，匹配的线特征对的端点并不一定匹配，如果直接按照点特征的三角化方法重建三维线特征，误差会非常大。因此，需要根据线特征的极线约束重新建立线特征的模型。

理论上，可采用三维点的重建方法，将线特征的部分点重建处理，再根据直线拟合得到线特征，但这样整个过程需要大量时间，因此需要匹配较多特征点，并对这些特征点三维重建。因此，需要根据直线特征在图像坐标下的表达方程，整体重建线特征在三维坐标系下的表达。

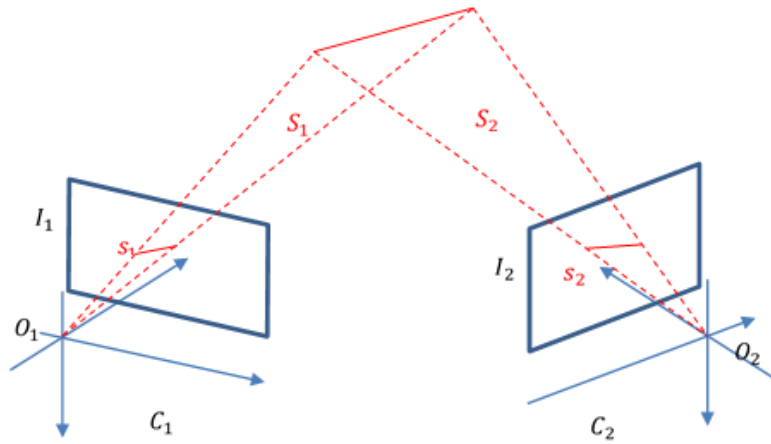


图 3-8 线特征的三维重建

如图 3-8 所示，对于空间中的一条直线 S ，在 C_1 与 C_2 两帧相机位置处，可以观察到 S 的两个图像 s_1 与 s_2 ，则 O_1 与 s_1 可以组成平面 S_1 ， O_2 与 s_2 可以组成平面 S_2 ，平面 S_1 与平面 S_2 的交线即为空间直线 S 。假设相机在 C_1 处的投影矩阵为 M_1 ，在 C_2 处的投影矩阵为 M_2 ， s_1 与 s_2 是空间直线 S 在两个相机平面上的二维投影直线，因此可以获取 s_1 与 s_2 在图像坐标系下的方程为：

$$\mathbf{s}_1^T \mathbf{u}_1 = 0 \quad (3-7)$$

$$\mathbf{s}_2^T \mathbf{u}_2 = 0 \quad (3-8)$$

其中， \mathbf{u}_1 和 \mathbf{u}_2 分别为直线 s_1 与 s_2 上点的齐次坐标， $\mathbf{s}_1 = (s_{11}, s_{12}, s_{13})^T$ 与 $\mathbf{s}_2 = (s_{21}, s_{22}, s_{23})^T$ 称为直线 s_1 与 s_2 的直线方程。相机在 C_1 处的投影方程为：

$$Z_{C1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}_1 | \mathbf{t}_1) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{M}_1 \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3-9)$$

其中， $\mathbf{M}_1 = \mathbf{K}(\mathbf{R}_1 | \mathbf{t}_1)$ ，空间点 P 坐标为 $\mathbf{P} = (X \ Y \ Z)^T$ ，将上式简记为：

$$Z_{C1} \mathbf{u}_1 = \mathbf{M}_1 \mathbf{P} \quad (3-10)$$

对上式两边同乘以 \mathbf{s}_1^T ，得到：

$$\mathbf{s}_1^T Z_{C1} \mathbf{u}_1 = \mathbf{s}_1^T \mathbf{M}_1 \mathbf{P} \quad (3-11)$$

华 中 科 技 大 学 硕 士 学 位 论 文

对于空间中的任意一个三维点，上式都成立，但若空间点 \mathbf{P} 位于 S_1 平面上，则直线 s_1 一定经过该空间点在图像平面上的投影点，因此式 3-11 的左侧等于零。根据式 3-11 可以得到下式：

$$\mathbf{s}_1^T \mathbf{M}_1 \mathbf{P} = 0 \quad (3-12)$$

上式为关于 \mathbf{P} 的线性方程，其成立的前提条件是 \mathbf{P} 点位于 S_1 平面上，也就是说式 3-12 是 S_1 平面的方程。

同理，可以得到 S_2 平面的方程为：

$$\mathbf{s}_2^T \mathbf{M}_2 \mathbf{P} = 0 \quad (3-13)$$

由于 S 是 S_1 平面和 S_2 平面的交线，所以式 3-12 和式 3-13 联立就是 S 在世界坐标系下的表达。从上式可以看出，若已知成像平面上直线 s_1 和 s_2 的直线方程，以及两帧相机处的投影矩阵 \mathbf{M}_1 和 \mathbf{M}_2 后，不需要分解相机的内外参数，就可以获取空间直线 S 在世界坐标系下的表达。

已知线特征所在的空间直线，现在需要求解其两端的端点坐标。以 I_1 为准进行求解，设线特征的起始点在 I_1 图像平面上的齐次坐标为 $(u_s, v_s, 1)^T$ ，起始点在世界坐标系下的坐标标记为 \mathbf{X}_s ， \mathbf{M}_{11} ， \mathbf{M}_{12} ， \mathbf{M}_{13} 是投影矩阵 \mathbf{M}_1 的行向量。根据相机的投影模型，可知：

$$s \begin{pmatrix} u_s \\ v_s \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{11} \\ \mathbf{M}_{12} \\ \mathbf{M}_{13} \end{pmatrix} \mathbf{X}_s \Rightarrow \begin{matrix} \mathbf{M}_{13} \mathbf{X}_s u_s = \mathbf{M}_{11} \mathbf{X}_s \\ \mathbf{M}_{13} \mathbf{X}_s v_s = \mathbf{M}_{12} \mathbf{X}_s \end{matrix} \Rightarrow \begin{pmatrix} \mathbf{M}_{13} u_s - \mathbf{M}_{11} \\ \mathbf{M}_{13} v_s - \mathbf{M}_{12} \end{pmatrix} \mathbf{X}_s = 0 \quad (3-14)$$

而起始点也在直线 S 上，同样满足式 3-12 和 3-13，因此对起始点有：

$$\begin{pmatrix} \mathbf{s}_1^T \mathbf{M}_1 \\ \mathbf{s}_2^T \mathbf{M}_2 \\ \mathbf{M}_{13} u_s - \mathbf{M}_{11} \\ \mathbf{M}_{13} v_s - \mathbf{M}_{12} \end{pmatrix} \mathbf{X}_s = \mathbf{A} \mathbf{X}_s = 0 \quad (3-15)$$

通过最小二乘法对矩阵 \mathbf{A} 进行 SVD 分解，即可求得起始点的三维坐标。

同理，记终止点在世界坐标系下的坐标为 \mathbf{X}_e ，其在第一帧 I_1 图像平面上的齐次坐标

为 $(u_e, v_e, 1)^T$ ，有：

$$\begin{pmatrix} \mathbf{s}_1^T \mathbf{M}_1 \\ \mathbf{s}_2^T \mathbf{M}_2 \\ \mathbf{M}_{13} u_e - \mathbf{M}_{11} \\ \mathbf{M}_{13} v_e - \mathbf{M}_{12} \end{pmatrix} \mathbf{X}_e = \mathbf{B} \mathbf{X}_e = 0 \quad (3-16)$$

通过最小二乘法对矩阵 \mathbf{B} 进行 SVD 分解，即可求得终止点的三维坐标。

3.4 线特征的误差模型

为了将 ORB-SLAM 扩展到使用线特征，需要合理定义基于线特征的重投影误差和线特征参数化^[46]。定义线特征的端点的三维坐标为 $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^3$ ，其在图像平面上对应的二维坐标为 $\mathbf{p}_d, \mathbf{q}_d \in \mathbb{R}^2$ ， $\mathbf{p}_d^h, \mathbf{q}_d^h \in \mathbb{R}^3$ 是二维坐标对应的齐次坐标。由此可以获得线特征在图像平面上所在直线的归一化系数：

$$\mathbf{I} = \frac{\mathbf{p}_d^h \times \mathbf{q}_d^h}{|\mathbf{p}_d^h \times \mathbf{q}_d^h|} \quad (3-17)$$

线特征的重投影误差 E_{line} 被定义为线段两个端点在图像平面上的重投影点到检测线段所在直线的距离之和。如图 3-9 所示， \mathbf{P}, \mathbf{Q} 为线特征的两个端点的三维坐标， \mathbf{p}, \mathbf{q} 为该线特征两个端点的重投影， l 是与该线段匹配的线段所在直线，则线特征的重投影误差为：

$$E_{\text{line}}(\mathbf{P}, \mathbf{Q}, \mathbf{I}, \theta, \mathbf{K}) = E_{pl}(\mathbf{P}, \mathbf{I}, \theta, \mathbf{K}) + E_{pl}(\mathbf{Q}, \mathbf{I}, \theta, \mathbf{K}) \quad (3-18)$$

其中，

$$E_{pl}(\mathbf{P}, \mathbf{I}, \theta, \mathbf{K}) = \mathbf{I}^T \pi(\mathbf{P}, \theta, \mathbf{K}) \quad (3-19)$$

\mathbf{I} 是检测到的线段所在直线的系数， $\pi(\mathbf{P}, \theta, \mathbf{K})$ 表示将线段的端点 \mathbf{P} 反投影至图像平面上， \mathbf{K} 为相机内参矩阵， $\theta = \{\mathbf{R}, \mathbf{t}\}$ 是相机的外参（即相机的位姿）。

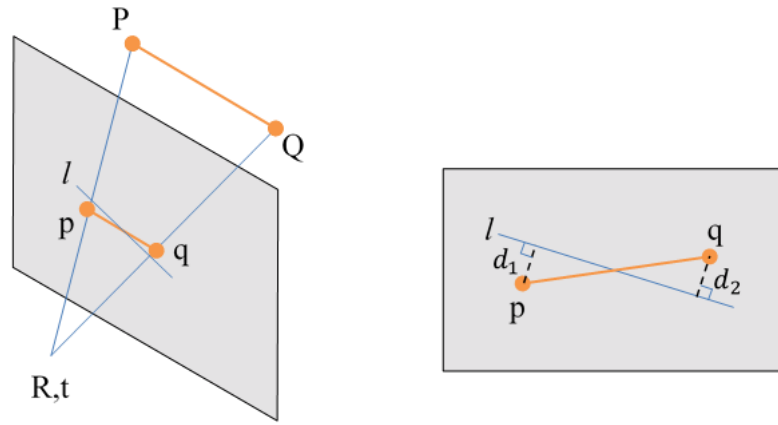


图 3-9 线特征的重投影误差

构建好线特征的重投影误差模型后，便可以通过高斯牛顿法和列文伯格-马夸尔特等优化算法对其进行求解，在此之前，需要对误差线性化，求解误差项关于优化变量的导数：

$$\mathbf{e}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{e}(\mathbf{x}) + \mathbf{J}\Delta\mathbf{x} \quad (3-20)$$

式 3-20 中的雅克比矩阵 $\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}}$ 指导了优化过程中迭代的方向，如果使用数值求导，会占用优化过程中较多的计算量并且精度低，而如果能够推导出解析形式，在迭代计算时就能快速计算出增量方程的系数，减小运算量。

对于用端点法表示的空间直线，虽然其误差模型与点特征的误差模型不同，但可以类比三维点的雅克比矩阵进行推导，下面重点阐述线段端点在线性化过程中的雅克比矩阵的推导过程。

对于 n 条三维空间直线段 L 及其投影 l ，需要求解相机的位姿 \mathbf{R} , \mathbf{t} ，将位姿对应的李代数表示为 ξ 。记某个空间直线段的一个端点坐标为 $\mathbf{S}_i = (X_i, Y_i, Z_i)^T$ ，其投影的像素坐标为 $\mathbf{u}_i = (u_i, v_i)^T$ 。根据相机的投影模型可知，像素位置与该端点的空间位置的关系为：

$$s_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \mathbf{K} \exp(\xi^\wedge) \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} \quad (3-21)$$

将其写成矩阵形式：

$$s_i \mathbf{u}_i = \mathbf{K} \exp(\hat{\xi}) \mathbf{S}_i \Rightarrow \mathbf{u}_i = \frac{1}{s_i} \mathbf{K} \exp(\hat{\xi}) \mathbf{S}_i \quad (3-22)$$

由公式 (3-17) 可以计算线段的观测方程系数, 记为 \mathbf{I} 。该端点的重投影误差为其重投影点到观测直线的距离: $e = \mathbf{u}_i \cdot \mathbf{I}$, 该误差是一个标量, 而相机位姿是一个 6 维向量, 因此, 线特征端点对应的雅克比矩阵是一个 1×6 的矩阵。

记线段一个端点变换到相机坐标系下的空间点坐标为 \mathbf{P}' , 其前三维为: $\mathbf{P}' = (\exp(\hat{\xi}) \mathbf{P})_{1:3} = (X', Y', Z')^T$ 。根据相机投影模型, 有: $s\mathbf{u} = \mathbf{K}\mathbf{P}'$, 可得到:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \quad (3-23)$$

$$\Rightarrow u = f_x \frac{X'}{Z'} + c_x, \quad v = f_y \frac{Y'}{Z'} + c_y \quad (3-24)$$

设线段观测方程 $\mathbf{I} = (l_1 \quad l_2 \quad l_3)^T$, 则 $e = \mathbf{u} \cdot \mathbf{I} = ul_1 + vl_2 + l_3$ 。

在用李代数表示相机位姿后, 可以将相机的位姿估计变成无约束的问题, 简化求解方式。而对相机位姿求导时, 可以采用李代数的扰动模型^[47], 以简化求导过程。现在, 对 $\hat{\xi}$ 左乘扰动变量 $\delta\hat{\xi}$, 然后考虑 e 的变化关于扰动量的导数, 利用链式法则, 可以列写如下:

$$\frac{\partial e}{\partial \delta\hat{\xi}} = \lim_{\delta\hat{\xi} \rightarrow 0} \frac{e(\delta\hat{\xi} \oplus \hat{\xi}) - e(\hat{\xi})}{\delta\hat{\xi}} = \frac{\partial e}{\partial \mathbf{P}'} \frac{\partial \mathbf{P}'}{\partial \delta\hat{\xi}} \quad (3-25)$$

其中, \oplus 表示李代数上的左乘扰动。第一项是误差关于投影点的导数:

$$\begin{aligned} \frac{\partial e}{\partial \mathbf{P}'} &= \left(\frac{\partial e}{\partial X'} \quad \frac{\partial e}{\partial Y'} \quad \frac{\partial e}{\partial Z'} \right) \\ &= \left(\frac{\partial u}{\partial X'} l_1 + \frac{\partial v}{\partial X'} l_2 \quad \frac{\partial u}{\partial Y'} l_1 + \frac{\partial v}{\partial Y'} l_2 \quad \frac{\partial u}{\partial Z'} l_1 + \frac{\partial v}{\partial Z'} l_2 \right) \end{aligned} \quad (3-26)$$

$$\text{其中, } \frac{\partial u}{\partial X'} = \frac{f_x}{Z'}, \frac{\partial v}{\partial X'} = 0, \frac{\partial u}{\partial Y'} = 0, \frac{\partial v}{\partial Y'} = \frac{f_y}{Z'}, \frac{\partial u}{\partial Z'} = -\frac{f_x X'}{Z'^2}, \frac{\partial v}{\partial Z'} = -\frac{f_y Y'}{Z'^2}$$

因此,

华中科技大学硕士学位论文

$$\frac{\partial e}{\partial \mathbf{P}'} = \begin{pmatrix} \frac{f_x}{Z'} l_1 & \frac{f_y}{Z'} l_2 & -\frac{f_x X'}{Z'^2} l_1 - \frac{f_x Y'}{Z'^2} l_2 \end{pmatrix} \quad (3-27)$$

第二项为变换后的点关于李代数的导数，采用李代数的扰动模型进行求导：

$$\begin{aligned} \frac{\partial(\mathbf{P}')}{\partial \delta \xi} &= \frac{\partial(\mathbf{T} \mathbf{p})}{\partial \delta \xi} \\ &= \lim_{\delta \xi \rightarrow 0} \frac{\exp(\delta \xi^\wedge) \exp(\xi^\wedge) \mathbf{p} - \exp(\xi^\wedge) \mathbf{p}}{\delta \xi} \\ &\approx \lim_{\delta \xi \rightarrow 0} \frac{(I + \delta \xi^\wedge) \exp(\xi^\wedge) \mathbf{p} - \exp(\xi^\wedge) \mathbf{p}}{\delta \xi} \\ &= \lim_{\delta \xi \rightarrow 0} \frac{\delta \xi^\wedge \exp(\xi^\wedge) \mathbf{p}}{\delta \xi} \\ &= \lim_{\delta \xi \rightarrow 0} \frac{\begin{pmatrix} \delta \phi^\wedge & \delta \rho \\ \mathbf{0}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} \mathbf{p} + \mathbf{t} \\ 1 \end{pmatrix}}{\delta \xi} \\ &= \lim_{\delta \xi \rightarrow 0} \frac{\begin{pmatrix} \delta \phi^\wedge (\mathbf{R} \mathbf{p} + \mathbf{t}) + \delta \rho \\ 0 \end{pmatrix}}{\delta \xi} \\ &= \begin{pmatrix} \mathbf{I} & -(\mathbf{R} \mathbf{p} + \mathbf{t})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix} = \begin{pmatrix} \mathbf{I} & -\mathbf{P}'^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix} \end{aligned} \quad (3-28)$$

$$\text{其中, } \mathbf{P}'^\wedge = \begin{pmatrix} 0 & -Z' & Y' \\ Z' & 0 & X' \\ -Y' & -X' & 0 \end{pmatrix}$$

而在 \mathbf{P}' 的定义中，我们取出了前 3 维，于是得到：

$$\frac{\partial \mathbf{P}'}{\partial \delta \xi} = \left(\mathbf{I} \quad -\mathbf{P}'^\wedge \right)_{3 \times 6} \quad (3-29)$$

将这两项相乘，就得到了 1*6 的雅克比矩阵：

$$\begin{aligned} &\frac{\partial e}{\partial \delta \xi} \\ &= \begin{pmatrix} \frac{f_x}{Z'} l_1 & \frac{f_y}{Z'} l_2 & -\frac{f_x X'}{Z'^2} l_1 - \frac{f_x Y'}{Z'^2} l_2 & -f_y l_2 - \frac{f_x l_1 X' Y'}{Z'^2} - \frac{f_y l_2 Y'^2}{Z'^2} & f_x l_1 + \frac{f_x l_1 X'^2}{Z'^2} + \frac{f_y l_2 X' Y'}{Z'^2} & -\frac{f_x l_1 Y'}{Z'} + \frac{f_y l_2 X'}{Z'} \end{pmatrix} \end{aligned} \quad (3-30)$$

该雅克比矩阵表述了线段端点重投影误差关于相机位姿李代数的一阶变化关系。另一方面，除了优化位姿，还希望优化线段特征的空间位置。因此，也需要求得误差关于空间线段端点的导数，仍然利用链式法则，有：

$$\frac{\partial e}{\partial \mathbf{P}} = \frac{\partial e}{\partial \mathbf{P}'} \frac{\partial \mathbf{P}'}{\partial \mathbf{P}} \quad (3-31)$$

第一项已经在前文进行推导，对于第二项，按照定义，有 $\mathbf{P}' = \exp(\xi^{\wedge})\mathbf{P} = \mathbf{R}\mathbf{P} + \mathbf{t}$ ， \mathbf{P}' 对 \mathbf{P} 求导后将只剩下 \mathbf{R} ，因此有：

$$\frac{\partial e}{\partial \mathbf{P}} = \begin{pmatrix} \frac{f_x}{Z'} l_1 & \frac{f_y}{Z'} l_2 & -\frac{f_x X'}{Z'^2} l_1 - \frac{f_y Y'}{Z'^2} l_2 \end{pmatrix} \mathbf{R} \quad (3-32)$$

如上所述，即可推导出观测相机方程关于相机位姿和线特征的两个导数矩阵，这两个矩阵能够在优化过程中提供重要的梯度方向，指导优化的迭代。

3.5 点线特征的综合误差模型

在图优化为框架的视觉 SLAM 中，BA（Bundle Adjustment）是核心算法，该算法是指从视觉重建中提炼出最优的 3D 模型和相机参数（内参数和外参数）^[48]。如第二章所述，在视觉 SLAM 中，图优化中模型的顶点是相机的位姿和路标点，对于点特征，顶点（相机位姿和地图点）之间的边是三维点的重投影误差；对于线特征，顶点（相机位姿和地图线）之间的边是线特征端点的重投影点到对应二维线段的距离。

用 \mathbf{T}_{cw} 表示图模型中相机的位姿，表示相机在世界坐标系的位姿，相机的位姿参数 $\theta = \{\mathbf{R}, \mathbf{t}\}$ 用李代数表示，用 BA 进行优化。

用 $\mathbf{X}_j \in \mathbb{R}^3$ 表示环境地图中的第 j 个特征点，对于第 i 个关键帧，该点可以通过相机投影模型投影到相机成像平面上：

$$\tilde{\mathbf{x}}_{i,j} = \pi(\mathbf{X}_j, \theta_i, \mathbf{K}) \quad (3-33)$$

其中， $\theta_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ 表示第 i 个关键帧的位姿，该点的观测值为 $\mathbf{x}_{i,j}$ ，因此可以定义点特征的重投影误差为：

$$\mathbf{e}_{i,j} = \mathbf{x}_{i,j} - \tilde{\mathbf{x}}_{i,j} \quad (3-34)$$

类似的，我们用 \mathbf{P}_j 和 \mathbf{Q}_j 表示环境地图中第 j 个线段的两个端点，其在相同关键帧

上的图像平面上的坐标为：

$$\tilde{\mathbf{p}}_{i,j}^h = \pi(\mathbf{P}_j, \theta_i, \mathbf{K}) \quad (3-35)$$

$$\tilde{\mathbf{q}}_{i,j}^h = \pi(\mathbf{Q}_j, \theta_i, \mathbf{K}) \quad (3-36)$$

已知该线段的两个端点的观测值为 $\mathbf{p}_{i,j}$ 和 $\mathbf{q}_{i,j}$ ，使用方程(3-17)估计观测线段的系数为 $\tilde{\mathbf{I}}_{i,j}$ 。可以定义该线特征的误差为：

$$\mathbf{e}'_{i,j} = (\tilde{\mathbf{I}}_{i,j})^T (\mathbf{K}^{-1} \mathbf{p}_{i,j}^h) \quad (3-37)$$

$$\mathbf{e}''_{i,j} = (\tilde{\mathbf{I}}_{i,j})^T (\mathbf{K}^{-1} \mathbf{q}_{i,j}^h) \quad (3-38)$$

误差(3-37)和(3-38)实际上是点到线的误差。如论文^[46]所述，由于端点 \mathbf{P}_j 和 \mathbf{Q}_j 沿着对应的3D直线偏移，这两个误差不是常数，作为隐式正则化，它们允许在BA中使用非最小线性化参数。

在本文中，假设观测噪声服从高斯分布，并将点特征和线特征的观测误差分布引入到误差模型中，从而提高噪声的鲁棒性以及位姿估计的精度。由2.2节后端优化中介绍可知，视觉SLAM可以抽象为观测方程：

$$z_{k,j} = h(y_j, x_k) + v_{k,j} \quad (3-39)$$

表示在位姿 x_k 处对路标 y_j 进行的一次观测，假设观测噪声 $v_{k,j}$ 满足零均值的高斯分布 $v_k \sim N(0, Q_{k,j})$ 。本文中使用了点、线两种特征，假设点特征的观测噪声和线特征的观测噪声均服从零均值的高斯分布（点特征 $v_{kp} \sim N(0, Q1_{k,j})$ ，线特征 $v_{kl} \sim N(0, Q2_{k,j})$ ），但是由于线特征的两个端点不如点特征稳定，因此线特征的观测噪声的标准差 $Q2_{k,j}$ 大于点特征的观测噪声的标准差 $Q1_{k,j}$ 。为了提高SLAM系统对噪声的鲁棒性和位姿估计的准确性，本文根据两种特征的观测噪声对后端优化中的误差方程做加权计算。

因此，针对点、线特征的误差项，可以建立一致性的误差方程：

$$C = \sum_{i,j} \rho(\mathbf{e}_{i,j}^T \boldsymbol{\Omega}_{i,j}^{-1} \mathbf{e}_{i,j} + \mathbf{e}_{i,j}'^T \boldsymbol{\Omega}_{i,j}'^{-1} \mathbf{e}_{i,j}' + \mathbf{e}_{i,j}''^T \boldsymbol{\Omega}_{i,j}''^{-1} \mathbf{e}_{i,j}'') \quad (3-40)$$

其中， ρ 是Huber核函数， $\boldsymbol{\Omega}_{i,j}^{-1}$ ， $\boldsymbol{\Omega}_{i,j}'^{-1}$ 分别是特征点和特征线段的观测协方差矩阵

的逆矩阵，它们决定了这些特征的尺度，并提高了位姿估计的精度，减弱视觉 SLAM 对噪声的敏感性。为了降低误差对匹配结果的影响，防止 BA 往错误值的方向优化，引入 Huber 核函数，降低代价函数中的异常值的影响。

3.6 本章小结

本章首先描述了线特征的提取、描述和匹配方法；接着描述了线特征的表达方式，说明选择端点表示法的原因；然后给出了空间中线特征的三维重建过程，建立线特征的误差模型，并推导了线特征关于优化变量的雅克比矩阵；最后描述了点线特征的综合误差模型，为第四章的点线综合 SLAM 做准备。

4 点线综合特征的单目 SLAM

视觉 SLAM 中的点特征在光线明暗、角度变化下不稳定，且部分低纹理、缺少变化的场景中难以提取到足够的点，同时三维点重建的稀疏地图不利于人机交互，线特征能够改善地图的直观性，见图 4-1。因此，本文结合前述章节的内容，建立基于点线综合特征的单目视觉 SLAM，本章将对综合框架中的关键部分进行重点论述。

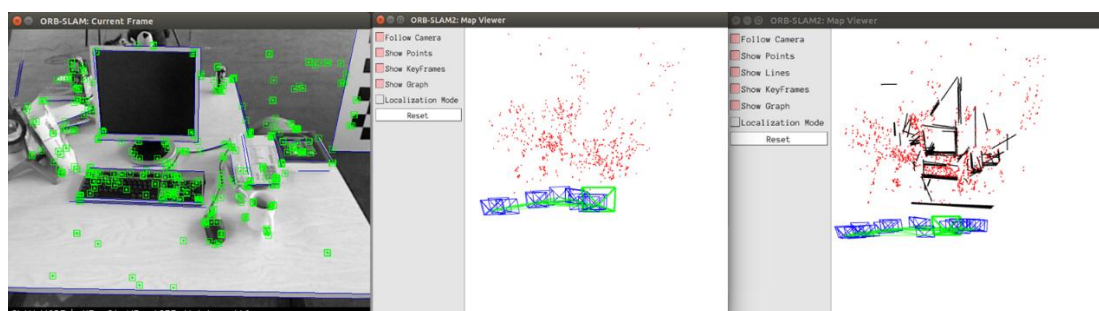


图 4-1 环境地图示意图

根据 2.4 节所述，整个点线综合特征的单目 SLAM 包括三个线程：跟踪线程、局部建图线程和闭环检测线程，由于在全部的环境地图中对线特征匹配的计算量太大，所以在闭环检测和计算相似变换时没有使用线特征。根据上一章线特征的理论介绍，本章内容主要说明在原来单目 ORB-SLAM 的基础上，对跟踪线程和局部建图线程所做的工作。

4.1 跟踪线程

跟踪线程通过相邻时刻图像的信息，粗略估计出相机的运动，完成视觉 SLAM 的前端工作，给后端优化提供初始值，下面分别介绍在原始单目 ORB-SLAM 的基础上，为了结合线特征而所做的工作。

4.1.1 图像预处理和特征提取

为了获得更好的成像效果，相机前方安装了透镜，而透镜自身的形状会对影响光线传播，且透镜在安装过程中会与成像平面存在一定的夹角，因此相机获取的图像存在畸变。为了减少畸变对后续工作的影响，需要对图像进行畸变校正。畸变校正处理一般有两种，一是先对整张图像去畸变，得到去畸变后的图像，然后对去畸变的图像进行特征提取；二是先提取图像的特征，然后按照去畸变方程，获取去畸变后的特征。因为本文采用了点特征和线特征，线特征单独去畸变过程复杂，所以采用第一种处理方式。

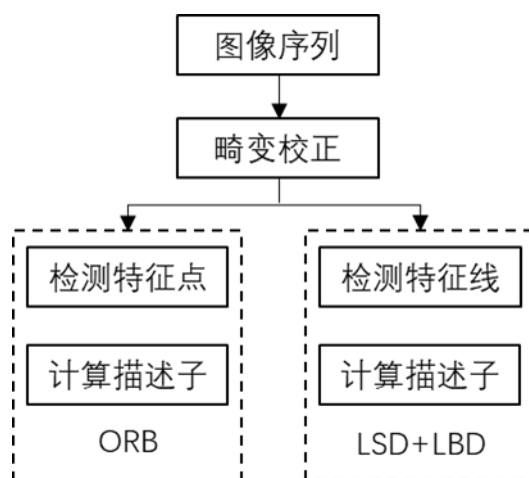


图 4-2 图像处理与特征提取

由于线特征的提取时间大于点特征的提取时间，为了同步两种特征，在特征提取阶段采用并行线程对图像分别提取点特征和线特征，点特征采用 ORB 提取算法，线特征采用 LSD 进行检测，两种特征的描述子均为二进制描述子，便于后续特征匹配时能够快速计算描述子距离，下图为同时提取点特征和线特征的示意图。

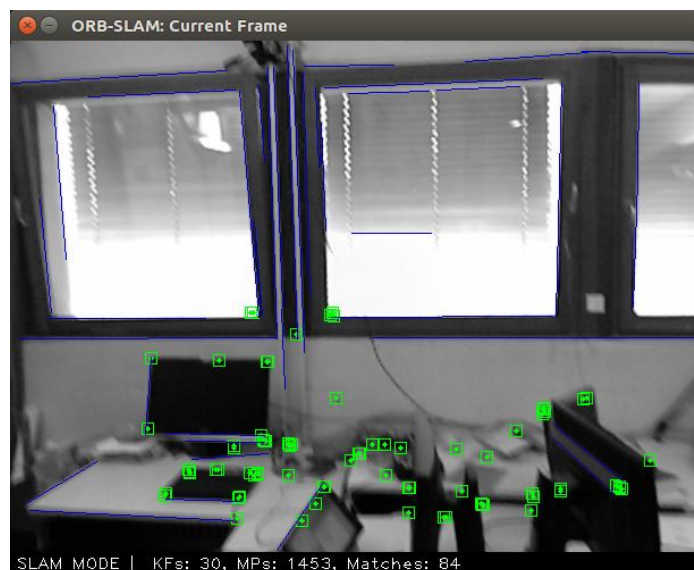


图 4-3 同时提取点、线特征示意图

4.1.2 环境地图初始化

视觉 SLAM 在正式跟踪前，需要对环境地图初始化，然后跟踪线程再根据初始地图进行跟踪和位姿估计。由于单目没有尺度，不能直接从一帧图像获取特征的三维坐标，因此初始化过程对单目视觉 SLAM 至关重要。

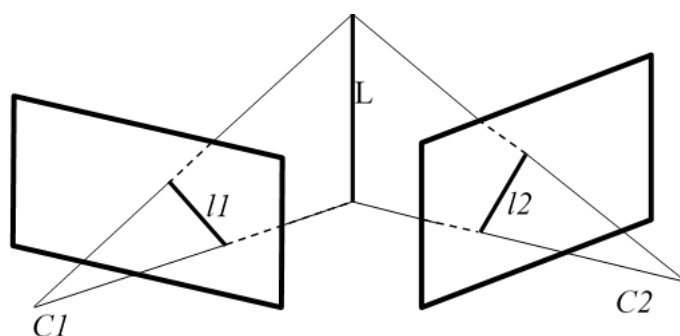


图 4-4 线特征投影模型

如图 4-4 所示为线特征的投影模型，假设有一条空间直线 L ，其与相机光心所成的平面称为该直线的投影平面，投影平面与相机平面的交线即为该直线在相机成像平面上的投影直线，而在投影平面上的任意三维直线的投影，对应的都是同一条二维投影直线。两条对应的二维投影直线所在的两个空间投影平面的交线，即为两条投影直线对应的三维直线结构。现在任意移动第二个相机（两个相机光心不能重合），就可以得到另外一个投影平面。该投影平面与第一个相机的投影平面相交，就可以得到另外一个三维直线结构，但该三维直线结构在两帧图像上的投影直线仍和之前的投影直线相同，因此两个相机的相对位置，不能通过两帧图像上的直线特征进行约束。

由于单目 SLAM 的初始环境地图是后续跟踪的基础，而线特征的两个端点又需要通过图像的位姿计算，因此本文的环境地图初始化过程仍然采用 ORB-SLAM 的初始化原理，初始化流程如图 4-5 所示。

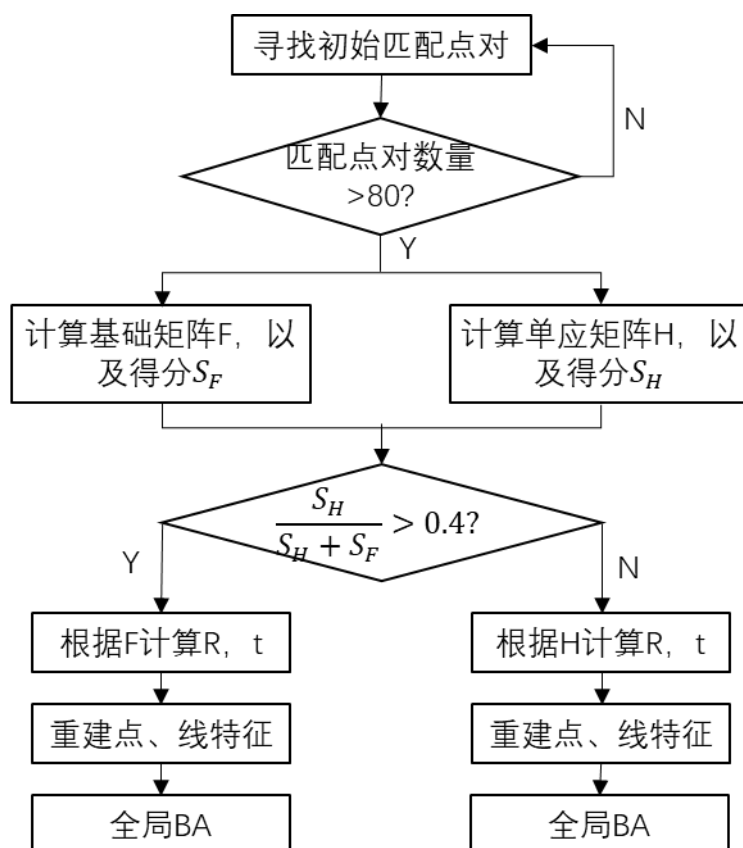


图 4-5 初始化流程

首先寻找两个点特征对和线特征对的匹配数量都比较多的帧，如果匹配数量较少（本文设定的阈值为 80），就重新寻找匹配对；然后同时计算单应性矩阵和基础矩阵，以及两个模型对应的得分 S_H 和 S_F ；通过两个模型的得分占比选择适合的模型，分解得到前两帧相机之间的位姿变化；通过计算得到的位姿对点特征和线特征进行三维重建，最后执行一个全局 BA 优化得到初始的环境地图和最初的两帧相机的位姿。全局 BA 优化的内容在 4.3 章节中详细介绍。

环境地图初始化成功后，就可以通过跟踪上一帧确定下一帧的初始位姿，主要方法是恒速运动模型和跟踪参考帧模型^[12]，恒速运动模型认为帧间运动为匀速运动，跟踪参考帧模型认为当前帧位姿和与其最近的关键帧位姿相同。

4.1.3 跟踪局部地图

一旦有了相机的初始位姿，并且有了一组初始的特征匹配，就可以将环境中的地图特征重投影到图像平面上，从而搜索更多的匹配对。为了限制在大型地图下的复杂性，

只投影局部地图，跟踪局部地图的流程如图 4-6 所示。首先更新局部地图，包括局部关键帧、局部地图点和局部地图线；然后分别在局部地图中查找与当前帧匹配的地图点和地图线；之后根据两种特征对相机位姿进行优化，但不优化两类特征；最后根据位姿优化的结果更新地图点和地图线的观测数据，并返回跟踪状态。其中位姿优化内容在 4.3 章节中详细阐述。

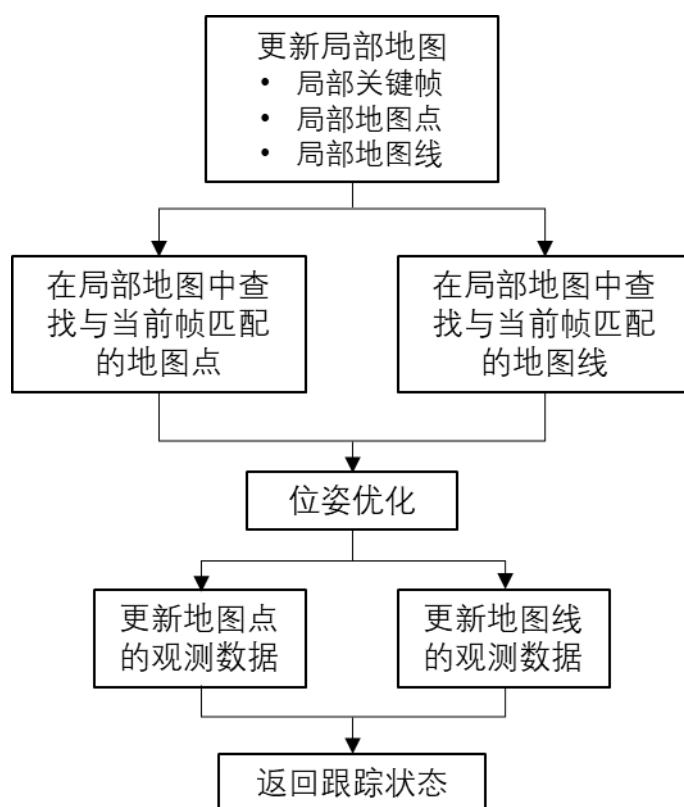


图 4-6 跟踪局部地图流程图

由于本文中对点特征的局部地图跟踪未做修改，因此只介绍对于线特征的局部地图跟踪过程，该过程的关键部分在于，在局部地图中查找与当前帧匹配的地图线。该部分的原理是将局部地图中的线特征重投影到图像平面上，根据重投影的线特征在图像平面上搜索相匹配的线特征。搜索匹配条件：

- (1) 重投影线段的中点与待匹配线段的重点距离小于搜索范围 r ;
- (2) 重投影线段的斜率与待匹配线段的斜率之差小于 $0.01*r$;
- (3) 待匹配线段的金字塔层数应该在预估金字塔层数范围内;
- (4) 计算重投影线段的描述子与待匹配线段的描述子的距离，当最近距离小于次

近距离的 60%，则认为匹配成功。

4.1.4 关键帧的选取

最初提出的单目 SLAM 方法是通过滤波实现的，该类方法对每一帧都会进行处理，从而估计地图特征点和相机位姿。但是该方法在处理连续帧时会浪费大量的计算时间，然而更新的信息量很少，会累积线性误差。另一方面，基于关键帧的方法，只根据选择的关键帧对地图进行估计，允许执行更精确的后端优化，即使后端优化的计算时间也很大，但是建图工作不会和图像序列的帧率相关。而文献^[6]中也论述了，在同样的计算量下，基于关键帧的方法要比基于滤波的方法更为精确。

因此，视觉 SLAM 的精度和关键帧的选取尤为相关，当关键帧加入过于频繁时，许多信息会重复引入系统中，加大计算量；当关键帧选取过于稀疏时，帧间匹配难度增大，可能会使系统跟踪失败。

根据 ORB-SLAM 中原本的关键帧选择策略，本文增加了关于线特征的判断依据来选择是否插入一个关键帧。由于在局部建图线程中有剔除关键帧的机制，所以为了使得跟踪线程受相机运动的影响更小，会尽快插入关键帧。当满足以下条件时，就可以插入新的关键帧：

- (1) 距离上次全局重定位后已经过去了至少 20 帧；
- (2) 局部建图处于空闲状态，或者距离上一次插入关键帧已经过去了至少 20 帧；
- (3) 当前帧至少有 50 个特征点和 10 条特征线；
- (4) 当前帧对参考帧的跟踪点少于 90%。

4.2 局部建图线程

若跟踪线程判断当前帧可以作为一个关键帧，就在局部地图中加入当前帧，并跟新局部地图，将跟踪线程提供的相机位姿、特征点和特征线段的三维坐标作为初始值，通过局部 BA 进行优化。局部建图线程的过程如图 2-6 所示，主要包括地图管理和局部地图优化两个部分，局部地图优化内容在 4.4 章节中详细介绍。

地图管理包括删除冗余的空间点线特征，创建新的空间点线以及剔除冗余的关键帧，由于本文并未修改点特征的增删步骤和剔除关键帧的内容，因此只说明线特征的地图管

理内容。

(1) 空间线特征的剔除

如 3.1.3 节所述, 尽管在对线特征匹配时, 经过初步筛选会去除大多数错误的线特征匹配, 但是错误匹配仍然存在, 因此在三维重建线特征时, 会导致错误的结果。此外, 由于线特征在图像中容易存在遮挡现象, 可能只能被某些关键帧检测到。因此, 需要对新添加的线特征进行筛选, 提高环境地图中线特征的正确率, 筛选条件为:

a. 跟踪到该地图线特征的帧数在预计可以观测到该地图线特征的帧数中占比大于 25%;

b. 每个地图线特征要至少能在三个关键帧中观测到。

(2) 创建新的空间线特征

对于当前关键帧中未匹配的线特征, 在与其共视的其他关键帧中进行搜索匹配, 此处的匹配算法采用暴力匹配法中的 K 近邻算法。通过这些匹配对创建新的三维线特征, 线特征的三维重建方法采用 3.3 节所述的方法。待线特征重建好后, 检测线特征的两个端点的深度在相机坐标系下是否为正值。

为了合理利用平台的计算资源, 提高运行效率, 点、线特征的删除和创建均采用并行线程, 分别同时处理点、线特征。

4.3 后端优化

在跟踪和局部建图环节, 后端优化主要采用了 3 种优化类型: 全局优化、位姿优化和局部优化。其中全局优化用于环境地图初始化, 位姿优化用于跟踪线程, 局部优化用于局部建图环节。这三种优化使用场景不同, 其待优化变量也不同, 所有的优化过程均在 g2o 框架中建立图模型, 并利用第三章的误差模型和雅可比矩阵。

在使用 g2o 框架做不同优化模型时, 需要做以下工作:

(1) 定义顶点和边的类型。由于本文中的线特征使用端点表示法, 所以线特征的顶点仍然可以采用 g2o 中对于三维坐标的定义, 无需重复定义; 边的类型需要根据使用环境做不同的定义, 具体需要定义以下函数:

a. 误差计算函数: `computeError()`。该函数需要取出误差边所连接的顶点的当前估

计值，根据相机投影模型和 3.4 节中对线特征误差的定义，计算线特征端点的重投影点到观测直线的距离。

b. 边对顶点的解析求导函数：`linearizeOplus()`。该函数中定义了雅克比矩阵的解析形式，用于优化过程中的梯度计算，雅克比矩阵的形式参考 3.4 节。

c. 存盘和读盘函数：`read`，`write`。用于读写观测值和信息矩阵。

(2) 构建图模型。根据三种优化的变量，分别构建图模型。

(3) 选择优化算法。本文采用的是 Levenberg-Marquadt 算法，该方法属于信赖区域法，能够避免线性方程组的系数矩阵的非奇异和病态问题，提供的增量更为鲁棒。

4.3.1 全局优化

全局 BA 优化是优化所有的关键帧和环境地图点、线特征，在环境地图初始化结束后，执行全局 BA 能够得到一个全局一致的地图，为后续跟踪线程做准备。

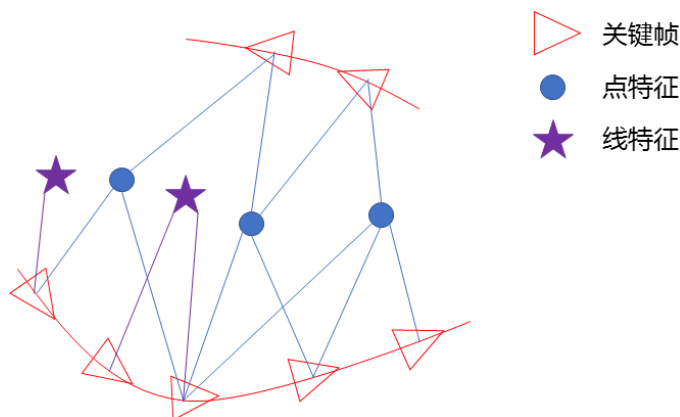


图 4-7 全局 BA 示意图

在 g2o 框架中，图优化的顶点为：

(1) 所有的关键帧位姿，类型为：`g2o::VertexSE3Expmap`；

(2) 所有的点特征和线特征的端点，类型为：`g2o::VertexSBAPointXYZ`；

误差边有：

(1) 点特征和关键帧之间的误差边，继承自二元边，类型为：`g2o::EdgeSE3Project`；

(2) 线特征和关键帧之间的误差边，该误差边需要自定义，继承自二元边 `BaseBinaryEdge`，定义的类型为：`EdgeLineProjectXYZ`。雅克比矩阵包括误差对于相机位姿的导数，以及误差对线特征的端点的导数。

4.3.2 位姿优化

位姿优化是只优化当前帧的位姿，而地图点和地图线保持不变。在视觉 SLAM 中，由于路标数量远大于位姿数量，一个关键帧往往关联了数百个特征点和特征线，实时全局 BA 的计算量会很大，因此可以构建一个只优化轨迹的图模型，而位姿节点之间的边，由两个关键帧之间通过特征匹配之后得到的运动估计来给定初始值，但是路标点的位置不再进行优化。由此，可以省去大量的特征点和特征线优化的计算，只保留了关键帧的轨迹。

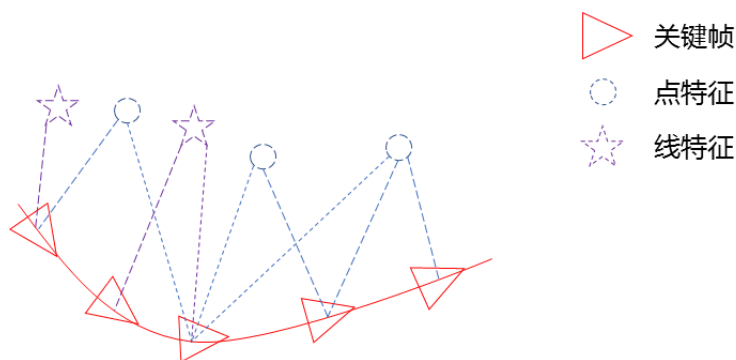


图 4-8 位姿优化示意图

在 g2o 框架中，图优化的顶点为：

- (1) 当前关键帧的位姿，类型为： `g2o::VertexSE3Expmap`；

误差边有：

- (1) 点特征和关键帧之间的误差边，继承自一元边 `BaseUnaryEdge`，类型为：`g2o::EdgeSE3ProjectXYZOnlyPose`；
- (2) 线特征和关键帧之间的误差边，该误差边需要自定义，继承自一元边 `BaseUnaryEdge`，定义的类型为：`EdgeLineProjectXYZOnlyPose`。雅克比矩阵只包括误差对于相机位姿的导数。

4.3.3 局部优化

局部 BA 优化用于局部建图线程，图优化结构如图 4-9 所示。在 g2o 中，待优化的顶点包括当前关键帧、在共视图中与当前关键帧相连的关键帧，以及这些关键帧观测到的地图点和地图线，所有其他观测到这些地图特征的关键帧也包含在图优化中，但在优化过程中保持固定不变。

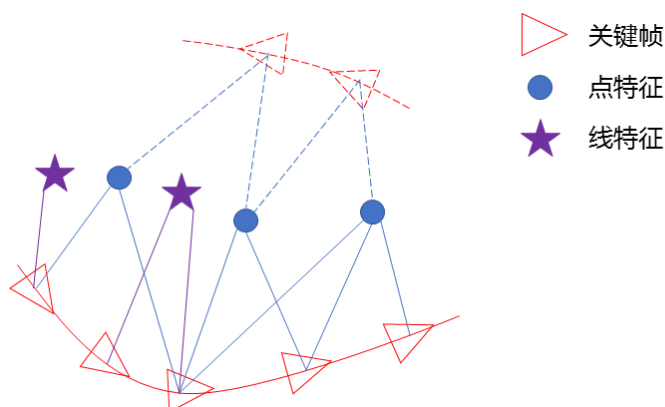


图 4-9 局部 BA 优化示意图

关键帧共视图是指将每个关键帧都作为一个节点，如果两个关键帧能够观察到相同的地图特征（至少 15 个地图点或者 5 个地图线），这两个关键帧之间就存在边，能同时观察到的点和线特征的数量则决定了边的权重。

在 g2o 框架中，图优化的顶点为：

- （1）局部地图的关键帧位姿，类型为：`g2o::VertexSE3Expmap`；
- （2）能被局部关键帧观测到的点特征（称为局部地图点）和线特征（称为局部地图线）的端点，类型为：`g2o::VertexSBAPointXYZ`；
- （3）能观测到局部地图点和局部地图线，但不属于局部地图的关键帧，类型为：`g2o::VertexSE3Expmap`，在优化过程中保持固定不变。

误差边有：

- （1）点特征和以上两种关键帧之间的误差边，继承自二元边，类型为：`g2o::EdgeSE3Project`；
- （2）线特征和以上两种关键帧之间的误差边，该误差边需要自定义，继承自二元边 `BaseBinaryEdge`，定义的类型为：`EdgeLineProjectXYZ`。雅克比矩阵包括误差对于相机位姿的导数，以及误差对线特征的端点的导数。

4.4 本章小结

根据前述章节中的点线综合特征的单目 SLAM 框架流程，详细介绍了跟踪线程、局部建图线程和后端优化线程的工程实现。

5 实验与分析

针对原始的单目视觉 SLAM 在低纹理场景下容易跟踪失败，以及显示部分不利于人机交互的问题，第三章和第四章详细阐述了如何在单目 SLAM 中对线特征进行三维重建，并描述了线特征的误差模型。本章采用标准数据集对本文提出的 PL-SLAM 进行了测试，并与原来的 ORB-SLAM 进行了对比分析。

5.1 实验环境

本文算法的运行环境：Intel CPU 的内核为 i7-7500U，内存为 16.0GB，系统为 Ubuntu 16.04。

5.2 基于标准数据集的实验分析

本文是将 ORB-SLAM 系统进行扩展构建的，在原有的程序框架中添加了线特征的提取、匹配，线特征的三维重建，非线性优化以及可视化界面等，下面简称为 PL-SLAM。为了评估本文提出的综合点线视觉 SLAM 的定位精度，本文使用 TUM 公开数据集中的 RGB-D 序列^[49]做实验，比较了原始 ORB-SLAM 和 PL-SLAM 的定位精度。TUM 数据集中的 RGB-D 序列是通过手持 Kinect 相机测试得到的数据，包括彩色图和深度图，图像序列采集的频率为 30hz，图像分辨率为 640*480，相机的真实轨迹通过一个高精度运动捕捉系统以 100hz 的频率采集得到。因此，可以用 RGB-D 序列以及其真实轨迹来评估 SLAM 系统的定位精度。下图所示为其中一个数据集的处理过程：

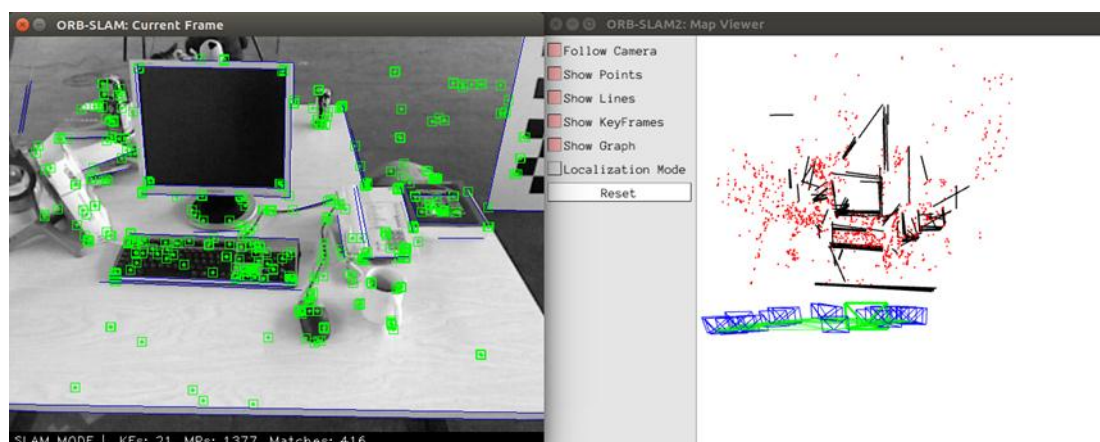


图 5-1 TUM 数据集测试过程

5.2.1 相似矩阵计算

由于本文采用的是单目视觉 SLAM，单目没有尺度性，因此，在计算误差之前，需要对所有的相机轨迹结果进行相似转换。在计算误差之前，还需要通过时间戳将相机的真实位姿与估计的位姿之间进行配对。假设一组配对好的关键帧的三维坐标为：

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}, \mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\} \quad (5-1)$$

假设相似转换矩阵为： $\mathbf{T} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$ ，第 i 对坐标的误差项为：

$$\mathbf{e}_i = \mathbf{p}_i - (s\mathbf{R}\mathbf{q}_i + \mathbf{t}) \quad (5-2)$$

则对所有匹配点对可以构建最小二乘问题：

$$\min_{s, R, t} J = \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i - (s\mathbf{R}\mathbf{q}_i + \mathbf{t})\|_2^2 \quad (5-3)$$

首先，计算两组坐标点的质心： $\mathbf{p} = \frac{1}{n} \sum_{i=0}^n \mathbf{p}_i$ ， $\mathbf{q} = \frac{1}{n} \sum_{i=0}^n \mathbf{q}_i$ ，因此：

$$\begin{aligned} \min_{s, R, t} J &= \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i - (s\mathbf{R}\mathbf{q}_i + \mathbf{t})\|_2^2 \\ &= \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i - s\mathbf{R}\mathbf{q}_i - \mathbf{t} - \mathbf{p} + s\mathbf{R}\mathbf{q} + \mathbf{p} - s\mathbf{R}\mathbf{q}\|_2^2 \\ &= \frac{1}{2} \sum_{i=1}^n \|(\mathbf{p}_i - \mathbf{p} - s\mathbf{R}(\mathbf{q}_i - \mathbf{q})) + (\mathbf{p} - s\mathbf{R}\mathbf{q} - \mathbf{t})\|_2^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\|\mathbf{p}_i - \mathbf{p} - s\mathbf{R}(\mathbf{q}_i - \mathbf{q})\|_2^2 + \|\mathbf{p} - s\mathbf{R}\mathbf{q} - \mathbf{t}\|_2^2 + 2 \underbrace{(\mathbf{p}_i - \mathbf{p} - s\mathbf{R}(\mathbf{q}_i - \mathbf{q}))^T}_{=0} (\mathbf{p} - s\mathbf{R}\mathbf{q} - \mathbf{t})) \\ &= \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p} - s\mathbf{R}(\mathbf{q}_i - \mathbf{q})\|_2^2 + \|\mathbf{p} - s\mathbf{R}\mathbf{q} - \mathbf{t}\|_2^2 \end{aligned} \quad (5-4)$$

对于第一项，记 $\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}$ ， $\mathbf{q}_i^* = \mathbf{q}_i - \mathbf{q}$ ，因此有：

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{p} - s\mathbf{R}(\mathbf{q}_i - \mathbf{q})\|^2 &= \frac{1}{2} \sum_{i=1}^n \|\mathbf{p}_i^* - s\mathbf{R}\mathbf{q}_i^*\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\underbrace{\mathbf{p}_i^{*T} \mathbf{p}_i^*}_{\text{与R无关}} + s^2 \underbrace{\mathbf{q}_i^{*T} \mathbf{R}^T \mathbf{R} \mathbf{q}_i^*}_{=\mathbf{I}} - 2s \mathbf{p}_i^{*T} \mathbf{R} \mathbf{q}_i^*) \end{aligned} \quad (5-5)$$

上式的第一项与 \mathbf{R} 无关，第二项由于 $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ 与 \mathbf{R} 也无关，因此，待优化的函数变为：

$$\sum_{i=1}^n -2s \mathbf{p}_i^{*T} \mathbf{R} \mathbf{q}_i^* = \sum_{i=1}^n -tr(\mathbf{R} \mathbf{q}_i^* \mathbf{p}_i^{*T}) = -tr(\mathbf{R} \sum_{i=1}^n \mathbf{q}_i^* \mathbf{p}_i^{*T}) \quad (5-6)$$

设 $\mathbf{W} = \sum_{i=1}^n \mathbf{q}_i^* \mathbf{p}_i^{*T}$ ， \mathbf{W} 是一个 3×3 的矩阵，对 \mathbf{W} 进行 SVD 分解，得：

$$\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (5-7)$$

其中， $\mathbf{\Sigma}$ 是对角矩阵，对角元素为从大到小的奇异值，而 \mathbf{U} 和 \mathbf{V} 为对角矩阵。当 \mathbf{W} 满秩时， \mathbf{R} 为：

$$\mathbf{R} = \mathbf{U} \mathbf{V}^T \quad (5-8)$$

对于尺度因子 s ，由 5-5 式可得：

$$s = \frac{\sum_{i=1}^n \|\mathbf{p}_i^*\|}{\sum_{i=1}^n \|\mathbf{R} \mathbf{q}_i^*\|} \quad (5-9)$$

根据式 5-4 可得：

$$\mathbf{t} = \mathbf{p} - \mathbf{R} \mathbf{q} \quad (5-10)$$

5.2.2 定位精度分析

实验中采用绝对路径均方根误差（ATE）作为定位精度的评估方法^[45]，该误差是直接测量相机的真实轨迹和估计轨迹之间的距离之差。设相机的真实轨迹为 $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ ，估计的轨迹为 $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$ ，根据 5.2.1 节可以求得估计轨迹到真实轨迹的相似变换矩阵 \mathbf{T} ，则第 i 对匹配对的误差向量为：

$$\mathbf{e}_i = \mathbf{p}_i - (s\mathbf{R}\mathbf{q}_i + \mathbf{t}) = (\Delta x_i, \Delta y_i, \Delta z_i)^T \quad (5-11)$$

然后，对以上所有匹配对的误差求均方根误差，最终的结果为：

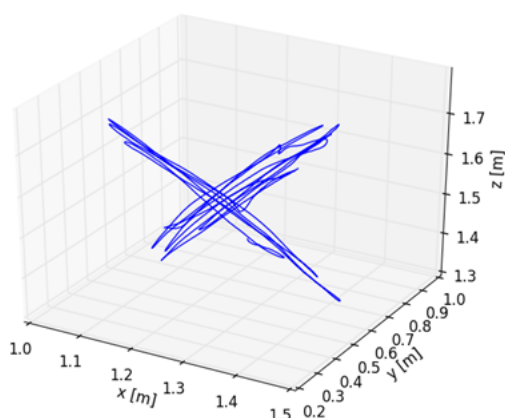
$$RMSE(e_i) = \left(\frac{1}{n} \sum_{i=1}^n \| \mathbf{e}_i \|^2 \right)^{1/2} = \left(\frac{1}{n} \sum_{i=1}^n (\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2) \right)^{1/2} \quad (5-12)$$

除此之外，考虑到单目视觉 SLAM 初始化、位姿优化和全局重定位阶段的随机性，所有的实验均运行五次，然后取五次结果的中位数进行评估。下表列出了该实验的结果，在表中粗体标注了每一序列中的最小误差：

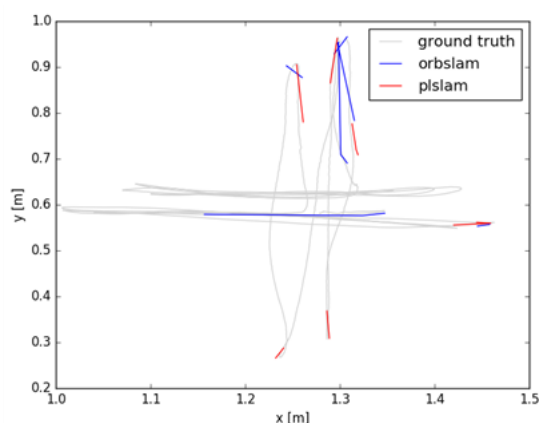
表5-1 定位误差分析
关键帧绝对轨迹均方根误差，cm

TUM RGB-D 数据集	ORB-SLAM	PL-SLAM
f1_xyz	0.92392	0.7557
f2_xyz	0.25124	0.6474
f2_desk_person	0.81612	1.8317
f2_360_kidnap	6.05768	9.7556
f3_long_office	2.4703	1.9130
f3_sit_xyz	1.04288	0.9084
f3_sit_halfsph	1.65082	1.6249
f3_walk_halfsph	16.03024	7.5984

图 5-2~9 分别为上述数据集的相机真实运动轨迹以及用 ORB-SLAM 和 PL-SLAM 估计的相机位姿图对比。



(a) 相机真实运动轨迹



(b) 轨迹估计对比

图 5-2 f1_xyz 轨迹图

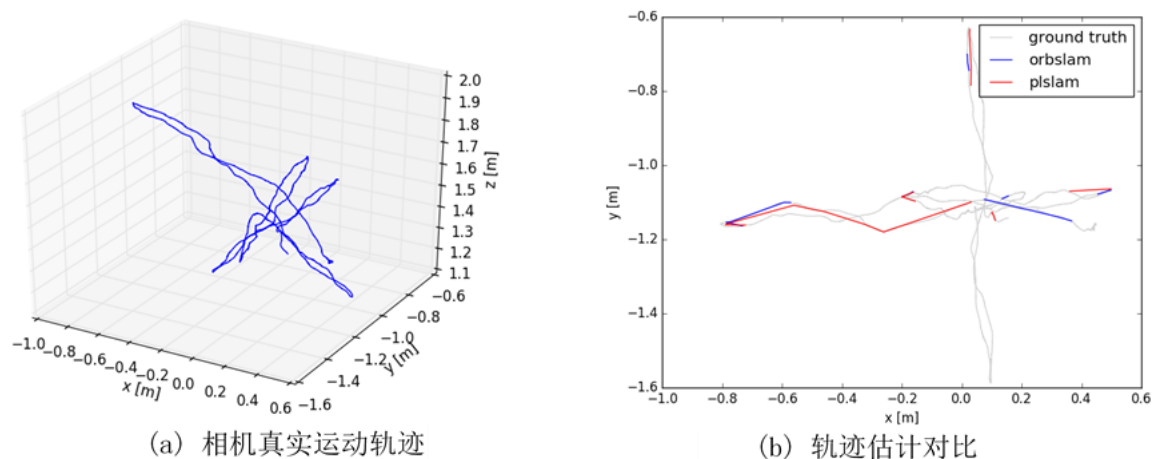


图 5-3 f2_xyz 轨迹图

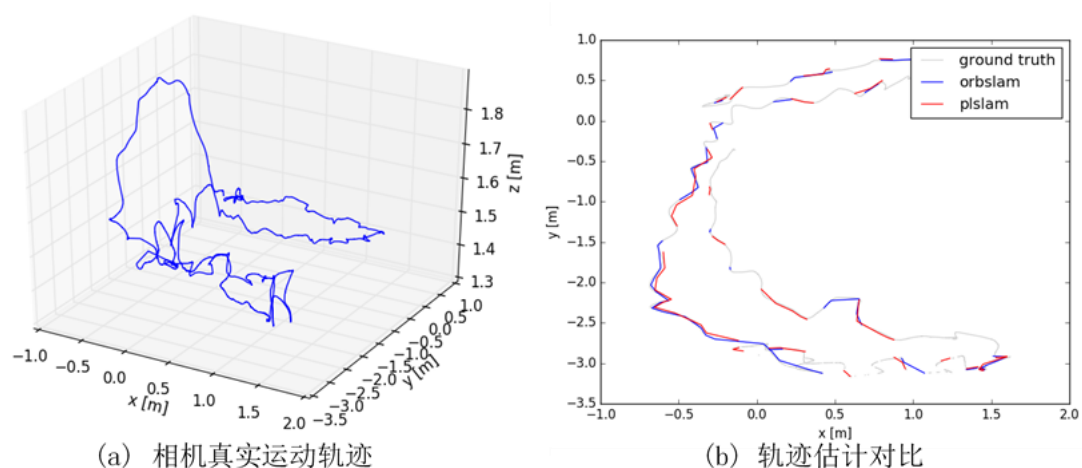


图 5-4 f2_desk_person 轨迹图

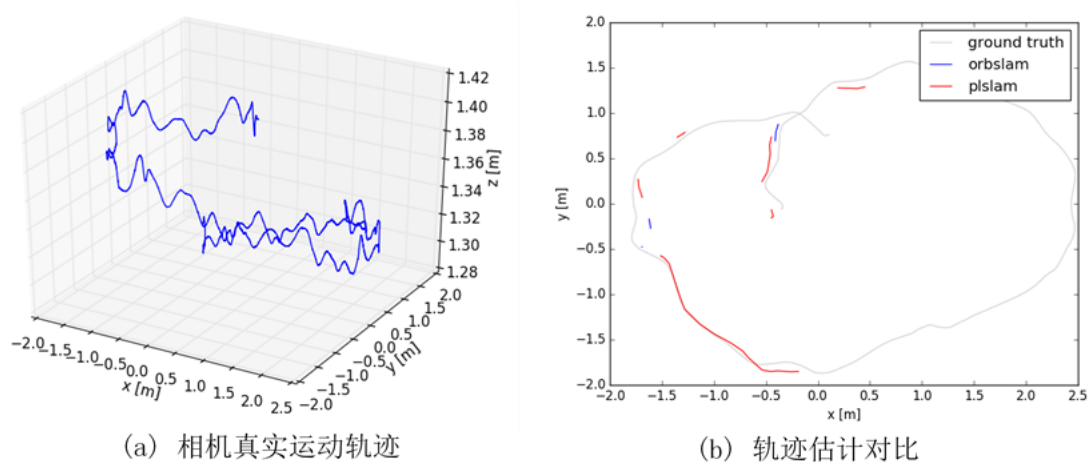


图 5-5 f2_360_kidnap 轨迹图

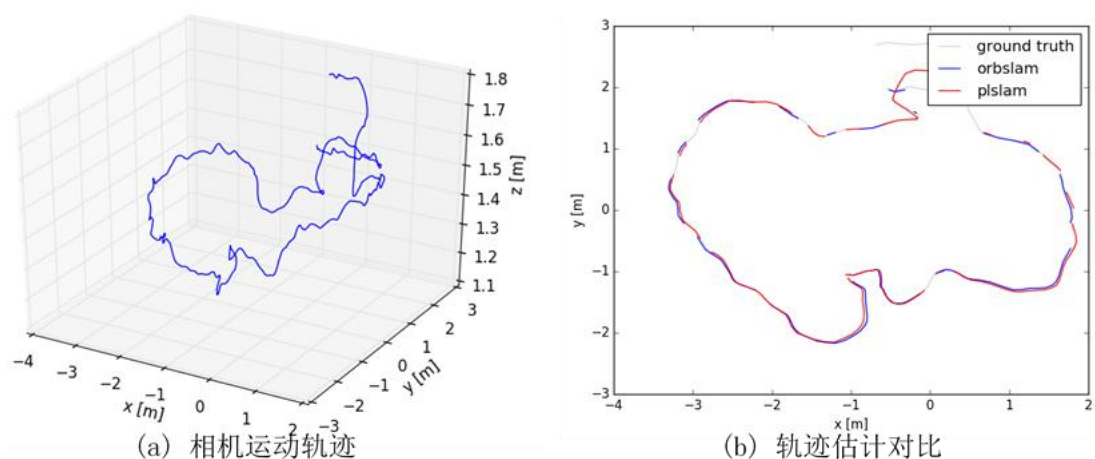


图 5-6 f3_long_office 轨迹图

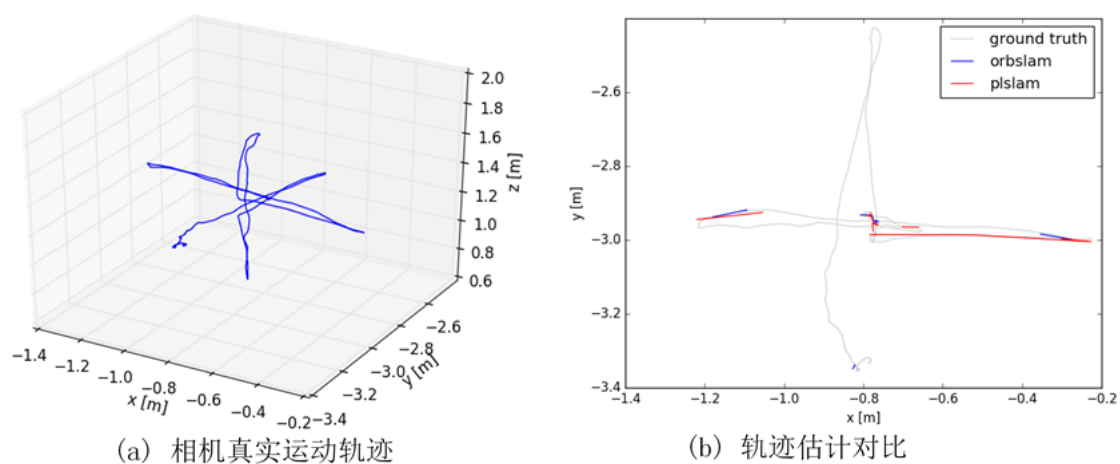


图 5-7 f3_sit_xyz 轨迹图

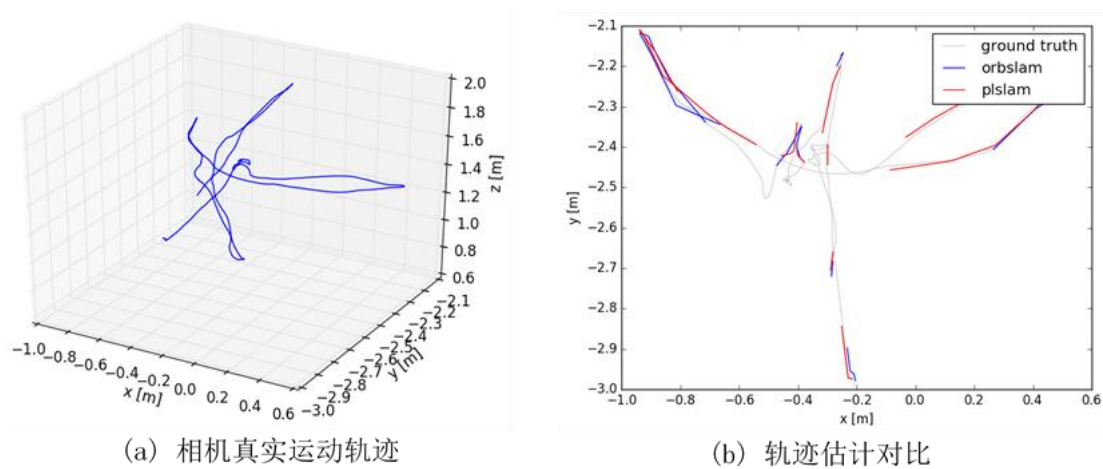


图 5-8 f3_sit_halfsph 轨迹图

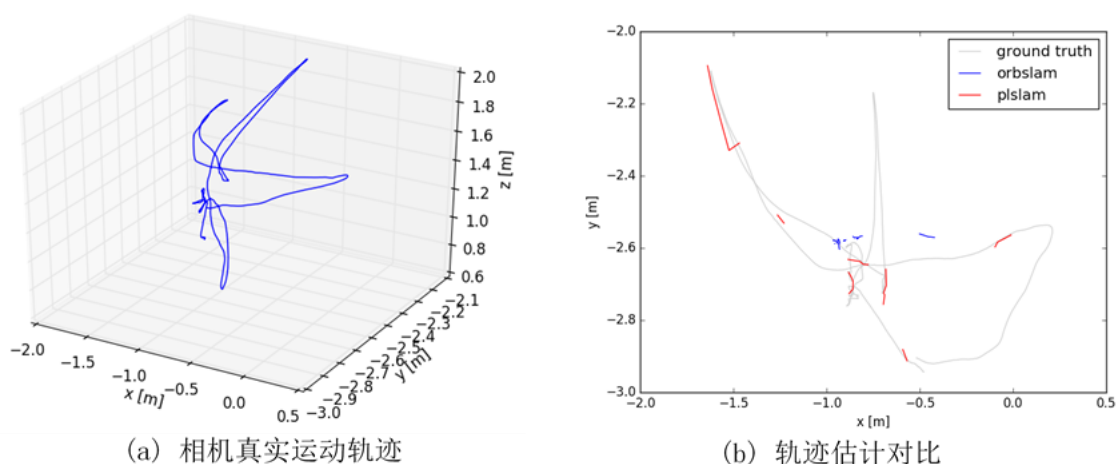


图 5-9 f3_walk_halfsph 轨迹图

从表中数据可以看出，本文提出的 PL-SLAM 在 8 个序列中有 5 个序列的结果取得了较好的结果，对相机位姿估计的精度有一定的提升，但在另外 3 个场景中，相机定位的精度并不会因为线特征而有所改善。具体分析如下：

上述数据集中，f1_xyz, f2_xyz, f2_360_kidnap 和 f3_long_office 中的场景均为静态场景，其中 f2_360_kidnap 中会对相机进行“绑架”（即盖住相机镜头）测试。在这几组场景中，PL-SLAM 在相机运动比较剧烈的 f1_xyz 和 f3_long_office 中的精度较 ORB-SLAM 好，而在运动较为平缓的 f2_xyz 中的定位精度比 ORB-SLAM 的差。在 f2_360_kidnap 的轨迹估计对比图中可以看出，虽然 PL-SLAM 的定位精度低于 ORB-SLAM 的定位精度，但是相机“绑架”后重定位时，PL-SLAM 能够较快进行重定位。

在 f2_desk_person, f3_sit_xyz, f3_sit_halfsph 和 f3_walk_halfsph 的场景中均有动态障碍物，这几组的实验中，除了 f2_desk_person 数据集，PL-SLAM 的定位精度优于 ORB-SLAM 的定位精度。

综上所述，PL-SLAM 定位精度较好的几组数据均为相机运动剧烈（如 f1_xyz 序列和 f3_long_office）或者场景中有动态的人运动（f3_sit_xyz, f3_sit_halfsph 等）；而在相机运动平缓的场景中（f2_xyz），PL-SLAM 的定位精度反而比原来的 ORB-SLAM 要低。主要原因在于，线特征检测不如点特征检测稳定，本文采用的 LSD 线段检测算法的速度虽然与像素数量成正比，检测出来的线段在图中分布也较为均匀，但是线段的端点不如特征点稳定，而本文在三角化以及后端优化中使用了线段的端点，所以引入的噪声较

ORB 多，使得相机位姿估计的精度有所下降。

5.2.3 时间分析

当在视觉 SLAM 中添加线特征后，虽然定位精度和对环境的鲁棒性有了提高，但是也增加了计算复杂度，进而计算时间增加。本文测试了添加线特征后的 SLAM（记为 PL-SLAM）在跟踪线程和局部建图线程两个环节中所需要的时间，并和原始的 ORB-SLAM 做了对比，如下表所示：

表5-2 系统耗时

平均执行时间（ms）			
线程	过程	PL-SLAM	ORB-SLAM
跟踪线程	特征提取	43.382	19.797
	初始位姿估计	5.820	4.768
	跟踪局部地图	9.248	5.959
	总计	58.450	30.524
	频率	17hz	33hz
	插入关键帧	18.145	15.865
	剔除特征	1.017	0.71
局部建图线程	创建地图特征	201.737	18.325
	局部 BA	179.071	118.417
	剔除冗余关键帧	5.884	5.613
	总计	405.854	158.93
	频率	2.5hz	6.3hz

从表中可以看出，在跟踪线程中，PL-SLAM 主要是在特征提取处耗时比 ORB-SLAM 长，这里主要是因为 LSD 直线检测算法比 ORB 特征提取耗时；在局部建图线程中，PL-SLAM 在创建地图特征时所耗时间远大于 ORB-SLAM 所需时间，而局部 BA 和剔除特征所需时间比 ORB-SLAM 稍长一些。由于跟踪线程和局部建图线程是并行进行的，因此跟踪线程所耗时间决定了 SLAM 系统的处理能力。添加线特征的 PL-SLAM 的跟踪线

程处理时间平均为 58.45ms, 处理图像序列的帧率为 17Hz, 可以基本达到一个实时 SLAM 的目标。后期可以在 LSD 提取特征和创建地图特征（尤其是线特征）上继续进行优化, 提高处理图像的帧率。

5.3 本章小结

本章首先介绍了实验环境, 然后对点线综合的单目视觉 SLAM 和 ORB-SLAM 在 TUM 中的 RGB-D 数据集上做了实验验证, 构建了综合点、线特征的环境地图, 然后对实验中的定位精度和时间效率做了分析。

6 总结与展望

6.1 全文总结

传统的视觉 SLAM 的前端主要是通过点特征来估计相机的位姿，建立环境地图。但是点特征在场景光线变化、相机视角剧烈运动时难以跟踪，且在部分低纹理场景下不能保证提取到足够多的特征点，同时三维重建的稀疏地图不利于人机交互。因此，本文在基于特征点的单目视觉 SLAM 中添加了线特征，改善系统在场景环境剧烈变化时的鲁棒性。本文的主要工作包括以下内容：

(1) 在单目 ORB-SLAM 的基础之上，修改了跟踪线程和局部建图线程，构建了基于点、线综合特征的单目视觉 SLAM 系统，使得原始的单目 ORB-SLAM 在低纹理场景下也可以使用。

(2) 采用 LSD 快速检测线特征，描述子用二进制描述子 LBD，从而加速特征匹配。空间线特征采用端点法描述，根据线特征的对极约束模型，使用直接线性变换方法计算出线特征的空间坐标，保证了与点特征的尺度一致性。

(3) 构建线特征的误差模型时，使用线特征的重投影端到检测直线的距离作为误差，然后推导了线特征的雅克比矩阵的解析形式，最后结合点特征的误差和线特征的误差，分别构建了位姿图优化、局部图优化和全局图优化的模型。

(4) 采用 TUM 中的 RGB-D 数据集对修改后的 PL-SLAM 和原始的 ORB-SLAM 做了测试，实验结果表明，在相机剧烈运动和场景中存在大量的动态障碍物时，PL-SLAM 的定位精度较原始的 ORB-SLAM 有了提升。

6.2 论文展望

在目前本文所做的工作基础之上，可以通过以下工作继续改善 PL-SLAM 的精度和鲁棒性：

(1) 目前深度学习在图像识别、匹配、分类的快速发展和应用，可考虑在前端引入深度学习完成点线特征的识别和匹配；

(2) 本文的 SLAM 在初始化时仍然采用原始的 ORB-SLAM 的初始化方法，在只

有线特征的环境下仍不能进行初始化，因此，可以建立综合点线特征的单目 SLAM 的初始化方法，提高系统对环境的适应性。

（3）本文只修改了 ORB-SLAM 的跟踪线程和局部建图线程，可以建立基于线特征的词袋模型，使 PL-SLAM 在闭环检测的时候也能通过线特征进行判断。

致 谢

时光荏苒，转眼间我的研究生生活即将画上句号，而我拉着行李箱，抱着录取通知书兴致昂扬的走进机械学院的场景还历历在目。在过去的三年学习生活中，老师、同学、父母给予了太多的帮助与支持，也是因为他们，我才没有虚度这三年的光阴。

首先，感谢我的恩师熊良才副教授，熊老师温和宽厚的性格使得师生关系十分融洽，能够让我积极提出自己对很多问题的看法和思路。在对待研究生选题方向上，熊老师给我指定了一些方向，并在具体实施环节上给了我极大的自由。此外，还非常感谢恩师让我能在研究生期间去企业实习，不仅让我找到了科研学习的方向，也让我增长了对外界的认知。在熊老师的悉心指导下，我才能够顺利完成毕业论文。恩师的言传身教，也将成为我以后人生道路的一盏指明灯。

然后，非常感谢男友在我研究生阶段最迷茫的时候不断鼓励我、支持我，让我能够有信心去面对自己曾经惧怕的事物，安心去学习、成长；同时还要感谢实验室的师兄弟们，实验室良好的学习氛围对于研究生阶段的成长是必不可少的。感谢同年级的刘超峰同学，在基础数学理论上给予的指导，谢谢章露、李锐、黄晓晓、胡洪斐、翟康师兄在学习道路上给我树立的楷模和榜样，还要感谢镇亮、卜云德、刘磊、张珂师弟给予的支持。

最后，感谢我的父母和弟弟，在我独自在外求学的过程中，是你们一直在背后无时无刻的关心我、支持我，让我可以没有任何后顾之忧的去做自己想做的事，你们的支持与理解，支撑着我不断向前去奋进。

参考文献

- [1] Choi H, Yang K W, Kim E. Simultaneous Global Localization and Mapping[J]. IEEE/ASME Transactions on Mechatronics, 2014,19(4):1160-1170.
- [2] Thrun S. Probabilistic robotics[M]. MIT Press, 2005.
- [3] 谢晓佳. 基于点线综合特征的双目视觉SLAM方法[D]. 浙江大学, 2017.
- [4] Leonard J J, Durrantwhyte H F. Simultaneous map building and localization for an autonomous mobile robot[J]. IEEE/RSJ International Workshop on Intelligent Robots & Systems, 1991.
- [5] ZhenheChen, JagathSamarabandu, RangaRodrigo. Recent advances in simultaneous localization and map-building using computer vision[J]. Advanced Robotics, 2007,21(3-4):233-265.
- [6] Strasdat H, Montiel J M M, Davison A J. Visual SLAM: Why filter? ☆[J]. Image & Vision Computing, 2012,30(2):65-77.
- [7] Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age[J]. IEEE Transactions on Robotics, 2017,32(6):1309-1332.
- [8] 权美香, 朴松昊, 李国. 视觉SLAM综述[J]. 智能系统学报, 2016(06):768-776.
- [9] Davison A J. Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007,29.
- [10] Pupilli M, Calway A. Real-Time Camera Tracking Using Known 3D Models and a Particle Filter: International Conference on Pattern Recognition, 2006[C].
- [11] Klein, Georg, Murray, et al. Parallel Tracking and Mapping for Small AR Workspaces[J]. Proc.of IEEE & Acm Int.sympo.on Mixed & Augmented Reality, 2007:1-10.
- [12] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2017,31(5):1147-1163.
- [13] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[M]. Springer International Publishing, 2014.
- [14] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry: IEEE International Conference on Robotics and Automation, 2014[C].
- [15] Smith P. Real-Time Monocular SLAM with Straight Lines: British Machine Vision Conference 2006, Edinburgh, Uk, September, 2006[C].
- [16] Eade E, Drummond T. Edge landmarks in monocular SLAM[J]. Image & Vision Computing, 2009,27(5):588-596.
- [17] Zhang G, Suh I H. Building a partial 3D line-based map using a monocular SLAM: IEEE International Conference on Robotics and Automation, 2011[C].
- [18] Zhou H, Zou D, Pei L, et al. StructSLAM: Visual SLAM With Building Structure Lines[J]. IEEE Transactions on Vehicular Technology, 2015,64(4):1364-1375.
- [19] Zhang G, Jin H L, Lim J, et al. Building a 3-D Line-Based Map Using Stereo SLAM[J].

- IEEE Transactions on Robotics, 2015,31(6):1364-1377.
- [20]Lu Y, Song D. Robust RGB-D Odometry Using Point and Line Features: IEEE International Conference on Computer Vision, 2015[C].
- [21]Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004,60(2):91-110.
- [22]Bay H, Tuytelaars T, Gool L V. SURF: speeded up robust features: European Conference on Computer Vision, 2006[C].
- [23]Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF: International Conference on Computer Vision, Barcelona, 2011[C].
- [24]Rosten E, Drummond T. Machine Learning for High-Speed Corner Detection[M]. Springer Berlin Heidelberg, 2006.
- [25]Muja M. FLANN -Fast Library for Approximate Nearest Neighbors User Manual[J]. 2009.
- [26]Hartley R, Zisserman A. Multiple View Geometry in Computer Vision[M]. Cambridge University Press, 2003.
- [27]邱茂林, 马颂德, 李毅. 计算机视觉中摄像机定标综述[J]. 自动化学报, 2000,26(1):43-55.
- [28]熊有伦. 机器人技术基础[M]. 华中理工大学出版社, 1996.
- [29]Kschischang F R, Frey B J, Loeliger H A. Factor graphs and the sum-product algorithm: IEEE TRANSACTIONS ON INFORMATION THEORY, 1998[C].
- [30]Shewhart W A, Wilks S S. Robust Statistics: Theory and Methods[M]. John Wiley, 2006.
- [31]Dellaert F. Factor Graphs and GTSAM: A Hands-on Introduction[J]. Georgia Institute of Technology, 2012.
- [32]Kümmerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization: IEEE International Conference on Robotics and Automation, 2011[C].
- [33]Alvarruiz Serrano A. A Ceres solver based bundle adjustment module[J]. 2014.
- [34]Kaess M, Ranganathan A, Dellaert F. iSAM: Incremental Smoothing and Mapping[J]. IEEE Transactions on Robotics, 2008,24(6):1365-1378.
- [35]Polok L, Ila V, Solony M, et al. Incremental Block Cholesky Factorization for Nonlinear Least Squares in Robotics: Robotics: Science and Systems, 2013[C].
- [36]Sivic J, Zisserman A. Video Google: A Text Retrieval Approach to Object Matching in Videos: IEEE International Conference on Computer Vision, 2003[C].
- [37]Nister D, Stewenius H. Scalable Recognition with a Vocabulary Tree: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006[C].
- [38]Cummins M. FAB-MAP : Probabilistic localization and mapping in the space of appearance[J]. Int.j.robot.res, 2008,27(6):647-665.
- [39]Milford M J, Wyeth G F. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights: IEEE International Conference on Robotics and Automation, 2012[C].
- [40]Ho K L, Newman P. Loop closure detection in SLAM by combining visual and spatial appearance[J]. Robotics & Autonomous Systems, 2006,54(9):740-749.

- [41]Ding L, Goshtasby A. On the Canny edge detector[J]. Pattern Recognition, 2001,34(3):721-725.
- [42]Gioi R G V, Jakubowicz J, Morel J M, et al. LSD: A line segment detector[J]. Image Processing on Line, 2012,2(4):35-55.
- [43]Wang Z, Liu H, Wu F. MSLD: A robust descriptor for line matching[J]. Pattern Recognition, 2009,42(5):941-953.
- [44]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 & Image Representation, 2013,24(7):794-805.
- [45]马颂德, 张正友. 计算机视觉:计算理论与算法基础[M]. 科学出版社, 1998.
- [46]Vakhitov A, Funke J, Moreno-Noguer F. Accurate and Linear Time Pose Estimation from Points and Lines[M]. Springer International Publishing, 2016.
- [47]郝矿荣, 丁永生. 机器人几何代数模型与控制[M]. 科学出版社, 2011.
- [48]Triggs B, Mclauchlan P, Hartley R. Bundle Ajustment -- A Modern Synthesis[J]. Vision Algorithms Theory & Practice, 2000,1883(1883):153-177.
- [49]Sturm J, Burgard W, Cremers D. Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark[J]. Proc of the Workshop on Color, 2012.